Knowledge Transfer in Incremental Learning for Multilingual Neural Machine Translation

Kaiyu Huang[1], **Peng Li**[*1,4], **Jin Ma**[5,6], **Ting Yao**[5], **Yang Liu**[* 1,2,3,4]
[1]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[2]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[3]Beijing National Research Center for Information Science and Technology
[4]Shanghai Artificial Intelligence Laboratory, Shanghai, China [5]Tencent
[6]Sch. of Comp. Sci. & Tech., University of Science and Technology of China
{huangkaiyu,lipeng}@air.tsinghua.edu.cn; tessieyao@tencent.com
majin01@mail.ustc.edu.cn; liuyang2011@tsinghua.edu.cn

Abstract

In the real-world scenario, a longstanding goal of multilingual neural machine translation (MNMT) is that a single model can incrementally adapt to new language pairs without accessing previous training data. In this scenario, previous studies concentrate on overcoming catastrophic forgetting while lacking encouragement to learn new knowledge from incremental language pairs, especially when the incremental language is not related to the set of original languages. To better acquire new knowledge, we propose a knowledge transfer method that can efficiently adapt original MNMT models to diverse incremental language pairs. The method flexibly introduces the knowledge from an external model into original models, which encourages the models to learn new language pairs, completing the procedure of knowledge transfer. Moreover, all original parameters are frozen to ensure that translation qualities on original language pairs are not degraded. Experimental results show that our method can learn new knowledge from diverse language pairs incrementally meanwhile maintaining performance on original language pairs, outperforming various strong baselines in incremental learning for MNMT.[1]

1 Introduction

Multilingual neural machine translation (MNMT) aims at handling multiple translation directions in a single model and achieves great success in recent years (Wenzek et al., 2021; Goyal et al., 2022; Cheng et al., 2022). However, the powerful MNMT models need to be retrained from scratch using a mixture of original and incremental training data when new language pairs arrive (Tang et al., 2020). Considering that the original training data of MNMT models is often large-scale (Fan et al., 2021; Costa-jussà et al., 2022), and thus the method that utilizes original data to train incrementally is time-consuming and cumbersome (Ebrahimi and Kann, 2021). Therefore, a practical scenario is that these models can continually support new language pairs while preserving the previously learned knowledge without accessing previous training data, which belongs to incremental learning, and has drawn much attention recently (Dabre et al., 2020; Gu et al., 2022; Zhao et al., 2022).

Method	*en*→*uk*	*en*→*bn*
From-scratch	23.70	15.54
Replay-Based	21.37	10.32
Regularization-Based	20.33	7.84
Adapter	19.92	9.46

Table 1: The BLEU scores of incremental learning methods on new translation directions. The original model is mBART25 which does not support the languages of Ukrainian (*uk*) and Bengali (*bn*).

In this scenario, existing studies attempt to overcome the issue of catastrophic forgetting (French, 1993) on original language pairs, such as replay-based methods (Garcia et al., 2021; Liu et al., 2021) and regularization-based methods (Huang et al., 2022; Zhao et al., 2022). However, the methods primarily focus on balancing performance between old and new translation directions and use only incremental data to acquire new knowledge, restricting the development of new language pairs (Escolano et al., 2021; Ke and Liu, 2022). As shown in Table 1, prior incremental learning methods cannot

*Corresponding authors: Peng Li (lipeng@air.tsinghua.edu.cn) and Yang Liu (liuyang2011@tsinghua.edu.cn)
[1]Our code will be released at https://github.com/THUNLP-MT/ktnmt

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15286–15304

achieve comparable performance on new translation directions, compared with training a bilingual translation model from scratch. Therefore, is it possible to leverage external knowledge without increasing the amount of incremental training data to facilitate the learning procedure of new language adaptation?

Fortunately, the development of Transformer-based language models unveils that Feed-Forward Networks (FFN) might be a core component that stores the knowledge (Geva et al., 2021; Dai et al., 2022; Geva et al., 2022; Vázquez et al., 2022). In these efforts, external knowledge is injected into FFN layers to enhance the performance of pre-trained language models. More importantly, it opens the door to leveraging the knowledge from neural models in adapting MNMT models to incremental language pairs.

In this work, considering the knowledge in neural networks, we propose a **k**nowledge **t**ransfer (KT) method that can efficiently adapt MNMT models to diverse incremental language pairs. First, we convert incremental training data into continuous representation by additional parameters, forming pluggable modules. Then the pluggable modules can be flexibly introduced in the embedding layer and FFN layers of original MNMT models, respectively. The two stages are regarded as a process of knowledge transfer and equip original models with knowledge of unseen languages before adaptation, alleviating the representation gap between original models and introduced parameters. And the knowledge transfer method can further provide better optimization than training from scratch for the introduced parameters. Moreover, except for the pluggable modules, all the parameters of the original model are frozen. Therefore, our architecture can also retain previously learned knowledge from the original translation model to completely maintain the translation qualities on original language pairs. To sum up, our contributions are as follows:

- We propose a knowledge transfer method with pluggable modules to acquire more knowledge of new languages, which achieves competitive translation qualities on incremental language pairs.
- Our architecture can efficiently adapt to diverse language pairs in incremental learning and naturally retain the performance on original language pairs when the original training data is not available.
- Experiments show that our method can learn knowledge from the other large-scale translation models for adapting original models with different sizes to new language pairs.

2 Related Work

Replay-Based Methods. The first branch of works utilizes previous training data or create pseudo data that is essentially a replay on old tasks (de Masson D'Autume et al., 2019; Liu et al., 2021; Kanwatchara et al., 2021). Specifically, previous data sometimes cannot be accessed due to data protection and security (Feyisetan et al., 2020; Qu et al., 2021). In this scenario, Sun et al. (2019) replay pseudo samples of previous tasks, which can avoid forgetting previously learned knowledge. However, the pseudo data with noise significantly hurts the performance for both old and new tasks, and the data generation procedure requires additional time costs, restricting the efficiency of incremental learning (Peng et al., 2020; Garcia et al., 2021). In contrast to these methods, our approach does not require extra data and is more flexible in the real-world scenario.

Regularization-Based Methods. The second branch of works introduces additional penalty terms to the learning objective on the parameters, alleviating the issue of catastrophic forgetting (Kirkpatrick et al., 2017; Thompson et al., 2019; Castellucci et al., 2021; Gu et al., 2022). In particular, Shao and Feng (2022) propose a complementary online knowledge distillation method that utilizes previous models (teachers) to supplement current model (student) training. In contrast to these methods, our architecture can naturally avoid forgetting previously learned knowledge and retain the performance of old tasks. It allows our method to focus solely on learning new knowledge better instead of preserving old knowledge.

