SEQ2SEQ PRE-TRAINING WITH DUAL-CHANNEL RECOMBINATION FOR TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequence to sequence (*seq2seq*) pre-training has achieved predominate success in natural language generation (NLG). Generally, the powerful encoding and language generation capacities from the pre-trained seq2seq models can significantly improve most NLG tasks when fine-tuning them with task-specific data. However, as a cross-lingual generation task, machine translation needs an additional ability of representation transferring on languages (or *translation model*). Fine-tuning the pre-trained models to learn the translation model, which is not covered in the self-supervised processing, will lead to the *catastrophic forgetting* problem. This paper presents a dual-channel recombination framework for translation (DCRT) to address the problem mentioned above. In the proposed approach, we incorporate two cross-attention networks into the pre-trained seq2seq model to fetch the contextual information and require them to learn the *translation* and *language* models, respectively. Then, the model generates outputs according to the composite representation. Experimental results on multiple translation tasks demonstrate that the proposed DCRT achieves considerable improvements compared to several strong baselines by tuning less than 20% parameters. Further, DCRT can incorporate multiple translation tasks into one model without dropping performance, drastically reducing computation and storage consumption.

1 INTRODUCTION

Large-scale pre-trained models (PTMs) have become a foundation of natural language processing (NLP) in recent years (Qiu et al., 2020; Han et al., 2022). Generally, fine-tuning PTMs with task-specific training data can achieve significant improvements compared to training the model from scratch. There is a wide variety of pre-trained language models, which have different structures and training objectives for different types of downstream tasks (Devlin et al., 2019; Lample & Conneau, 2019; Lewis et al., 2019). For natural language generation (NLG), *e.g.*, summarization (Qi et al., 2020) or dialogue (Zhong et al., 2022), sequence-to-sequence pre-training (S2SPTM) is more effective than others. Different from the pre-trained encoders (Devlin et al., 2019; Lample & Conneau, 2019) or decoders (Radford et al., 2018; 2019), S2SPTM builds the ability of encoding and generation in a single model, which naturally suits the conditional text generation scenario (Lewis et al., 2019; Liu et al., 2020; Qi et al., 2020).

Nevertheless, neural machine translation (NMT) does not reap the dividend of the S2SPTM, especially for improving the performance of rich-resource tasks (Liu et al., 2020; Song et al., 2019). Unlike other NLG tasks, NMT needs an additional capacity for representation transferring on languages (or *translation modeling*). The self-supervised training objective determines that PTMs only learn representations in the same language unless using a large number of parallel datasets. This intrinsic property leads to fine-tuning the PTMs to directly learn the translation model will cause the *catastrophic forgetting* problem (McCloskey & Cohen, 1989; Goodfellow et al., 2013). The model will gradually forget the ability of language modeling in the process of learning translation. Thus, it is worth studying how to explore the S2SPTM in NMT.

Recent studies have noticed the problem mentioned above and proposed some methods. As shown in Figure 1, we divide them into several categories. The straightforward idea is to freeze the pre-trained encoder or decoder and tune the rest with the NMT training objective (Figure 1b). Compared to fine-tuning all parameters (Figure 1a), these approaches expect to preserve the pre-trained representation

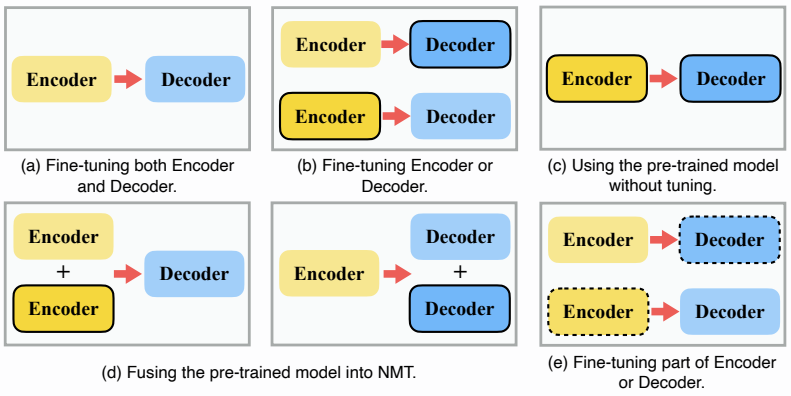

Figure 1: An illustration of the different strategies of exploiting PTMs for NMT.

