

# Counter-Interference Adapter for Multilingual Machine Translation

Yaoming Zhu<sup>1</sup>    Jiangtao Feng<sup>1</sup>    Chengqi Zhao<sup>1</sup>    Mingxuan Wang<sup>1</sup>    Lei Li<sup>2\*</sup>

<sup>1</sup> ByteDance AI Lab

<sup>2</sup> University of California, Santa Barbara

{zhuyaoming, fengjiangtao, zhaochengqi.d, wangmingxuan.89}@bytedance.com

lilei@cs.ucsb.edu

## Abstract

Developing a unified multilingual model has long been a pursuit for machine translation. However, existing approaches suffer from performance degradation — a single multilingual model is inferior to separately trained bilingual ones on rich-resource languages. We conjecture that such a phenomenon is due to interference caused by joint training with multiple languages. To accommodate the issue, we propose CIAT, an adapted Transformer model with a small parameter overhead for multilingual machine translation. We evaluate CIAT on multiple benchmark datasets, including IWSLT, OPUS-100, and WMT. Experiments show that CIAT consistently outperforms strong multilingual baselines on 64 of total 66 language directions, 42 of which see above 0.5 BLEU improvement. Our code is available at <https://github.com/Yaoming95/CIAT>.

## 1 Introduction

Machine translation (MT) is a core task in natural language processing. In recent years, neural machine translation (NMT) approaches have made tremendous progress and takes the lead in the field (Bahdanau et al., 2015; Vaswani et al., 2017; Johnson et al., 2017). Conventionally, each NMT model only tackles a single language direction (e.g. English  $\rightarrow$  German). A commonly used model like Transformer has  $S = 240$  million parameters. Therefore, Translating  $N$  language pairs requires training models separately for each direction, resulting in  $S \times N$  total parameters. The huge size of all models for every language direction can be too costly to deploy, considering more than 100 popular languages worldwide. Hence, developing a single unified yet parameter-efficient model for multilingual machine translation, enabling the translation of multiple directions, becomes crucially important. There is much effort towards multilingual

machine translation. Johnson et al. (2017) first proposed training a single multilingual model via an additional target language tag, which became the paradigm for training multilingual MT models henceforth (Gu et al., 2018; Tan et al., 2018, 2019; Siddhant et al., 2020). Their approach is simple and parameter-efficient, however, it often lags behind separately trained bilingual models, especially on resource-rich language pairs, where the phenomenon is known as performance degradation (Aharoni et al., 2019).

This paper analyzes the performance degradation between multilingual MT models and separate bilingual ones. Prior research suggests that unified embedding on a joint vocabulary leads to meaning conflation deficiency in multilingual training (Camacho-Collados and Pilehvar, 2018). For example, words with identical spelling may have distinct meanings in different languages — *bride* in English refers to a woman soon to get married while in French it means horse bridle. When it comes to machine translation, the effect goes beyond word embedding. As a single model has bounded capacity, the multilingual learning may cause negative influences among shared parameters (Liu et al., 2017; Zhang et al., 2020). We conjecture that such a performance degradation is due to the interference across languages brought by joint training on multiple language directions. Such interference affects both joint token embedding and representations from intermediate layers. We argue that resolving the interference is critical to improving multilingual translation performance.

Inspired by the insights above, we propose *Counter-Interference Adapter* for multilingual machine Translation (CIAT). The CIAT includes a major multilingual base model (i.e., a multilingual Transformer), which is universally pre-trained on multiple language directions, and two kinds of designed adapter modules, which are trained on specific language directions. Specifically, we propose

\*Work is done while at ByteDance.

*embedding adapter* and *layer adapter* to reduce multilingual interference among embedding and intermediate layers, respectively. We also seek a new parallel connection method for adapter units, which is more effective for multilingual MT than the previous series connection. We validate CIAT among three real-world datasets and observe that CIAT gains significant improvements in translation performance over all other multilingual baselines.

Our contributions are as follows: *a)* We analyze performance degradation in multilingual NMT and formulate it as two issues; *b)* We propose CIAT, an adapter-based framework to tackle the two issues and enhance the performance of the multilingual model with small amounts of extra parameters; *c)* We demonstrate the efficacy of CIAT through extensive experiments on IWSLT, OPUS-100, and WMT benchmark datasets, surpassing other multilingual models over most of the translation directions.

## 2 Related Work

**Multilingual Machine Translation.** The multilingual MT enjoys a rich research history, dating back to the age of statistical machine translation (Gao et al., 2002; Haffari and Sarkar, 2009; Seraj et al., 2015). In recent years, the prosperity of neural machine translation (NMT) has led to the growing prominence and popularity of multilingual MT systems. The encoder-decoder framework has made the de facto standard for NMT (Bahdanau et al., 2015; Vaswani et al., 2017). Dong et al. (2015) did the pioneering work on extending conventional NMT to one-to-many translation, where the authors added a distinct decoder for each target language. Firat et al. (2016) further extended such framework into many-to-many settings by building exclusive encoders and decoders for each language. Those attempts still faced problems such as low parameter utilization. On the other hand, Lee et al. (2017) treated all sources as the same language by translating on a character level. However, it only meets the many-to-one scenarios. Johnson et al. (2017) managed to train a single model that applied to multiple translation directions. Their solution is relatively simple: they attached a dedicated token at the beginning of the source sentence to specify the target language, while the rest of the model was shared among all languages. The paper has set a milestone of multilingual MT and has become the basis for most subsequent work.

Recent studies paid more attention to the performance improvement of multilingual models based on Johnson et al. (2017)’s effort. Several improved the model with external knowledge from human or other models: Tan et al. (2018) boosted the multilingual model by knowledge distillation, Tan et al. (2019) pre-clustered languages to assist similar languages. Several studies enhance the model from data: Xia et al. (2019) and Siddhant et al. (2020) conducted data augmentation to low-resource languages via related high-resources or monolingual data. Taitelbaum et al. (2019) improved translation with relevant auxiliary languages. Some other studies enhanced the Transformer model by introducing language-aware modules and learning language-specific representation (Wang et al., 2019; Zhu et al., 2020).