Parameter-Isolation Based Methods. The third branch of works introduces extra parameters to support new tasks and freeze all original parameters to completely retain the performance on previous tasks (Bapna and Firat, 2019; Madotto et al., 2021; Zhu et al., 2021). However, the methods only utilize incremental training data to optimize the additional parameters which are randomly initialized (Escolano et al., 2021; He et al., 2021),

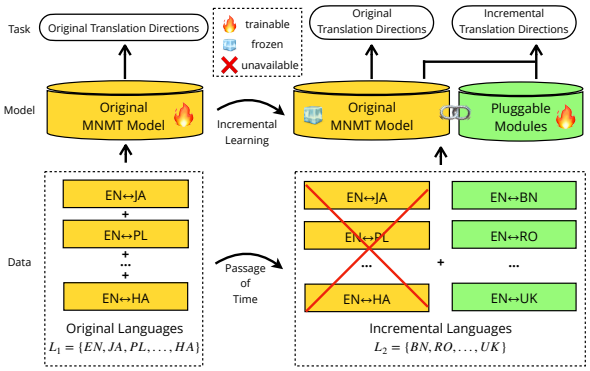


Figure 1: Illustration of incremental learning for MNMT. The green parts are trainable for incremental adaptation. The parameters of the original model are frozen at the stage of incremental learning.

hindering old models from learning knowledge of incremental languages (Dabre et al., 2020; Ke and Liu, 2022). Chalkidis et al. (2021) combine pseudo data with the prompt-tuning method to alleviate this issue for multilingual tasks. Our method attempts to exploit the potentiality of incremental training data to acquire new knowledge via knowledge transfer while not leveraging extra data, compared with existing methods.

3 Method

In this work, we aim to completely maintain the performance of previous translation tasks without original training data. As shown in Figure 1, we introduce additional components for new language pairs and adopt a strategy that does not disturb the parameters of the original model. We hope to minimize the impact on the original model during the incremental learning process. As a result, we exclusively concentrate on how to handle the situation of learning new language pairs. Furthermore, the additional components are transferred from parameters in another pre-trained translation model, in a similar way to a pluggable module, rather than randomly initialized, as shown in Figure 2. It can also reduce the cost of learning new language pairs during the training stage, enhancing the practicability and efficiency of incremental learning methods in the real-world scenario.

3.1 Task Definition

An ideal requirement is that original MNMT models can be continually updated to support new language pairs while retaining translation qualities on original language pairs without accessing previous training data, as shown in Figure 1.

Formally, an MNMT model is trained on initially selecting a set of available parallel data $\mathcal{D} = \{\mathcal{D}_1, ..., \mathcal{D}_i, ..., \mathcal{D}_N\}$ which covers N languages, and $\mathcal{D}_i$ represents the original parallel training corpus on the i-th language pair. The training objective of the initial MNMT model is to maximize the log-likelihood $\mathcal{L}$:

$$\mathcal{L}_{\mathcal{D}}(\theta) = \sum_{\mathcal{D}_i \in \mathcal{D}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_i} \log p(\mathbf{y}|\mathbf{x};\theta) \quad (1)$$

where θ represents the trainable parameters of MNMT models. The source sentence is denoted as $\mathbf{x}$, while the target sentence is denoted as $\mathbf{y}$. In order to specify the source and target languages, two language tokens are added at the beginning of each source and target sentence, respectively.

Incremental learning is updating the original MNMT model on an updated set of parallel data $\mathcal{D}^{(U)} = \{\mathcal{D}_1, ..., \mathcal{D}_N, ..., \mathcal{D}_M\}$ which covers M languages, and $N > M$. A dilemma, though, is that the original training data $\mathcal{D}$ is often unavailable due to data security. Thus, we can only utilize the new data $\mathcal{D}' = \{\mathcal{D}_{N+1}, ..., \mathcal{D}_j, ..., \mathcal{D}_M\}$ to incrementally train the original MNMT model, and $\mathcal{D}_j$ represents the incremental parallel training corpus on the j-th language pair. The optimization objective in incremental learning is given by:

$$\mathcal{L}_{\mathcal{D}'}(\theta) = \sum_{\mathcal{D}_j \in \mathcal{D}'} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_j} \log p(\mathbf{y}|\mathbf{x};\theta) \quad (2)$$

As a result, the number of language pairs that the MNMT model support increases from N to M.

3.2 Knowledge Transfer via Pluggable Modules

Based on the original MNMT model, we open up additional spaces for adapting to new language pairs, as shown in Figure 2. However, the additional spaces with initialized parameters are weak in their abilities to capture the shared linguistic features among different languages. Because the linguistic distribution of the original model is fitted previously, which leads to a representation gap between the original model and introduced spaces. Thus, we not only leverage new training data but also exploit the potentiality of these data by introducing two types of pluggable modules to bridge the representation gap.

Vocabulary Adaptation. On the one hand, if the new language has a distinct script with the

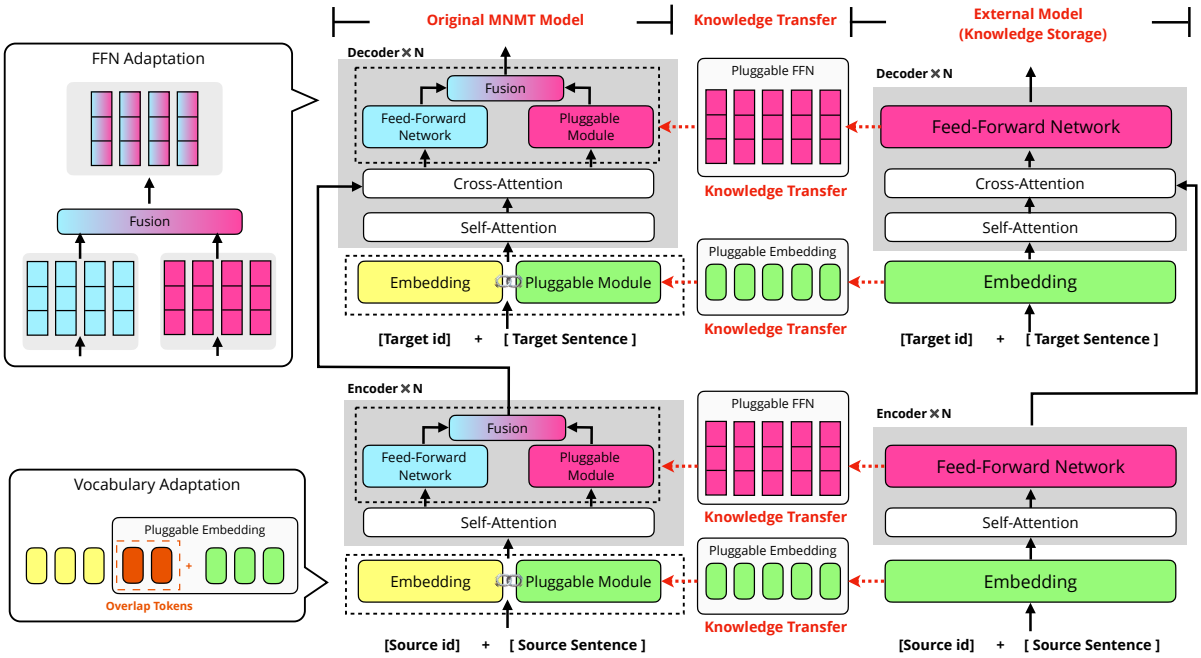


Figure 2: Illustration of our architecture. We extend two parts of space in the embedding layer and the FFN layers, respectively. These spaces are injected with pluggable modules that are pre-trained by another translation model. During the training stage, the parameters of the original model are frozen and the pluggable modules are trainable.

set of original languages, a certain proportion of the out-of-vocabulary (OOV) tokens with unclear semantics will occur due to different character sets between the original and incremental languages, which hinders performance on new language pairs (Zhang et al., 2022). However, the external model is not troubled by this situation and covers sufficient tokens for the incremental language pairs. Therefore, we expand an extra space in the embedding layer and concatenate the embeddings of non-overlap tokens between the original model and the external model, bridging the representation gap through vocabulary adaptation.