and learn to translate. However, these methods do not address the gap between the PTM and NMT, which makes them ineffective (Weng et al., 2020; Zhu et al., 2020). For the same reason, the prompting-based approaches also do not work in NMT (Brown et al., 2020; Tan et al., 2021) (Figure 1c). Instead of trying to make the PTM learn to translate, some studies use it as an external knowledge base (Zhu et al., 2020; Yang et al., 2019; Weng et al., 2020). As shown in Figure 1d), NMT can read the pre-trained representation when performing the translation. This series of methods effectively use pre-trained encoders. However, they ignore the powerful generative capability of the S2SPTM, which is equally essential for NMT. In the meantime, parameter efficient fine-tuning (PEFT) shows its advantages in the NLP community. The main idea of PEFT is to tune a small number of parameters or a lightweight adapter (Figure 1e), which has shown competitive results to fine-tuning the whole model (Houlsby et al., 2019; Bapna & Firat, 2019; Li & Liang, 2021; Philip et al., 2020). In NMT, only adapting the cross-attention network in mBART outperforms fine-tuning all parameters (Gheini et al., 2021). We believe that PEFT is a promising way to exploit S2SPTM for NMT, but the current work does not study it in depth.

In this paper, inspired by PEFT, we present a dual-channel recombination framework for translation (DCRT). In the proposed framework, we incorporate two cross-attention networks from the S2SPTM and NMT to connect the encoder and decoder. The two cross-attention networks, named as *language channel* and *translation channel*, are responsible for *language generation* and *cross-lingual transferring*, respectively. In addition, we notice that the representation from the cross-attention networks is easy to degenerate. Specifically, the representation from one channel will become similar after fine-tuning, which is detrimental to the subsequent continuation of the translation. To avoid the aforementioned catastrophic forgetting problem, we propose two training objectives to ensure the consistency of the representation contained in the same channel and the relationship between representations from different channels before and after fine-tuning.

To illustrate the effectiveness of the DCRT, we conduct experiments on widely used rich-resource and low-resource translation tasks. Experimental results on multiple tasks demonstrate that DCRT based on the mBART (Liu et al., 2020) gets absolutely improvements by only tuning less than 20% parameters. DCRT achieves state of the art on both WMT14 EN→DE and EN→FR tasks. Moreover, we can easily extend DCRT as a multilingual NMT model, which can improve the utilization of multilingual S2SPTM. In this setting, DCRT gets a comparable performance with the single model.

2 APPROACH

In this section, we will briefly introduce Transformer (Vaswani et al., 2017) and the typical training objectives of NMT and S2SPTM. Then, we will expound on the proposed DCRT based on them.

2.1 BACKGROUND

Transformer The Transformer network (Vaswani et al., 2017) is widely used in the natural language process (NLP) and computer vision (CV). The Transformer is composed of an encoder network and a decoder network. Here, we can further disassemble the decoder into a cross-attention and

self-attention network. The specific definition is as follows:

$$\textsc{Transformer}(\cdot;\theta) = \textsc{Dec}(\textsc{Enc}(\cdot;\theta_e);\theta_d) = \textsc{XAtt}(\textsc{SAtt}(\textsc{Enc}(\cdot;\theta_e);\theta_s);\theta_x), \tag{1}$$

where the Transformer is parameterized by θ. The θ_e and θ_d are the parameter of the encoder and decoder, respectively. The θ_d is decomposed into θ_s and θ_x, which are the parameter of the self-attention network and the cross-attention network, respectively. Thus, θ is equal to $\{\theta_e, \theta_x, \theta_s\}$. The three modules are composed of multiple self-attention layers; the number of layers is N. The flow of the representation at the nth layer of the decoder is