### Adapter Network for Machine Translation.

Our design derives from the residual adapters of the domain adaptation task. Concretely, Rebuffi et al. (2017) proposed the residual adapters in the computer vision area. They appended small networks (named adapters) to a pre-trained base network and only tuned the adapter on the specific task. Houlsby et al. (2019) adopted the idea into NLP domain adaptation tasks and designed the adapter for the Transformer, as shown in Fig. 2a. Bapna and Firat (2019) further extended the model to MT domain adaptation, and they regarded multilingual MT as a domain adaptation task. Based on their design, Philip et al. (2020) proposed the monolingual adapter for easy extension to new pairs, and Zhang et al. (2021) introduced conditional language-specific routing strategy (CLSR) to enhance model capacity in language-specific representation. However, their adapter designs followed a *serial* connection manner, which might be limited for multilingual MT. We will discuss this in the following sections.

## 3 Challenges on Multilingual Machine Translation

The ultimate goal of multilingual machine translation is to build a universal model to achieve mutual translation among all natural languages. That is, given a source sentence  $s$  and the target language  $l$ , the multilingual MT system shall output a sentence that resembles human reference  $t$ .

Currently, Transformer (Vaswani et al., 2017) gains popularity and becomes the paradigm for state-of-the-art NMT systems. Here, we follow

the recent implementations (Klein et al., 2017; Vaswani et al., 2018) of the pre-norm transformer, whose layer normalization is applied to the input of each sub-layer. The transformation of  $i$ -th sub-layer taking  $x_i$  as input can be formulated as:

$$x_{i+1} = F_{\theta}(x_i) = \text{sub-layer}_{\theta}(\text{LN}(x_i)) + x_i \quad (1)$$

where  $\text{LN}(\cdot)$  is the layer normalization function and sub-layer function denotes one basic layer, that is, self-attention, cross-attention or feed-forward layer.  $\theta$  is the trainable parameters of the sub-layer.

The classic multilingual translation approach (Johnson et al., 2017) takes sentences from all language pairs and results in a universal model that translates across various languages. However, as previous researches have discovered, the multilingual model yields an inferior performance on high-resource languages compared to the bilingual models under the same configuration. In this paper, we reconsider the limitations of multilingual models and attribute the performance degradation to the following two issues:

**Multilingual Embedding Deficiency** Camacho-Collados and Pilehvar (2018) addressed the *meaning conflation deficiency* problem of the word embedding as a single vector is limited for representing polysemy. We extend the meaning conflation deficiency into the multilingual scenario. Generally, words/tokens may have unrelated or even opposite meanings in different languages. For example, “娘” denotes mother in Chinese but daughter in Japanese. As the word embeddings are usually jointly trained on a multilingual corpus, representing a multilingual word with just one single vector may burden the model’s semantic representation. We refer to the problem as *multilingual embedding deficiency*.

**Multilingual Interference Effects** Besides the word embedding, the insufficient capacity of a single NMT model also bottlenecks its performance on multilingual tasks (Aharoni et al., 2019; Zhang et al., 2020). The parameter-sharing among different languages may be a potential cause of negative interference (Liu et al., 2017; Wang et al., 2020). We here formulate this phenomenon as *multilingual interference effects*; that is, when a single model tries to learn multiple languages simultaneously, the extracted language features interfere with each other impose adverse effects upon overall performance. Accordingly, we regard the model trained

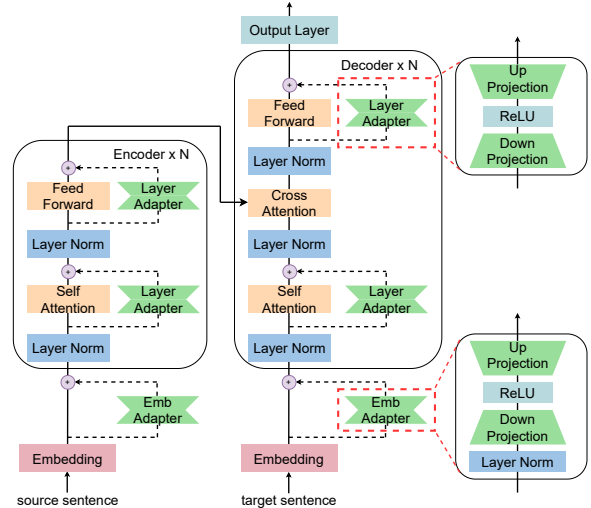


Figure 1: The schematic diagram shows the overall layout of the model and the architecture of the adapters. First, the standard Transformer (modules connected by the solid line) is trained on the entire corpus as a multilingual base model, and its parameters are frozen afterward. The adapters (modules connected by the dashed line) are then plugged into the base model and fine-tuned on the bilingual corpus to enhance model performance on the specific language pair.

on bilingual data offers the approximately optimal solution on language representation, compared to which the representation of the multilingual model is bias-influenced:

$$F_{\theta^m}(x_i) = F_{\theta^b}(x_i) + \delta_i \quad (2)$$

where  $\theta^b$  denotes the parameters of bilingual baselines and  $\theta^m$  indicates the multilingual model.  $\delta_i$  is the interference noise in the  $i$ -th layer.

## 4 Proposed Method

We propose a Counter-interference Adapter for Multilingual Machine Translation (CIAT) to address the two issues mentioned above with adapter-based architectures: embedding adapter and layer adapter. Figure 1 illustrates the overall architecture of CIAT, which we will describe in detail.