Feed-Forward Adaptation. On the other hand, FFN layers can be seen as key-value memories, which has previously been investigated by (Sukhbaatar et al., 2019). Each FFN layer consists of two linear networks with a non-linearity activation function and is given by:

$$\mathrm{FFN}(\mathbf{H}) = f(\mathbf{H} \cdot K^\top) \cdot V \tag{3}$$

where $K, V \in \mathbb{R}^{d_{\text{ffn}} \times d_{\text{model}}}$ are parameter matrices, d_{ffn} is the dimension of the FFN, $\mathbf{H}$ is the input hidden state of FFN layers and the dimension is d_{model}, f represents the non-linearity activation function, e.g., GELU and ReLU.

The first linear network is regarded as the keys and activates a set of intermediate neurons. The second linear network integrates the corresponding value matrices by weighted sum, using the activated neurons as weights. The FFN layers store knowledge in the key-value memories manner. Thus, we leverage the continuous representation in the FFN layers of the external model that stores useful language knowledge and transfer the knowledge into the FFN layers of the original model, forming a type of pluggable module. We combine the output of the original FFN layers and the injected pluggable modules to share linguistic knowledge, alleviating the representation gap through Feed-Forward layers adaptation. The fusion FFN output $\mathbf{H}^{(\mathrm{f})}$ is given by:

$$\mathbf{H}^{(\mathrm{f})} = \mathrm{FFN}_{\text{original}}(\mathbf{H}) + \mathrm{FFN}_{\text{external}}(\mathbf{H}) \tag{4}$$

3.3 Training and Inference

During the training stage, previously learned knowledge can be naturally preserved with a frozen training strategy, which can avoid the issue of catastrophic forgetting. The training procedure of our method is divided into two stages, as shown in Figure 2.

Stage 1: External Model Training. To convert incremental training data into continuous representation by additional parameters. We first leverage

the incremental training data to train an external Transformer-based neural network.

$$\mathcal{L}_{\mathcal{D}'}(\hat{\theta}) = \sum_{\mathcal{D}_j \in \mathcal{D}'} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_j} \log p(\mathbf{y}|\mathbf{x}; \hat{\theta}) \quad (5)$$

where $\hat{\theta}$ represents the trainable parameters of the external neural models. We only retain the parameters in the embedding layer ($\hat{\theta}_e$) and FFN layers ($\hat{\theta}_f$) of the external model as the pluggable modules for the next training stage.

Stage 2: Pluggable Module Tuning. Directly transferring the additional parameters limits the MNMT model capacity, especially for the language pairs with sufficient data. Therefore, we further train the pluggable modules in the second stage:

$$\mathcal{L}_{\mathcal{D}'}(\hat{\theta}_e, \hat{\theta}_f) = \sum_{\mathcal{D}_j \in \mathcal{D}'} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_j} \log p(\mathbf{y}|\mathbf{x}; \hat{\theta}_e, \hat{\theta}_f) \quad (6)$$

where $\hat{\theta}_e$ and $\hat{\theta}_f$ represent the trainable parameters of pluggable modules in the embedding layer and FFN layers, respectively.

Inference. For the inference stage, the original translation directions follow the original model without any pluggable modules while the incremental translation directions require the concatenation of the original model and pluggable modules.

4 Experiments

4.1 Datasets

To ensure the reliability of the experiments, the original MNMT model is implemented on a multilingual machine translation dataset[2] (WMT-7) that covers seven languages (Farhad et al., 2021). And we provide four incremental languages considered for incremental adaptation[3]. An extensive description and comprehensive information regarding the datasets for all languages can be found in Appendix A. All training data are sourced from the WMT (Workshop on Machine Translation) and FLoRes datasets, ensuring reliable quality.

Language Choice. As contrasted to previous studies for incrementally adapting translation models to new languages, we further provide a comprehensive language setting. Previous works often investigate the situation of the related languages which are similar language families and scripts to the original languages. In our setting, the incremental languages have distinct scripts and belong to several language families, which leads to a serious language representation gap. Please refer to Appendix A.2 for more details of language consideration in our setting.

4.2 Implementation Details

Baselines. We implement a vanilla Transformer for original languages as the initial model which is trained on multiple parallel data jointly (Johnson et al., 2017). And we compare the proposed method with different architectures for adapting the original model to new language pairs. All methods utilize the preprocessing script of a shared BPE model with 32k tokens based on the Sentencepiece library[4]. The baselines can be listed as follows:

From-scratch (Johnson et al., 2017): A vanilla Transformer is trained from scratch on the incremental languages with the multilingual training strategy. Note that the models do not support the original translation directions.

Adapter (Bapna and Firat, 2019): We follow previous adapter architectures and introduce extra parameters in each FFN layer of the original MNMT model. All original parameters are frozen and only the adapters are trainable.

Extension (Lakew et al., 2018): On the basis of the adapter architecture, we extend the original vocabulary ($\mathcal{V}_\mathcal{P}$) for new languages adaptation. Initially, a supplementary vocabulary ($\mathcal{V}_\mathcal{Q}$) is created using the standard Byte-Pair Encoding (BPE) procedure from the incremental training data. Subsequently, $\mathcal{V}_\mathcal{P}$ and $\mathcal{V}_\mathcal{Q}$ are combined to form a unified vocabulary $\mathcal{V}$, which is defined as $\mathcal{V} = \mathcal{V}_\mathcal{P} \cup \mathcal{V}_\mathcal{Q}$. The embeddings of the original models are expanded to match the size of the complete vocabulary ($\mathcal{V}$), and the additional embeddings are initialized using a Gaussian distribution.

Serial/Parallel (Zhu et al., 2021): We follow Zhu et al. (2021) to introduce adapters in the serial or parallel connection manner. Our pluggable modules in the FFN layers can also be converted into a serial manner.

Training Setup. We implement all models based on the open-source toolkit fairseq[5] (Ott et al., 2019). For a fair comparison, we employ the same configuration of Transformer-Big (Vaswani et al., 2017)

[2] https://www.statmt.org/
[3] https://data.statmt.org/cc-matrix/
[4] https://github.com/google/sentencepiece
[5] https://github.com/pytorch/fairseq

Method	Modules	*WMT16*		*WMT14*		*FLoRes*		*FLoRes*	
		ro→en	*en→ro*	*de→en*	*en→de*	*bn→en*	*en→bn*	*uk→en*	*en→uk*
From-scratch	-	30.63	23.58	31.35	26.55	30.58	15.54	28.22	23.70
Adapter	Serial	34.39	22.85	30.34	19.43	16.94	0.12	28.47	19.01
	Parallel	32.98	20.74	25.97	17.24	13.51	0.11	26.04	15.42
Extension	Serial	32.06	22.36	30.53	22.05	27.78	13.42	30.65	21.12
	Parallel	32.31	20.87	28.66	20.77	27.51	12.49	30.69	20.60
KT (Ours)	Serial	34.38	25.52	31.24	26.33	30.73	15.63	31.91	24.46
	Parallel	**34.44**	**25.62**	**32.04**	**26.73**	**30.81**	**15.71**	**32.01**	**25.20**

Table 2: Results in BLEU of adding a single language pair merely for MNMT in incremental learning. The highest score on each translation direction is highlighted in **bold**. The second highest score on each translation direction is highlighted in underline.