$$\mathbf{S}^n = \textsc{XAtt}^n(\widetilde{\mathbf{S}}^n, \mathbf{H}; \theta_x), \widetilde{\mathbf{S}}^n = \textsc{SAtt}^n(\mathbf{S}^{n-1}; \theta_s), \mathbf{H} = \textsc{Enc}(\boldsymbol{x}; \theta_d), \tag{2}$$

where the matrix $\mathbf{H}$ is the contextual representation of the input sentence $\boldsymbol{x}$ encoded by the encoder, and the matrix $\mathbf{S}^n$ is the output representation generated by the cross-attention network. Generally, in NLG, the output representation from the final layer $\mathbf{S}^N$ is used to generate the output sentence.

Neural Machine Translation Given a *source-target* parallel sentence pair $\{\boldsymbol{x}, \boldsymbol{y}\}$ from the data-set $\boldsymbol{B}$, where $|\boldsymbol{B}|$ is the number of parallel sentence pairs. The target of NMT is $\boldsymbol{y} = f_\theta(\boldsymbol{x})$, where the parameter θ is optimized by maximizing the likelihood $P_\theta(\boldsymbol{y}|\boldsymbol{x})$. Specifically, the loss function is:

$$\mathcal{L}_\mathcal{N}(\theta) = -\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\boldsymbol{B}}[\log P_\theta(\boldsymbol{y}|\boldsymbol{x})] = -\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\boldsymbol{B}}[\frac{1}{J}\sum_j \log P_\theta(y_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1})], \tag{3}$$

where J is the length of $\boldsymbol{y}$ and $\boldsymbol{y}_{1:j-1}$ means the sub-sequence before the jth word.

Sequence to Sequence Pre-training There are many successful S2SPTMs have been proposed recently (Song et al., 2019; Qi et al., 2020; Lample & Conneau, 2019), we only present the most general form here. Given a sentence $\boldsymbol{z}$ from the unlabeled data-set $\boldsymbol{M}$, which contains $|\boldsymbol{M}|$ sentences, the goal of S2SPTM is to reconstruct the sentence: $\boldsymbol{z} = f_\theta(\phi(\boldsymbol{z}))$, where $\phi(\cdot)$ is used to mask a percentage of words in the $\boldsymbol{z}$ (Devlin et al., 2019; Lample & Conneau, 2019; Song et al., 2019). The loss function of the S2SPTM is

$$\mathcal{L}_\mathcal{S}(\theta) = -\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{M}}[\log P_\theta(\boldsymbol{z}|\phi(\boldsymbol{z}))] = -\mathbb{E}_{\boldsymbol{z}\sim\boldsymbol{M}}[\frac{1}{I}\sum_I \mathbf{1}_{\mathrm{z_i}\notin\phi(\mathrm{z})} \log P_\theta(z_i|\phi(\boldsymbol{z}), \boldsymbol{z}_{1:i-1})], \tag{4}$$

where I is the length of $\boldsymbol{z}$ and $\mathbf{1}$ is an indicator function.

2.2 The Proposed DcRT

Preliminary To better illustrate our approach, we first give some necessary definitions. We define a S2SPTM train by Equation 4 as $f_{\theta^{\text{pm}}}(\cdot)$, where θ^{pm} is made of $\{\theta_e^{\text{pm}}, \theta_x^{\text{pm}}, \theta_d^{\text{pm}}\}$. Then, we train a NMT model $f_{\theta^{\text{mt}}}(\cdot)$ by only tuning the cross-attention network of the $f_{\theta^{\text{pm}}}(\cdot)$. The parameter θ^{mt} is $\{\theta_e^{\text{mt}}, \theta_x^{\text{mt}}, \theta_e^{\text{mt}}\}$, where θ_e^{mt} and θ_d^{mt} are equal to θ_e^{pm} and θ_d^{pm}, respectively.