### 4.1 Embedding Adapter

As discussed in section 3, a jointly trained multilingual word embedding  $\mathbf{E}^m$  could be problematic as a word may have different meanings among multiple languages. Empirically, we can fine-tune the whole embedding matrix for each language pair to address the multilingual embedding deficiency. However, tuning the whole matrix is quite expensive as the embedding matrix occupies a large part

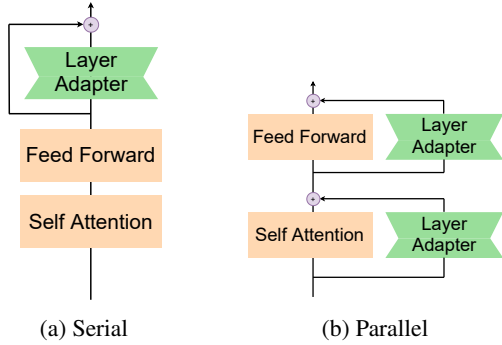


Figure 2: Architecture of Transformer with serial (Bapna and Firat, 2019) and parallel layer adapters (ours), where the add-and-norm is omitted for simplicity. Compared to the Serial design, our parallel layer adapter design de-noise the multilingual interference *pre* to the Transformer layers rather than *post* to the layers.

of model parameters, which also violets the advantage of parameter sharing in multilingual NMT. We hence introduce the embedding adapter to approximate the fine-tuned embedding matrix  $\mathbf{E}^f$  with much fewer parameters:

$$\tilde{\mathbf{E}}^f[w] = \mathbf{E}^m[w] - G_\psi(\mathbf{E}^m[w]) \quad (3)$$

where  $\mathbf{E}[w]$  is the embedding vector of token  $w$  under embedding  $\mathbf{E}$ , and  $\tilde{\mathbf{E}}^f$  is the approximation of the fine-tuned embedding matrix  $\mathbf{E}^f$ .  $G_\psi$  is our proposed embedding adapter parameterized by  $\psi$ .

Fig. 1 shows the layout of the embedding adapter, including a layer normalization (Ba et al., 2016) and a fully connected feed-forward neural network. Following the suggestions of Housby et al. (2019), we choose *bottle-neck* architecture for the adapter module to save the parameters: the first layer down-projects the embedding dimension  $d$  to a smaller size  $m$ , while the second layer projects it back to  $d$  dimension. We select ReLU (Nair and Hinton, 2010) as the activation function for the middle layer while using no activation for the output layer.

## 4.2 Parallel De-noise Layer Adapter

As formulated in Eq. 2, the multilingual representation  $F_{\theta^m}(x_i)$  is regarded as a bias-influenced one compared to the bilingual representation  $F_{\theta^b}(x_i)$ . To alleviate the interference, we introduce the layer adapter  $G_\phi$  to model the bias term  $\delta_i$ :

$$G_\phi(x_i) \approx -\delta_i = F_{\theta^b}(x_i) - F_{\theta^m}(x_i) \quad (4)$$

Eq. 4 means that the layer adapter shares the same input  $x_i$  as the sub-layer and de-noises the output

$F_{\theta^m}(x_i)$ . As a result, the output of  $i$ -th sub-layer is adapted to  $x'_{i+1} = F_{\theta^m}(x_i) + G_\phi(x_i)$ .

We connect layer adapters parallel to sub-layers, as shown in Fig. 1. The architecture of the layer adapter is the same as the embedding adapter, except for the removed layer normalization. In practice, we find that the layer adapter can share the layer normalization structure with the corresponding sub-layer to achieve the best performance.

**Comparison to the Serial Adapter Design** As the first study introduced adapter networks to machine translation, Bapna and Firat (2019) also conducted experiments on multilingual machine translation. They append adapters *serial* to the model architecture with a residual connection as illustrated in Fig. 2a, while we argue that our *parallel* connection is more suitable for multilingual machine translation. Compared to Eq. 4, we can formulate the serial style adapters as:

$$G_\phi(F_{\theta^m}(x_i)) \approx -\delta_i = F_{\theta^b}(x_i) - F_{\theta^m}(x_i) \quad (5)$$

where the adapter  $G$  receives the bias-influenced hidden states  $F_{\theta^m}(x_i)$  other than the original input  $x_i$ . However, the bias-influenced  $F_{\theta^m}(x_i)$  may not be distinguishable for training adapters, which is especially the case when the multilingual model is inferior, making  $F_{\theta^m}(x_i)$  fail to capture enough information of languages.

In contrast, our parallel design de-noise the bias-influenced term *pre* to the sub-layers. Corresponding to Eq. 4, the parallel layer adapters receive the same input as the sub-layers and de-noise directly to the output, which is a more intuitive and natural design for de-noising  $\delta_i$ , since the adapter is independent of the sub-layer output. The parallel adapter can also be regarded as the low-rank “patch” for the corresponding sub-layer, which adjusts the parametrization of the high-rank sub-layer to the specific language pair and fix the multilingual interference.

## 4.3 Model Training

The training process of the whole model consists of two phases: the pre-training on the multilingual model and the learning of the adapter modules. First, we pre-train the standard Transformer on the entire corpus, making a universal multilingual model. The parameters of the Transformer are frozen once the model converges. Then, we “plug-in” the randomly initialized adapters for each specific language pair and only fine-tune the adapter

parameters on this pair. Since the base Transformer model is frozen, each plugged adapter’s learning process is independent of other adapters. During the inference stage, we only apply the base model and the corresponding adapter to translate sentences into the target language.<sup>1</sup> All the adapters are plug-able during the inference stage: disabling the adapter will degenerate the model into a basic multilingual translation model, and when the model is required to translate a specific pair, we just “plug-in” the specific adapter into the base model.

Theoretically, the well-trained adapted network guarantees a better performance compared to the multilingual base. When the layer adapter is disabled (*i.e.*,  $G_\phi(x_i) \equiv 0$  when all parameters are zero), the model is reduced to the multilingual base model. With proper training, the layer adapter should boost the model performance.

## 5 Experiments

We conducted experiments on three multilingual translation datasets to show the effectiveness of CIAT.