Method	Modules	*WMT16*		*WMT14*		*FLoRes*		*FLoRes*	
		ro→en	*en→ro*	*de→en*	*en→de*	*bn→en*	*en→bn*	*uk→en*	*en→uk*
From-scratch	-	35.42	25.90	31.33	25.34	30.40	14.66	32.71	25.05
Adapter	Serial	31.02	15.11	24.34	7.86	10.04	0.54	23.19	9.91
	Parallel	35.38	14.58	29.57	6.56	24.95	0.34	31.52	9.40
Extension	Serial	32.17	19.88	26.90	15.94	24.40	9.88	27.56	14.33
	Parallel	36.28	23.17	30.45	20.92	28.78	11.62	31.94	19.59
KT (Ours)	Serial	36.34	25.53	30.96	24.52	29.82	12.95	32.95	24.31
	Parallel	**36.88**	**26.48**	**31.65**	**25.65**	**30.10**	**15.10**	**33.86**	**25.42**

Table 3: Results in BLEU of adding eight incremental language pairs for xx-to-English and English-to-xx simultaneously. The highest score on each translation direction is highlighted in **bold**. The second highest score on each translation direction is highlighted in underline.

in our experiments. All original parameters are frozen during the incremental learning procedure. More model training details are provided in Appendix B.1.

Evaluation. We report the detokenized case-sensitive BLEU of models by the SacreBLEU evaluation script (Post, 2018)[6]. We show the training time of each method in terms of kiloseconds and use the beam search decoding algorithm with a beam size of 5 and a length penalty of 1.0.

4.3 Main Results

Adding A Single Language

As shown in Table 2, we investigate the translation qualities when a new language pair arrive. The results demonstrate that our proposed method (KT) outperforms several baselines in terms of average BLEU scores for all incremental translation directions. Specifically, KT achieves an average BLEU score of 27.14 for xx→en and 21.32 for en→xx translations. In particular, based on KT, the pluggable modules that are injected in a parallel manner further improve the performance over the serial manner on all incremental language pairs.

[6]Signature: nrefs:1 | case:mixed | eff:no | tok:13a | smooth:exp | version:2.2.0.

Method	*WMT-7* (ΔBLEU)		*WMT16* (BLEU)	
	xx→en	*en→xx*	*ro→en*	*en→ro*
From-scratch	-	-	30.63	23.58
Fine-tuning	-18.51	-14.00	33.87	25.22
Replay	-0.74	-1.24	29.33	19.00
Replay†	-3.86	-2.28	27.51	21.93
EWC	-0.99	-2.82	28.14	17.87
Self-KD	-5.29	-8.33	29.55	19.47
LFR	-1.01	-2.56	32.13	22.73
Prompt	-0.00	-0.00	31.33	15.64
Prompt†	-0.00	-0.00	33.21	22.77
Prefix	-0.00	-0.00	32.71	18.74
Prefix†	-0.00	-0.00	33.17	22.62
KT (Ours)	**-0.00**	**-0.00**	**34.44**	**25.62**

Table 4: Results on the original (WMT-7) and incremental language pairs with different continual learning methods. "†" indicates that we further extend the embedding layers based on each method. The highest score is highlighted in **bold**.

The Adapter methods are more vulnerable to adapting original models to some incremental language pairs, e.g., 0.12 BLEU scores on *en→bn* and 16.94 BLEU scores on *bn→en*. Because the methods with an unaltered vocabulary result in the sentence being broken up into semantically meaningless OOV tokens. Although the Extension methods can alleviate the issue of OOV tokens and fragmentary semantics by rebuilding embedding layers, the

No.	Method	Model Size		Incremental Language Pairs (BLEU)			
		Original	External	*ro*→*en*	*en*→*ro*	*uk*→*en*	*en*→*uk*
1	KD	0.4B	0.4B	32.68	24.36	30.67	23.48
2	KD	0.4B	1.2B	33.98	24.33	31.56	24.76
3	KT	0.4B	0.4B	34.15	24.54	32.33	24.93
4	KT	0.4B	1.2B	34.37	24.74	32.77	25.53
5	KT+KD	0.4B	0.4B	33.31	25.72	32.49	25.18
6	KT+KD	0.4B	1.2B	**35.34**	**26.43**	**33.66**	**26.01**
7	KD	1.2B	0.4B	31.52	21.94	30.60	24.55
8	KD	1.2B	1.2B	32.89	23.38	31.77	24.89
9	KT	1.2B	0.4B	34.44	24.50	32.59	25.67
10	KT	1.2B	1.2B	35.07	24.58	32.96	26.05
11	KT+KD	1.2B	0.4B	34.56	23.21	32.66	25.98
12	KT+KD	1.2B	1.2B	**36.15**	**27.18**	**34.23**	**27.14**

Table 5: Results of knowledge transfer for the incremental language pairs. The external models are two sizes of M2M-100. The highest score of each original model is highlighted in **bold**.

extended parameters are still hard to optimize. The knowledge transfer method can further guide additional parameters to achieve greater improvements on these language pairs.

Considering the different introduced manners of pluggable modules, based on the baselines, parallel modules tend to be weaker than serials. It demonstrates that parallel architecture is more difficult to learn new knowledge from limited training data. And the results show that the knowledge transfer method mitigates this issue and explores the potential of parallel architectures, achieving obvious improvement on all eight translation directions, even outperforming the model training from scratch.

Adding Multiple Languages Simultaneously

As shown in Table 3, we examine the translation qualities in incremental learning when eight new language pairs arrive simultaneously. The results show that our proposed method can also achieve better performance compared with the baselines. Notably, in the low resource scenario (*ro* and *uk*), our method of adding multiple languages obtains better performance compared with adding a single language.

Besides, adding multiple languages simultaneously in incremental training makes more training samples available and it facilitates the optimization of challenging pluggable modules in a parallel manner. In this setting, the parallel pluggable modules of all methods demonstrate better performance than the serial. Moreover, the situation of incremental language pairs that are difficult to learn is still alive with the Adapter. It even shows more severe degeneration on the other incremental languages (17.24 BLEU on *en*→*de* of adding a single language while 6.56 BLEU of adding multiple languages simultaneously). However, our method does not significantly been disturbed by different conditions in incremental learning, which exhibits good stability, as shown in Table 2 and Table 3.

Degeneration in Incremental Learning.

As shown in Table 4, to demonstrate the reliability and effectiveness, we investigate the degeneration on the original translation directions, compared with various outstanding continual learning methods. The results demonstrate that our method achieves competitive performance on the incremental translation directions and even outperforms the fine-tuning strategy (up to +0.57/+0.40 for *ro*→*en* and *en*→*ro* respectively). Please refer to Appendix 4.6 and B.2 for more details of the original models and all baselines.

Besides, the results also show that prior replay-based and regularization-based methods still suffer from pronounced degeneration on the original translation directions without the original data. Although no degradation has occurred using Prompt and Prefix, they are vulnerable to learning new knowledge from updated training samples incrementally. More importantly, considering the reliability of the comparison, we have only selected the translation directions between Romanian and English. Because previous methods cannot obtain comparable results when the incremental languages are not related to the set of original languages.

4.4 Results on Pre-trained Models

As shown in Table 5, we leverage pre-trained M2M-100 models (Fan et al., 2021) as the external model and investigate the effectiveness of different knowl-

No.	Transfer Scopes		Incremental Language Pairs(BLEU)							
	Embedding	FFN	*ro→en*	*en→ro*	*de→en*	*en→de*	*bn→en*	*en→bn*	*uk→en*	*en→uk*
1	✗	✗	32.31	20.87	28.66	20.77	27.51	12.49	30.69	20.60
2	✗	✓	33.04	24.48	30.59	25.35	29.04	14.12	30.67	23.72
3	✓	✗	34.08	24.56	28.74	21.15	28.67	13.99	31.66	23.44
4	✓	✓	**34.44**	**25.62**	**32.04**	**26.73**	**30.81**	**15.71**	**32.01**	**25.20**

Table 6: Results on different transfer areas for the incremental language pairs. The highest score is highlighted in **bold**. "✓" indicates that the pluggable module is injected through knowledge transfer in this area. "✗" indicates that the pluggable module is randomly initialized with the Gaussian distribution.

Method	*ro→en*	*en→ro*	*bn→en*	*en→bn*
Ours	34.44	25.62	30.81	15.71
+Self-Attention	34.14	25.35	30.37	15.28
+Gate-Fusion	33.39	24.52	29.85	14.54
+Dropout	33.98	25.42	30.01	14.13

Table 7: Results on the incremental language pairs with different pluggable modules.

edge transfer methods. Knowledge distillation (KD) (Hinton et al., 2015) is a widely used technique to transfer knowledge between models. The results show that only utilizing KD cannot achieve comparable performance for incremental language adaptation. However, KD can be arbitrarily integrated into our method (KT) and further facilitate the procedure of knowledge transfer. The combination of KD and KT achieves better translation qualities than only using one alone based on all model settings.

Besides, both KD and KT are better at learning knowledge from the large pre-trained models. It proves that the large pre-trained model contains more useful knowledge. And we find that the size between models also determines the performance on incremental translation directions. The small M2M-100 model (0.4B) is beneficial for the same size original model (0.4B) but is insufficient to support the large original model (1.2B). In contrast, the large M2M-100 model (1.2B) plays a positive role in both small and large original models by knowledge transfer. However, the small original model (0.4B) limits learning sufficient knowledge from the large M2M-100 model according to the comparison between No.6 and No.12, as shown in Table 5.

4.5 Ablation Studies

Effects on Transfer Areas

As shown in Table 6, we further investigate the effectiveness of our method in different transfer areas. The results demonstrate that our method can help each pluggable module to be better optimized separately and achieves better performance when both two pluggable modules are injected through knowledge transfer for all incremental languages. Specifically, the method improves translation qualities related to Romanian and Ukrainian when it affects the pluggable module in the embedding layer. On the contrary, it is more effective to transfer the knowledge for the pluggable modules in the FFN layers on translation directions related to German and Bengali, according to the comparison between 2 and 3. A possible reason is that the resource of different language pairs influences the efficiency of knowledge transfer.

Effects on Pluggable Modules

Previous parameter-isolation based methods propose various components to introduce additional parameters in the hidden layers (He et al., 2021). As shown in Table 7, inspired by them, we modify the usage of the pluggable modules in the hidden layers and our method is stable on the four translation directions. In particular, we also inject the pluggable modules in the Self-Attention layer. However, the special modification of pluggable modules does not demonstrate effective performance in incremental learning for MNMT.

4.6 Results on Original Language Pairs

To demonstrate the validity and reliability of our method, we build two powerful MNMT models as the original models. As shown in Table 8, the original models achieve state-of-the-art performance on all original translation directions, compared with the other powerful MNMT models.

4.7 More Comparisons

Due to space limitation, we provide a more detailed analysis of our method in Appendix C, including the training cost of the incremental learning, the

Model	Size	*en→ha*	*en→is*	*en→ja*	*en→pl*	*en→ps*	*en→ta*	AVG.
Ours	0.4B	12.41	21.30	14.48	26.43	4.93	11.26	15.14
Ours	1.2B	13.42	22.62	16.12	27.91	5.41	12.03	16.25
M2M-100	0.4B	2.75	13.12	9.51	22.55	2.92	1.67	8.75
M2M-100	1.2B	6.14	18.60	11.67	28.08	4.79	1.82	11.85
Model	Size	*ha→en*	*is→en*	*ja→en*	*pl→en*	*ps→en*	*ta→en*	AVG.
Ours	0.4B	12.88	31.19	19.14	30.20	11.21	16.54	20.19
Ours	1.2B	13.41	32.31	19.27	31.91	12.29	17.62	21.14
M2M-100	0.4B	4.78	22.44	10.59	25.75	7.65	2.82	12.34
M2M-100	1.2B	9.24	29.33	13.43	28.87	10.91	2.70	15.75

Table 8: Results of English-to-xx and xx-to-English with different MNMT models on the original language pairs (WMT-7).

visualization of sentence representations on all language pairs, and the case study on new language pairs, demonstrating the effectiveness of the knowledge transfer method in incremental learning for new language adaptation.

5 Conclusion

In this work, we propose a knowledge transfer method in incremental learning for MNMT, which leverages the knowledge from neural models. It can encourage original models to learn new knowledge from updated training data while naturally mitigating the issue of degradation on previous translation directions. Moreover, it is more efficient to utilize the knowledge transfer scheme than introducing randomly initialized parameters in incremental learning. Experimental results demonstrate that the proposed method outperforms several strong baselines in the comprehensive language consideration.

Limitations

In this work, we attempt to extend an existing MNMT model to support new language pairs with an acceptable expense. In addition to the advantages, our method has the following limitations:

(1) Additional introduced parameters. We utilize the parameter-isolation based method to support new language pairs. The total parameters of the MNMT model have been increased by pluggable modules to achieve better performance than prior studies. In the future, we will compress the number of parameters to the same size of original models meanwhile preserve the performance on all translation directions.

(2) The gap between our scenario and the real-world scenario. Our proposed method is a white-box service in incremental learning. Thus, we train a powerful MNMT model as the original model instead of directly utilizing existing models from the Internet. And we only consider eight incremental language pairs due to the limitation of computation resources. We try our best to simulate the real-world scenario and we will apply our proposed method for large-scale pre-trained MNMT models (e.g., NLLB 54.5B and M2M 12B) to validate the effectiveness in industrial scenarios.

Acknowledgements

This work is supported by the National Key R&D Program of China (2022ZD0160502) and the National Natural Science Foundation of China (No. 61925601, 62276152, 62236011). We sincerely thank the reviewers for their insightful comments and suggestions to improve the quality of the paper.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. 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.

Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2021. Learning to solve nlp tasks in an incremental number of languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 837–847.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. Multieurlex-a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996.