### 5.1 Datasets

We focused on two mainstream multilingual cases: many-to-English and English-to-many, since the many-to-many case can be bridged via English as a pivot. We collected the following three datasets for our experiments:

**IWSLT**<sup>2</sup> is a small dataset from TED talks, where we used 8 languages  $\leftrightarrow$  English from year 2014 to 2016 release.

**OPUS-100** (Zhang et al., 2020)<sup>3</sup> is an English-centric dataset covering 99 languages  $\leftrightarrow$  English pairs. We selected 20 language pairs, 17 of which have 1 million data samples while 3 language pairs are under low resource setting.

**WMT** (Barrault et al., 2019) datasets are also involved, which contains five language pairs ranging from the year 2014 to 2019.

For simplicity, we use the ISO 639-1 code as the abbreviation for language names. The detailed data statistics are listed in the Appendix.

<sup>1</sup>Note that Bapna and Firat (2019) only fine-tune their model on high-resource pairs, while we find CIAT can be applied to both high-resource and low-resource pairs.

<sup>2</sup><https://wit3.fbk.eu>

<sup>3</sup><http://opus.nlpl.eu/OPUS-100.php>

### 5.2 Implementation Details

For each dataset, we tokenize sentences using SentencePiece (Kudo and Richardson, 2018) jointly learned on the source and target side, and we set vocabulary size to 32,000. For model setup, we follow the same configuration as Tan et al. (2018) on IWSLT, including 2 layers for both encoder and decoder. The embedding dimension was 256, and the size of feed-forward hidden units was 1,024. The attention head was set to 4 for both self-attention and cross-attention. For OPUS-100 and WMT, we follow the standard Transformer-Big setting (Vaswani et al., 2017), including 6 layers for encoder and decoder. The embedding dimension, feed-forward hidden size, and attention head were set to 1024, 4096, and 16, respectively. The hidden state’s dimension of the adapters’ inner layer are set to be half of the embedding size, *i.e.* 128 for IWSLT and 512 for OPUS-100 and WMT. We use Adam optimizer (Kingma and Ba, 2015) with the same schedule algorithm as Vaswani et al. (2017). During Inference, we use a beam width of 4 and length penalty of 0.6.

All our experiments are evaluated by tokenized BLEU (Papineni et al., 2002) using `multi-bleu.perl`<sup>4</sup>. We implement our models via TensorFlow (Abadi et al., 2016) and train models on NVIDIA Tesla V100 GPUs.

### 5.3 Main Results

We compared our model with several strong baselines and effective models:

**Bilingual:** The model is trained with only bilingual data with the same model configuration, which serves as the strong benchmark.

**Multilingual** (Johnson et al., 2017): The data of all language pairs are mixed to train the model.

**Knowledge Distillation (KD)** (Tan et al., 2018)<sup>5</sup>: The bilingual models are first trained as the teachers, then the multilingual models are trained as students. We only conduct KD on IWSLT due to computational resource limitations.

**Serial** (Bapna and Firat, 2019): We re-implement the series adapter model as illustrated in Fig 2a, which made the first attempt on applying adapters to machine translation.

<sup>4</sup><https://github.com/moses-smt/ Mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>

<sup>5</sup><https://github.com/RayeRen/multilingual-kd-pytorch>. Note that we remove the `lowercase` option in their preprocessing script.

	en-IWSLT		en-OPUS-100		en-WMT	
	←	→	←	→	←	→
Bilingual	32.42	23.44	31.58	28.13	29.03	26.92
Multilingual	31.56	20.69	35.75	29.59	30.06	25.47
KD	30.97	20.31	-	-	-	-
Serial	31.63	21.37	36.44	31.26	30.69	26.92
CIAT-basic	31.75	22.06	36.78	31.96	31.12	27.34
CIAT-block	32.31	22.13	36.89	32.05	31.14	<b>27.66</b>
CIAT	<b>32.39</b>	<b>22.48</b>	<b>36.91</b>	<b>32.07</b>	<b>31.24</b>	27.63
WR%	87.5	100	25	75	60	80

Table 1: Overall Performance. The overall score is the arithmetic mean of the case-sensitive tokenized BLEU score of the test set of all languages. WR% is the win ratio(%), which denotes the percentage of the language pairs in which CIAT exceeds all multilingual baselines by *at least* 0.5 BLEU.

To further analyze the impact of each component and compare the two designs of Serial and Parallel layer adapter under the same number of parameters, we introduce two model variants, named CIAT-layer and CIAT-block, and conduct the ablation study. CIAT-layer has the same number of parameters compared with Serial and CIAT-block removes embedding adapter. The ablation details are illustrated in the appendix.

We also reproduced a recently proposed adapter-based MT model, namely Mono (Philip et al., 2020). The model also follow a serial adapter connection manner, which we replace with a parallel connection to validate the effectiveness of our parallel design. We show the comprehensive results of the ablation study and parallel variant of Mono in the Appendix. We here list the overall results of two model variants.

We present the BLEU score of three datasets on Table 2, 3 and 7 (Appendix) respectively, and summarize the overall results on Table 1. Since the number of parameters is quite different among distinct baseline and model variants, we also indicate parameters amount for each model with respect to the number of translation directions. We present our findings as follows:

On the IWSLT dataset, the bilingual baselines are significantly better than the multilingual models, which align with our expectations. To our surprise, we observe the opposite phenomenon on OPUS-100, which may be because (1) the domain of sentences in the OPUS-100 dataset is close. (2) Most languages selected have relatives from the same language family, which leads to promotion between the related languages (Tan et al., 2019); while only *zh* has no similar language in the OPUS-

100, and bilingual performs better in *zh*.

Among various models, the gaps in BLEU scores of *any*→*en* are smaller than that of *en*→*any*. Meanwhile, *any*→*en* direction gain less improvement from CIAT and other baselines than *en*→*any* direction. We attribute such phenomenon to the over-representation of English in the English-centric corpus (Aharoni et al., 2019), so it is more difficult to improve the generation quality of English than other languages.

CIAT significantly outperforms other multilingual competitors among all datasets, and improve the BLEU score by at least 0.5 in 42 out of 64 language directions. On *en*→*any* directions, the performance improvement is even more significant.

We also find that the performance of CIAT and baselines on different languages is also affected by language families and resource scarcity. According to Table 2, for languages of the same language family, CIAT can better improve performance (e.g., Spanish, Portuguese, French, and Italian get a greater improvement than Arabic and Persian). And for low-resource languages(e.g. Afrikaans, Belarusian), compared with Serial baseline, CIAT can bring more BLEU score improvement, especially when there are languages similar to these low-resources in the training set.

## 5.4 Discussion on Adapters

To obtain a comprehensive understanding of the adapters in CIAT, we conduct a series of analyses on embedding adapters and layer adapters respectively.

### 5.4.1 Semantic Alignment of Embedding Adapters

To study whether the embedding adapter alleviates multilingual embedding conflation, we calculate the Average Cosine Similarity(ACS) (Lin et al., 2020) of words with the same meaning across different languages to verify if embedding adapter help to align cross-lingual synonyms. We select top frequent 1000 words of five language pairs from MUSE bilingual dictionaries<sup>6</sup>, and compare the ACS results between vanilla multilingual embedding and ones with the CIAT’s embedding adapter, where the models are trained on OPUS-100 dataset.

We plot the ACS results in Fig. 3 and observe two phenomena. Firstly, adding the embedding adapters increases ACS of the model among all

<sup>6</sup><https://github.com/facebookresearch/MUSE>

	params(M)	en-ar		en-fa		en-de		en-nl		en-af		en-da			
		←	→	←	→	←	→	←	→	←	→	←	→		
Bilingual	$242 \times N$	37.45	23.11	22.10	9.92	33.22	30.21	30.24	26.96	46.63	44.79	35.80	35.32		
Multilingual	242	39.94	22.92	24.34	10.29	34.55	29.67	33.73	28.59	53.31	45.67	38.12	36.21		
Serial	$242 + 12.6 \times N$	41.20	25.00	25.63	10.21	35.53	31.78	34.16	29.88	55.70	50.92	38.50	37.58		
CIAT	$242 + 27.3 \times N$	<b>41.58</b>	<b>25.34</b>	<b>25.77</b>	<b>11.03</b>	<b>35.89</b>	<b>32.76</b>	<b>34.53</b>	<b>30.25</b>	<b>56.02</b>	<b>52.17</b>	<b>39.38</b>	<b>39.02</b>		
		en-no		en-sv		en-nb		en-es		en-pt		en-fr		en-it	
		←	→	←	→	←	→	←	→	←	→	←	→	←	→
Bilingual		25.20	27.31	30.63	30.56	32.36	29.93	36.78	36.52	34.82	31.56	33.83	35.11	33.28	30.54
Multilingual		27.92	29.19	32.88	33.03	43.48	35.52	41.01	37.60	38.37	33.03	35.33	34.20	37.22	31.57
Serial		28.21	30.24	33.74	34.15	45.17	39.38	41.34	39.32	38.96	34.59	36.41	36.02	37.82	32.94
CIAT		<b>28.43</b>	<b>31.67</b>	<b>34.39</b>	<b>34.78</b>	<b>45.82</b>	<b>40.46</b>	<b>41.73</b>	<b>40.12</b>	<b>39.16</b>	<b>35.38</b>	<b>36.49</b>	<b>36.81</b>	<b>37.88</b>	<b>33.53</b>
		en-pl		en-cs		en-sk		en-ru		en-uk		en-be		en-zh	
		←	→	←	→	←	→	←	→	←	→	←	→	←	→
Bilingual		24.39	20.56	31.63	25.22	36.84	30.58	32.75	29.30	24.26	16.15	7.72	7.71	41.74	41.29
Multilingual		28.16	21.94	36.46	26.51	40.02	31.48	35.207	29.07	28.97	16.43	25.22	<b>19.85</b>	40.72	38.98
Serial		28.63	23.78	36.51	28.59	41.19	33.26	35.78	<b>31.05</b>	29.26	18.65	23.04	16.89	42.10	40.92
CIAT		<b>28.88</b>	<b>24.42</b>	<b>36.77</b>	<b>28.70</b>	<b>41.70</b>	<b>33.89</b>	<b>35.93</b>	30.41	<b>29.68</b>	<b>19.56</b>	<b>26.00</b>	19.56	<b>42.13</b>	<b>41.54</b>

Table 2: Case-sensitive tokenized BLEU on the OPUS-100 dataset. params(M) represents the total number of parameters required for each model in million, where  $N$  is the number of language pairs.

	params(M)	en-es		en-de		en-et		en-ru		en-lv	
		←	→	←	→	←	→	←	→	←	→
Bilingual	$242 \times N$	35.05	34.45	33.07	28.37	24.27	19.20	34.42	35.69	18.34	16.89
Multilingual	242	34.92	33.74	32.53	26.44	26.85	18.14	35.37	33.14	20.65	15.91
Serial	$242 + 12.6 \times N$	35.47	33.78	33.38	26.88	27.76	20.30	35.86	35.11	20.97	18.54
CIAT	$242 + 27.3 \times N$	<b>35.75</b>	<b>34.51</b>	<b>33.62</b>	<b>27.78</b>	<b>28.33</b>	<b>20.97</b>	<b>36.9</b>	<b>35.97</b>	<b>21.59</b>	<b>18.94</b>

Table 3: Case-sensitive tokenized BLEU on the WMT dataset. params(M) represents the total number of parameters required for each model in million, where  $N$  is the number of language pairs.