Yong Cheng, Ankur Bapna, Orhan Firat, Yuan Cao, Pidong Wang, and Wolfgang Macherey. 2022. Multilingual mix: Example interpolation improves multilingual neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4092–4102.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

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

Damai Dai, Wenbin Jiang, Qingxiu Dong, Yajuan Lyu, Qiaoqiao She, and Zhifang Sui. 2022. Neural knowledge bank for pretrained transformers. *arXiv preprint arXiv:2208.00399*.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567.

Carlos Escolano, Marta R Costa-Jussà, and José AR Fonollosa. 2021. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, 72(2):190–203.

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

Akhbardeh Farhad, Arkhangorodsky Arkady, Biesialska Magdalena, Bojar Ondřej, Chatterjee Rajen, Chaudhary Vishrav, Marta R Costa-jussa, España-Bonet Cristina, Fan Angela, Federmann Christian, et al. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88. Association for Computational Linguistics.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 178–186.

Robert French. 1993. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? *Advances in Neural Information Processing Systems*, 6.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Shuhao Gu, Bojie Hu, and Yang Feng. 2022. Continual learning of neural machine translation within low forgetting risk regions. *arXiv preprint arXiv:2211.01542*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Yichong Huang, Xiaocheng Feng, Xinwei Geng, and Bing Qin. 2022. Omniknight: Multilingual neural machine translation with language-specific self-distillation. *arXiv preprint arXiv:2205.01620*.

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.

Kasidis Kanwatchara, Thanapapas Horsuwan, Piyawat Lertvittayakumjorn, Boonserm Kijsirikul, and Peerapon Vateekul. 2021. Rational lamol: A rationale-based lifelong learning framework. 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 2942–2953.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Surafel M Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. In *International Workshop on Spoken Language Translation*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467.

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.

Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. 2020. Dictionary-based data augmentation for cross-domain neural machine translation. *arXiv preprint arXiv:2004.02577*.

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.

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Natural language understanding with privacy-preserving bert. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM 2021*. Association for Computing Machinery.

Chenze Shao and Yang Feng. 2022. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2023–2036, Dublin, Ireland. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of 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 2062–2068.

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, Hande Celikkanat, Vinit Ravishankar, Mathias Creutz, and Jörg Tiedemann. 2022. A closer look at parameter contributions when training neural language and translation models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4788–4800.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the wmt 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99.

Shiyue Zhang, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Mohit Bansal, and Francisco Guzman. 2022. How robust is neural machine translation to language imbalance in multilingual tokenizer training? *arXiv preprint arXiv:2204.14268*.

Yang Zhao, Junnan Zhu, Lu Xiang, Jiajun Zhang, Yu Zhou, Feifei Zhai, and Chengqing Zong. 2022. Life-long learning for multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:2212.02800*.

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

A Dataset Details

We utilize six language pairs to train the original MNMT model that covers 12 translation directions and 7 languages (WMT-7). All the original training data comes from the recent WMT general translation track. And we conduct eight incremental language pairs in incremental learning from the WMT news translation track and FLoRes. All data follow the license that can be freely used for research purposes (Farhad et al., 2021). The license of FLoRes dataset is CC-BY-SA 4.0. In addition, we follow Fan et al. (2021) to clean the training sample. We introduce the characteristics of different languages to analyze the linguistic diversity, as shown in Table 9. All language pairs are English-centric and the statistics of training data are shown in Table 10.

A.1 Data Statistics

As the general setting, all language pairs are divided into three categories in terms of the amount of parallel data, including high resource (>10M), medium resource (1M~10M), and low resource (100k~1M). Specifically, the original language pairs are, High resource: Japanese and Polish; Medium resource: Icelandic and Pashto; Low resource: Hausa and Tamil. And the incremental language pairs are, Medium resource: German and Bengali; Low resource: Ukrainian and Romanian. Note that incremental training data is often a non-high resource in the real-world scenario.

A.2 Language Consideration

In this work, we explore a more complex and comprehensive scenario for MNMT in incremental learning, taking into account the diversity of incremental languages. These incremental languages differ from the original languages in terms of their scripts and belong to different language families, which leads to a serious vocabulary and linguistic gap. Inspired by Zhang et al. (2022), if the incremental language has a distinct script with the set of original languages, a certain proportion of OOV tokens with unclear semantics will occur between the original and incremental languages and hinder the performance on new language pairs. Moreover, it is important to note that a language family refers to a group of languages that share a common ancestry, known as the proto-language[7]. This concept highlights the historical connections among languages and their evolution over time. Additionally, differences in grammar and word order can be observed across distinct language families[8]. These linguistic variations further contribute to the existing gap between incremental languages, making their translation more challenging.

In our setting, the 4 incremental languages include: Bengali, which is not related to any of the original 7 languages, and has a distinct script; Ukrainian, which is related to the original language Polish with the language family Slavic, but has a distinct script with Cyrillic; Romanian, is Romance language that is not related to all the original languages, but has a share script with Latin characters; German, which is similar to the original languages in the language families and scripts. The statistics and details of datasets for original and incremental languages are shown in Table 9.

Code	Language	Genus	Script	Order
ha	Hausa	West Chadic	Latin	SVO
is	Icelandic	Germanic	Latin	SVO
ja	Japanese	Japanese	Kanji	SOV
pl	Polish	Slavic	Latin	SVO
ps	Pashto	Iranian	Arabic	SOV
ta	Tamil	Dravidian	Tamil	SOV
de	German	Germanic	Latin	SVO
ro	Romanian	Romance	Latin	SVO
uk	Ukrainian	Slavic	Cyrillic	SVO
bn	Bengali	Indic	Bengali	SOV

Table 9: The characteristics of languages in our setting. The top half part represents the set of original languages. The second half represents the set of incremental languages. Genus represents a subcategory of language families.

B Model Details

B.1 Training Setup

We implement Transformer translation models in all our experiments. In particular, the small original model (0.4B) consists of 6 stacked encoder layers, 6 stacked decoder layers, and 16 multi-attention heads, followed by the configuration of Transformer-Big (Vaswani et al., 2017). The dimensions of d_{model} and d_{ffn} are 1024 and 4096 respectively. The large original model (1.2B) consists of 24 stacked encoder layers, 24 stacked decoder layers, and 16 multi-attention heads, followed by the configuration of M2M-100 (Fan et al., 2021). The

[7] https://en.wikipedia.org/wiki/Languagefamily

[8] https://wals.info

Language Pair	Data Sources			# Samples		
	Train	Dev	Test	Train	Dev	Test
ja-en	WMT21	WMT20	WMT21	18,001,428	993	1,005
pl-en	WMT20	WMT20	WMT20	10,206,520	2,000	1,001
is-en	WMT21	WMT21	WMT21	4,376,282	2,004	1,000
ps-en	WMT20	WMT20	WMT20	1,155,942	2,698	2,719
ha-en	WMT21	WMT21	WMT21	744,856	2,000	997
ta-en	WMT20	WMT20	WMT20	660,818	1,989	997
de-en	WMT14	WMT13	WMT14	4,508,785	3,000	3,003
ro-en	WMT16	WMT16	WMT16	610,320	1,999	1,999
uk-en	FLoRes	FLoRes	FLoRes	8,604,580	997	1,012
bn-en	FLoRes	FLoRes	FLoRes	925,896	997	1,012

Table 10: The statistics of train, dev, and test data for the original languages (WMT-7) and the incremental languages. The top half part represents the set of the original languages. The second half represents the set of the incremental languages.

dimensions of d_{model} and d_{ffn} are 1024 and 8192 respectively. We use Adam (Kingma and Ba, 2014) and a half-precision training scheme to optimize the parameters of all MNMT models. In addition, we reset the optimizer and learning scheduler in incremental learning and use the temperature-based sampling scheme (Arivazhagan et al., 2019) with a temperature of $T = 5$ to balance the training data between diverse language pairs. We adopt the early stop (patience is 10) strategy in incremental learning and the batch size is 4096×4 in all training procedures. To eliminate the randomness of the result, we report the mean BLEU scores of the models that are trained in five seeds. All incremental models are trained on 2 NVIDIA A100 GPUs.