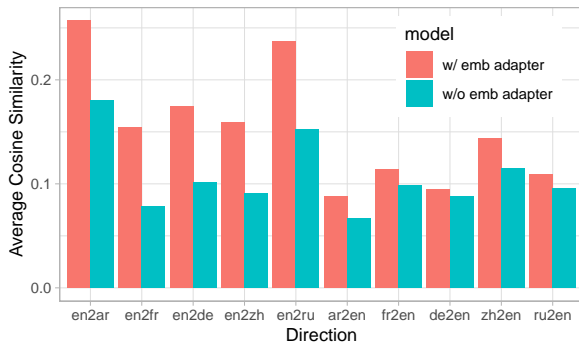


Figure 3: Average cosine similarity between vanilla multilingual embedding (w/o emb adapter) and the adapted ones(w/ emb adapter). The ACS increases after applying the embedding adapter among all selected languages, indicating that embedding adapter reduce the embedding distance between cross-lingual synonyms.

selected pairs, which indicates that embedding adapters capture more semantic information between synonyms from different languages. The ACS improvement also suggests that embedding adapters indeed relieve the multilingual embedding conflation problem as the latent representation of synonyms between the two languages is brought closer with the auxiliary of the embedding adapter, which improves the overall performance. Secondly, compared to *any*  $\rightarrow$  *en* directions, *en*  $\rightarrow$  *any* pairs generally gain more improvement on ACS and BLEU scores, indicating embedding adapter is more effective on translating pivot lan-

guage to other languages compared to the opposite direction.

#### 5.4.2 The Influence of Layer Adapters

We also perform an extension experiment to discover the influence of layer adapters on the main model, taking *en* $\leftrightarrow$ *zh* directions from OPUS-100 as the study cases.

We first plot the L2-norm ratio of the hidden state between the adapter and the base model across layers in Fig 4 to investigate how the “plug-in” adapter influence the multilingual model. In general, the adapters of decoder exerts a greater influence on the hidden states of the base model compared to the ones of the encoder, and the adapters provide a stronger signal in *en* $\rightarrow$ *zh* compared to the opposite direction.

We further examine these adapters’ impact by re-evaluating the trained model with certain adapters from continuous layer spans removed, and we illustrate the BLEU score drop on Fig. 5. We find that removing the adapters on the decoder side raises a greater performance decline, consistent with the trend of the L2-norm ratio. Adapters of the *en* $\rightarrow$ *zh* are more crucial to the multilingual models, which is in line with our experiments that adapters are more important for *en*  $\rightarrow$  *any* directions (Table 1). In addition, the decoder’s upper layers of the CIAT layer adapter have bigger impacts on the performance, consistent with Houlshy et al. (2019)’s findings on adapters for BERT model.

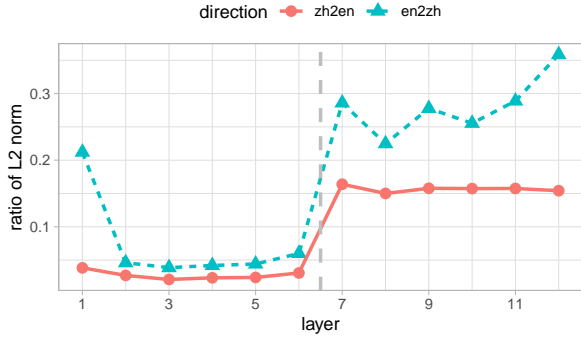


Figure 4: Hidden states L2-norm ratio of layer adapter over the base model in each layer. The first six layers are from the encoder, and the rests are from the decoder. The adapters of decoder influence more to the base model compared to the encoder ones.

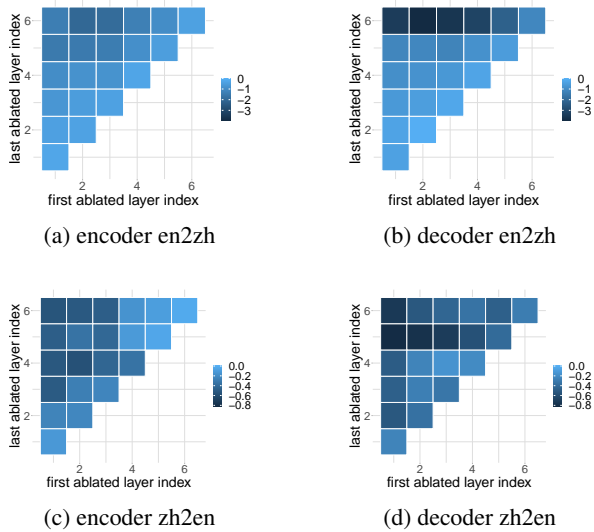


Figure 5: Performance of CIAT with ablated layers. The heat map shows the relative performance decrease as continuous layer spans are removed directly from the CIAT model. The  $x$  and  $y$  axis indicate the index number of the first and the last layer removed. The layer adapters from the higher layer of the decoder side exert the most impact on the model when they are disabled.

### 5.5 Parameter-Performance Trade-off

The bottle-neck adapter design utilizes a small middle layer to control parameter efficiency (Houlsby et al., 2019), while empirically, a larger layer dimension improves the performance via increased capacity. We explore the parameter-performance trade-off by varying the dimension of adapters and illustrate the BLEU score over different layer adapter and embedding adapter size in Fig. 6 with different color, respectively. Here we discuss the trade-off on  $en \leftrightarrow de$  directions of IWSLT. The plot shows that the dimension of the layer adapters has a significant impact to the performance: as the dimen-

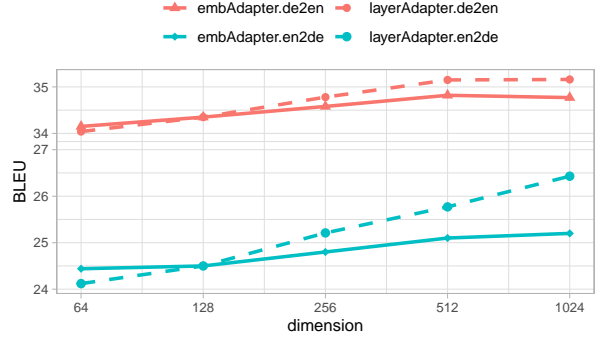


Figure 6: BLEU score vs. CIAT dimension on IWSLT en-de. The dashed line indicates the layer adapter, the solid line indicates the embedding adapter, and the color represents translation direction. The BLEU score is more sensitive to the dimension of layer adapters. Note part of the y-axis is truncated.