B.2 Continual Learning Baselines

We compare our method with various representative baselines in continual learning. The baselines are as follows:

- Replay (Sun et al., 2019): creating pseudo data for the original language pairs and training new language pairs jointly with the pseudo data and incremental training data.
- EWC (Kirkpatrick et al., 2017): computing the importance of the parameters with Fisher matrix and employing an additional penalty into the loss function to preserve original knowledge.
- Self-KD (Castellucci et al., 2021): utilizing the original models as the teacher model to distill old knowledge.
- LFR (Gu et al., 2022): constraining the parameters of original models with low forgetting risk regions. We choose the LRF-CM for adapting new language pairs.
- Prompt (Chalkidis et al., 2021): prepending prompts to the input embedding in the first layer.
- Prefix (Li and Liang, 2021): prepending prefixes to the keys and values of the attention at every layer.

Method	*ja*(old)→*ro*(new)	*de*(old)→*ro*(new)
Adapter	1.09	5.37
Adapter+LSE	10.12	18.05
KT	15.12	22.20

Table 11: The BLEU scores on zero-shot direction.

C More Comparisons

C.1 Training Cost

To further illustrate the efficiency of our method, we investigate the training time compared with the stronger baselines, as shown in Figure 3. The results show that the knowledge transfer method can reduce the training time of incremental learning, which is more efficient and practical than the other methods.

C.2 Visualization of Multilingual Representations

As shown in Figure 4, we visualize the sentence representations on xx-to-English translation directions to investigate the representation gap between languages. Due to comparability in one representation space, we need multi-source sentences that represent the same meaning in different languages.

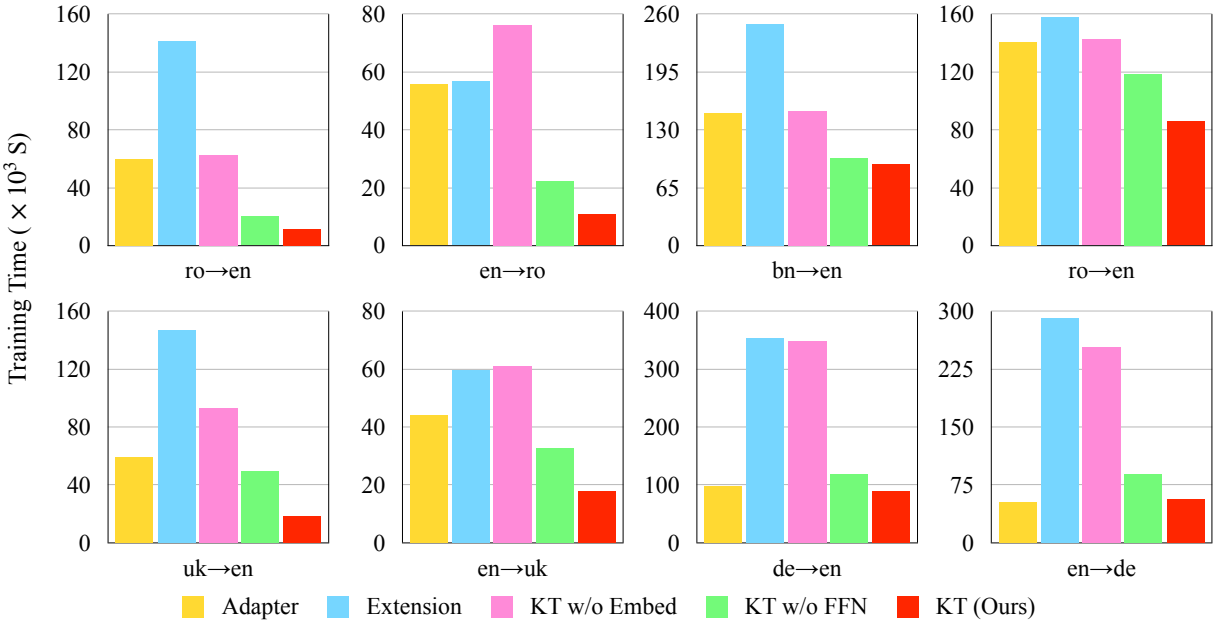
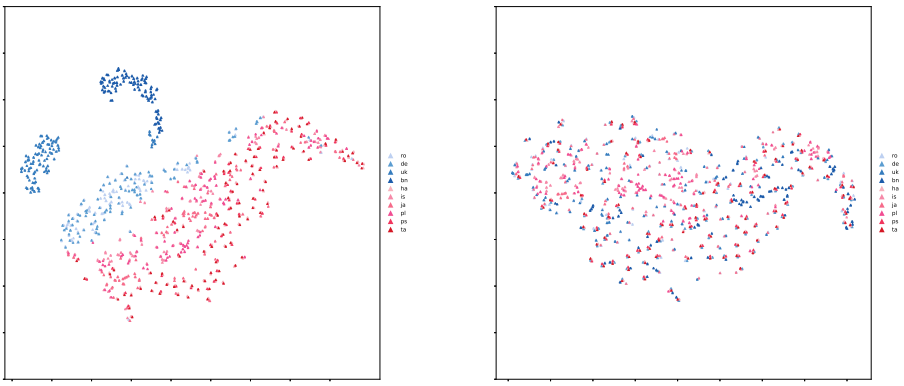


Figure 3: The training time of various methods in incremental learning for MNMT.



(a) Adapter (b) KT (Ours)

Figure 4: T-SNE visualizations of encoder representations of original and incremental languages on xx-to-English translation directions by Adapter and our method. Note that the blue color scheme represents incremental languages while the red color scheme represents original languages.

We use "FLoRes" and reduce the 1024-dim representations to 2-dim with t-SNE (Van der Maaten and Hinton, 2008) for visualization.

As Figure 4 shows, the sentence representations using our method are drawn closer than the standard Adapter method (one of the baselines). It demonstrates that our method can well adapt to the new language. Moreover, previous studies have shown that if sentences with similar semantics are closer together in the representation space, it can usually improve the translation performance of zero-shot translation. Experimental results in two translation directions show that our method can achieve better performance for zero-shot translation, which is consistent with our visualization.

C.3 Case Study

We present several translation examples to provide a comprehensive understanding of the knowledge transfer method, as shown in Table 12. The examples demonstrate that our method can effectively adapt original models to new languages especially when the incremental language is not related to the set of original languages. In particular, due to

the vocabulary gap, the Adapter method is vulnerable to learning incremental languages that have a distinct script with Latin. Although the Extension alleviates this issue by expanding the embedding layer, the additional parameters are not fully optimized to suffer from the off-target problem for MNMT.