sion doubles, the BLEU increases 0.58 and 0.28 by an average in  $en \rightarrow de$  and  $de \rightarrow en$  directions respectively. In comparison, changing the embedding adapters’ dimension impacts less on the final performance. When the dimension is large already, expanding the size hardly increases the BLEU score. Considering that the parameter amounts of CIAT are small compared to the base model, and the final performance is sensitive to the layer adapter’s dimension, we regard expanding dimension to be regarded as a simple and effective way to improve the performance of CIAT.

## 6 Conclusion

This work analyzes the performance degradation problem in multilingual NMT systems and decomposes into multilingual embedding deficiency and multilingual interference effects. We then propose a novel framework to deal with degradation, named Counter-interference Adapter for Multilingual machine translation (CIAT). CIAT alleviates two issues above respectively by introducing two kinds of adapters.

We validate the effectiveness of CIAT on three multilingual translation datasets, where the results show that CIAT improves the performance of the multilingual NMT model on various translation directions. The experiments also demonstrate that CIAT variants outperform several strong baselines, approving our analysis and framework design. Furthermore, we investigate the behavior and utility of each component via empirical studies.



## References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.
- Roe Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- 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.
- Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.
- Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788.
- 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.
- Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of NAACL-HLT*, pages 866–875.
- Yuqing Gao, Bowen Zhou, Zijian Diao, Jeffrey Sorensen, and Michael Picheny. 2002. Mars: A statistical semantic parsing and generation-based multilingual automatic translation system. *Machine Translation*, 17(3):185–212.
- Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354.
- Gholamreza Haffari and Anoop Sarkar. 2009. Active learning for multilingual statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 181–189.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.
- 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.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663.

- Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10.
- Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Language adapters for zero shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516.
- Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. 2015. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1379–1390.
- Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Xu Chen, Sneha Reddy Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2827–2835. Association for Computational Linguistics.
- Hagai Taitelbaum, Gal Chechik, and Jacob Goldberger. 2019. Multilingual word translation using auxiliary languages. 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 1330–1335.
- Xu Tan, Jiale Chen, Di He, Yingce Xia, QIN Tao, 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 962–972.
- Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2018. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.
- Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 193–199.
- 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 *Advances in Neural Information Processing Systems*, pages 5998–6008.
- 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.
- Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.
- Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796.
- Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *Ninth 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.
- Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655.

## Appendix

### 6.1 Detailed Dataset Statistics

We give the detailed statistics about the dataset used in Sec. 5

**IWSLT.** We almost follow the Tan et al. (2018)’s script<sup>7</sup> except that we removed their lowercase option. We collect training sets of 8 languages ranging from the year 2014 to 2016 and use the official valid/test set. We list the number of samples in training set in Table 4.

ar	de	es	fa	he	it	nl	pl
140k	160k	169k	89k	144k	167k	153k	128k

Table 4: The number of training pairs of IWSLT

**OPUS-100.** We collect data from Zhang et al. (2020)’s release<sup>8</sup>, and use its official valid/test set. Among the 20 language pairs selected, 17 have 1 million training samples while three language pairs are of low resources, which are be(67k), nb(142k), and af(275k).

**WMT.** We list the year of the training, valid and test set of each language in Table 5. Table 6 illustrate the number of samples in the training set.

	es	de	ru	et	lv
training	2013	2016	2016	2018	2017
valid	2012	2013	2019	OPUS	OPUS
test	2013	2014	2020	2018	2017

Table 5: The year of training, valid test set of WMT datasets. The OPUS denotes we use the valid set from OPUS-100 dataset.

es	de	ru	et	lv
15.18M	4.56M	2.59M	2.18M	4.51M

Table 6: The number of training pairs of WMT

### 6.2 Detailed Experiment Results of Ablation Study and Model Variants

To further study the efficacy of each component in CIAT, we propose two variants as ablation study: **CIAT-layer:** This variant keeps only the layer adapter and removes all the embedding adapter.

<sup>7</sup><https://github.com/RayeRen/multilingual-kd-pytorch/blob/master/data/iwslt/raw/prepare-iwslt14.sh>

<sup>8</sup><https://object.pouta.csc.fi/OPUS-100/v1.0/opus-100-corpus-v1.0.tar.gz>

**CIAT-basic:** Besides removing all embedding adapters, this variant introduces only one layer adapter for each attention block. We design this variant to compare with Serial (Bapna and Firat, 2019) under a similar amount of parameters to determine the effectiveness of our parallel connection further.

We present the detailed results on two CIAT variants on Table 7, 8 and 9 respectively. We give two major findings of ablation study: *a)* Compared with CIAT, CIAT-layer suffers from degradation in most language pairs (51 out of 66), especially in low-resource corpora(IWSLT). It further shows the effectiveness of the embedding adapter. *b)* With the same amount of parameters, CIAT-basic surpass Serial in both overall performance and the number of improved pairs among all datasets. It shows that parallel is a more suitable adapter connection schema for multilingual machine translation.

### 6.3 The Effectiveness of Parallel Connection on Other Adapter Model

As mentioned in Section 5, we also substitute another adapter-based model with parallel connection methods to illustrate our proposed parallel connection is more suitable for multilingual machine translation. We re-implement Philip et al. (2020)’s work, where they proposed monolingual adapter which is specific to the source/target language other than the translation pair. We present the results of their serial connection and our parallel variants in Tab. 7 as Mono-Serial and Mono-Parallel.