D Potential Risks of Our Method

Since our proposed method can increase the unlimited number of translation directions, it is possible for some malicious users to use the MNMT model to provide translation services for politically sensitive languages. For instance, a malicious user may utilize our model to generate hateful or offensive sentences in some politically sensitive languages.

Source en→bn	S1: Widespread looting reportedly continued overnight, as law enforcement officers were not present on Bishkek's streets. S2: Bishkek was described as sinking into a state of "anarchy" by one observer, as gangs of people roamed the streets and plundered stores of consumer goods. S3: Several Bishkek residents blamed protesters from the south for the lawlessness.
Reference	S1: রাতজুড়ে ব্যাপক লুণ্ঠন চলছে কারণ বিশকেক'স স্ট্রিট এ আইন প্রয়োগকারী কর্মকর্তারা উপস্থিত ছিলেন না। S2: "একজন পর্যবেক্ষকের দ্বারা বিশকেক বলতে বোঝানো হয়েছিল ""অরাজকতা"", কারন একদল লোক রাস্তায় ঘুরে বেড়াত এবং ভোগ্যপন্যের জিনিসপত্র লুঠ করত।" S3: বিশকেকের বেশ কিছু বাসিন্দা আইনশৃঙ্খলার ঘাটতির জন্য দক্ষিনের প্রতিবাদকারীদের দোষ দিয়েছেন।
Adapter (fail)	S1: বাংলা বাংলা বাংলা বাংলা, বাংলা বাংলা বাংলা, বাংলা বাংলা, বাংলা বাংলা বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বালা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বালা S2: বাংলা, বাংলা, বাংলা, বাংলা, বালা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, ব S3: বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, বাংলা, ব
Extension	S1: বিশকেকের রাস্তায় আইনশৃঙ্খলা রক্ষাকারী বাহিনীর কর্মকর্তারা উপস্থিত ছিলেন না বলেই রাতারাতি লুটপাট চালানো অব্যাহত ছিল বলে জানা গিয়েছে । S2: বিশকে একজন পর্যবেক্ষকের দ্বারা 'রাজকীয়' অবস্থায় ডুবে যাওয়া বলে বর্ণনা করা হয়েছিল, কারণ জনতার দল রাস্তায় ঘোরাফেরা করে এবং ভোক্তা পণ্যের দোকান লুট করে দিয়েছিল । S3: বেশ কয়েকজন বিশকেকের বাসিন্দা বেআইনি আচরণের জন্য দক্ষিণ থেকে প্রতিবাদকারীদের দোষারোপ করেছেন ।
Ours	S1: বিশকেকের রাস্তায় আইন প্রয়োগকারী কর্মকর্তারা উপস্থিত না থাকায় রাতভর ব্যাপক লুটপাট অব্যাহত ছিল বলে জানা গেছে । S2: বিশকেককে একজন পর্যবেক্ষকের দ্বারা 'রা৯জতন্ত্র' অবস্থার মধ্যে ডুবে যাওয়ার বর্ণনা দেওয়া হয়েছিল, কারণ একদল লোক রাস্তায় ঘোরাফেরা করে এবং ভোগ্যপণ্যের দোকান লুট করে নিয়ে যায় । S3: বিশকেকের বেশ কয়েকজন বাসিন্দা দক্ষিণ দিক থেকে আসা প্রতিবাদকারীদেরকে এই অধর্মের জন্য দোষারোপ করেছে ।
Source en→uk	S1: Angel (2006), explains the Continuum approach as a method being used to help organizations reach a higher level of performance. S2: It has been known for a long time that different types of brain damage, traumas, lesions, and tumours affect behaviour and cause changes in some mental functions. S3: The Sundarbans has been declared a UNESCO World Heritage Site. The part of the forest within Indian territory is called Sundarbans National Park.
Reference	S1: "Ангел" (2006) пояснює підхід Континуума як метод, що використовується для допомоги організаціям у досягненні вищого рівня продуктивності. S2: "Було давно відомо, що різні типи пошкоджень мозку, травми, ураження та пухлини впливають на поведінку і призводять до змін у деяких психічних функціях. S3: Сундарбан був оголошений об'єктом Світової спадщини ЮНЕСКО. Частина лісу в межах індійської території називається Національним парком Сундарбан.
Adapter (fail)	S1: Angel (2006), explică abordarea Continuum ca metodă utilizată pentru a ajuta organizaţiile la un nivel mai mare de performanţă. S2: Відомо давного часу відомилося, що різні типи мозоку пошкоду, травми, резіонів і туморів впливають поведінку і викликають зміни в деяких психічних функціях. S3: Die Sundarbans wurde eine UNESCO Weltkulturbebe erklärt, die Teil des Waldes innerhalb indischen Territoriums wird Sundarbans National Park genannt.
Extension	S1: Angel (2006), пояснює підхід Continuum як метод, який використовується, щоб допомогти організаціям досягти більш високого рівня видатності. S2: З давно відомо, що різні типи пошкодження мозку, травми, перізаціями і тумари впливають на поведінку і викликають зміни в деяких психічних функціях. S3: Сундарбанські були оголошені UNESCO Світньої Спадщини, частина лісу в межах Індії називається Сандарбанською Національним парком.
Ours	S1: Ангел (2006), пояснює підхід Континуума як метод, який використовується, щоб допомогти організаціям досягти більш високого рівня продуктивності. S2: Давно відомо, що різні типи пошкоджень мозку, травм, ураження та пухлини впливають на поведінку і викликають зміни в деяких психічних функціях. S3: Сундарбан був оголошений об'єктом Світової спадщини ЮНЕСКО. Частина лісу на території Індії називається Національний парком Сундарбан.

Table 12: Examples of several baselines and our method on en-to-bn and en-to-uk translation directions for each label based on FLoRes testset. We only highlight some words and fragments to show the representative difference between various methods.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We provide the limitations of our work in the section 'Limitation'.

☑ A2. Did you discuss any potential risks of your work?
We provide the potential risks of our work in Appendix D.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The paper's main claims are summarized in the section 'Abstract' and the section 1.

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

B ☑ Did you use or create scientific artifacts?

We provide the dataset and open toolkit in the section 4.1, 4.2, and Appendix A.

☑ B1. Did you cite the creators of artifacts you used?
We cite the dataset and open toolkit in the section 4.1, 4.2, and Appendix A.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We discuss the license of dataset in Appendix A.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We discuss the existing artifact was consistent with their intended use n the section 4.2 and Appendix A.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We check the details of dataset in Appendix A.

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

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

C ☑ Did you run computational experiments?

We provide the computational experiments in Appendix B and C.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We provide all implementation details and training setup in section 4, Appendix B.1 and Appendix B.2. The section 4.4 also contains the number of parameters in the models used. We report the training cost in Appendix C.1.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We provide the experimental setup with hyper-parameters and configuration in Appendix B.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report the statistics about our results in Appendix B.1.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We used existing packages of scripts and toolkit and we report these details in section 4.2.

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

Left blank.

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

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

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

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

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