We find parallel connection boosts the Mono adapter in 15 out of 16 language pairs in the IWSLT data set, the improvement is even more prominent in the *en*  $\rightarrow$  *any* ones. The results further prove that our parallel layer adapter can provide improvement for all multilingual adapter models.

### 6.4 Case Study

We also conduct qualitative analysis by case study. We invite two German speakers to compare the translation of a news report from the WMT test set. The contestants are generated by CIAT, Serial and vanilla multilingual. We find German translation of CIAT are more favored by human annotators for the following reasons: *a)* The sentence tense and clause pattern fit the original sentence; *b)* CIAT tend to use set phrases instead of simple expressions; *c)* CIAT better captures the relationship between modifiers and the subjects.

We list a sample translation in Tab 10.

	params	en-ar		en-de		en-es		en-fa		en-he		en-it		en-nl		en-pl		
		←	→	←	→	←	→	←	→	←	→	←	→	←	→	←	→	
Bilingual	$20M \times N$	31.35	13.82	34.65	25.81	40.62	35.41	22.76	11.90	36.78	25.16	34.13	30.72	36.65	30.18	22.44	14.51	
Multilingual	20M	30.01	12.09	33.54	22.40	39.56	32.18	22.01	11.15	34.54	20.35	33.98	27.88	35.72	27.36	23.15	12.07	
	KD	0	30.03	12.31	32.03	22.55	38.06	30.78	22.12	10.51	33.32	20.69	32.93	25.65	35.84	27.15	23.45	12.80
	Serial	265k	30.05	12.52	33.65	23.48	39.57	33.02	22.21	11.90	34.59	20.37	33.54	28.80	35.86	28.19	23.57	12.66
CIAT-basic	$20M + 264k \times N$	30.11	12.48	33.46	24.12	39.74	33.73	22.22	12.48	34.82	22.28	34.14	29.32	36.01	28.90	23.50	13.22	
CIAT-layer	$20M + 528k \times N$	30.51	12.99	34.29	24.02	<b>40.48</b>	33.49	22.63	12.59	<b>35.20</b>	22.55	<b>34.73</b>	29.34	36.55	28.80	<b>24.13</b>	13.24	
CIAT	$20M + 660k \times N$	<b>30.74</b>	<b>13.31</b>	<b>34.35</b>	<b>24.5</b>	40.13	<b>34.16</b>	<b>23.64</b>	<b>12.73</b>	35.12	<b>22.7</b>	34.50	<b>29.74</b>	<b>36.67</b>	<b>29.33</b>	23.96	<b>13.41</b>	
Mono-Serial	$20M + 132k \times L$	30.25	11.14	33.97	22.66	39.27	32.26	22.46	10.79	33.86	20.76	33.53	28.11	36.45	27.46	23.04	12.14	
Mono-Parallel	$20M + 132k \times L$	30.69	12.40	33.93	23.42	39.83	33.15	22.74	11.56	34.35	20.82	34.17	28.84	36.50	28.30	23.68	12.61	

Table 7: Full results of Case-sensitive tokenized BLEU on the IWSLT dataset. params(M) represents the total number of parameters required for each model in million, where  $N$  is the number of language pairs, and  $L$  is the total number of languages.

	$\Delta$ params	en-ar		en-fa		en-de		en-nl		en-af		en-da			
		←	→	←	→	←	→	←	→	←	→	←	→		
CIAT-basic	12.6M	40.94	25.21	25.58	10.94	35.79	32.09	34.13	30.38	56.45	52.55	39.12	38.13		
CIAT-layer	25.2M	41.11	25.13	25.83	10.96	35.92	32.27	34.23	30.33	56.18	52.46	39.19	38.21		
		en-no		en-sv		en-nb		en-es		en-pt		en-fr		en-it	
		←	→	←	→	←	→	←	→	←	→	←	→	←	→
CIAT-basic	28.63	31.03	33.85	34.59	45.61	39.88	41.86	39.65	39.36	34.96	36.35	36.28	37.95	33.41	
CIAT-layer	28.58	31.08	33.98	34.82	45.89	40.11	41.8	39.72	39.63	34.75	36.37	36.72	37.81	33.11	
		en-pl		en-cs		en-sk		en-ru		en-uk		en-be		en-zh	
		←	→	←	→	←	→	←	→	←	→	←	→	←	→
CIAT-basic	28.79	23.84	36.39	29.05	41.18	34.08	35.96	31.82	29.88	19.5	25.58	20.19	42.29	41.74	
CIAT-layer	28.46	24.44	37.06	29.22	41.14	33.92	36.21	31.77	29.86	19.41	25.94	20.85	42.51	41.62	

Table 8: Case-sensitive tokenized BLEU score on the OPUS-100 dataset of two model variants.

	$\Delta$ params	en-es		en-de		en-et		en-ru		en-lv	
		←	→	←	→	←	→	←	→	←	→
CIAT-basic	12.6M	35.79	34.25	33.62	27.36	28.11	20.68	36.62	35.96	21.45	18.45
CIAT-layer	25.2M	35.70	34.40	33.65	27.85	28.21	20.91	36.55	36.22	21.58	18.91

Table 9: Case-sensitive tokenized BLEU score on the WMT dataset of two model variants.

Source	The Kluser lights protect cyclists, as well as those travelling by bus and the residents of Bergle.
mTransformer	Die Kluser-Lichter schützen Radfahrer, Busfahrer und Bergleiter.
Serial Adapter	Die Lichter von Kluser schützen Radfahrer, aber auch Busreisende und die Bewohner von Bergle.
CIAT	Die Kluser-Leuchten schützen Radfahrer, Busfahrer und Einwohner von Bergle.
Human	Die Kluser-Ampel sichere sowohl Radfahrer als auch Busfahrgäste und die Bergle-Bewohner.

Table 10: Sampled translation of CIAT and baselines. We also list human reference here for comparison. The sample shows that the CIAT’s translation is more accurate and smooth.