Dual-channel Recombination Structure Just like Equation 2, we can formalize the dual-channel recombination structure as

$$\mathbf{C}_n = \mathrm{DCR}(\widetilde{\mathbf{S}}_n, \mathbf{H}; \theta_c, \theta_l, \theta_t), \widetilde{\mathbf{S}}^n = \textsc{SAtt}^n(\mathbf{C}^{n-1}; \theta_s), \mathbf{H} = \textsc{Enc}(\boldsymbol{x}; \theta_d), \tag{5}$$

where $\mathrm{DCR}(\cdot)$ can be decomposed as:

$$\mathbf{C}_n = \textsc{Com}(\mathbf{S}_n^l, \mathbf{S}_n^t; \theta_c), \begin{cases} \mathbf{S}_n^l = \textsc{XAtt}^n(\widetilde{\mathbf{S}}_n, \mathbf{H}; \theta_l) \\ \mathbf{S}_n^t = \textsc{XAtt}^n(\widetilde{\mathbf{S}}_n, \mathbf{H}; \theta_t). \end{cases} \tag{6}$$

Compared to the standard Transformer, our model has two cross-attention networks and a combination function $\textsc{Com}(\cdot)$. The two cross-attention networks come from the S2SPTM and NMT models, respectively. In other words, we use θ_x^{pm} to initialize θ_l and θ_x^{mt} to initialize θ_t. Here, the $\textsc{XAtt}(;\theta_l)$ serves as the *language channel* to preserve the generation ability from the S2SPTM, and the $\textsc{XAtt}(;\theta_t)$ is the *translation channel* which learns the representation transferring on languages.

Then, we consider the generation of the $\mathbf{C}_n$ as a probabilistic decision processing. Specifically, given the jth hidden state $\boldsymbol{s}_{n,j}^l$ from $\mathbf{S}_n^l$ and $\boldsymbol{s}_{n,j}^t$ from $\mathbf{S}_n^t$, the corresponding output $\boldsymbol{c}_{n,j}$ is computed by:

$$\boldsymbol{c}_{n,j} = P_{\theta_c}(c_{n,j}|c_{n,j-1}) * \boldsymbol{s}_{n,j}^l + (1 - P_{\theta_c}(c_{n,j}|c_{n,j-1})) * \boldsymbol{s}_{n,j}^t. \tag{7}$$

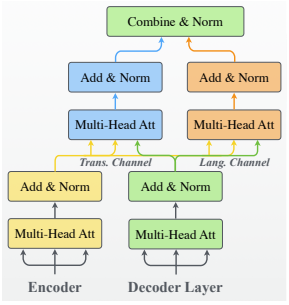

Figure 2: An illustration of the dual-channel recombination structure.

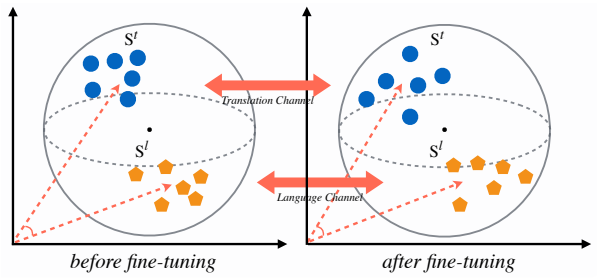

Figure 3: An illustration of the change of representation in semantic space before and after fine-tuning.

Following the nature of sequence prediction, we define a conditional probability $P_{\theta_c}(c_{n,j}|c_{n,j-1})$ to decide the ratio of using $\boldsymbol{s}^l_{n,j}$ and $\boldsymbol{s}^t_{n,j}$ for each state. The $P_{\theta_c}(c_{n,j}|c_{n,j-1})$ is computed by

$$P_{\theta_c}(c_{n,j}|c_{n,j-1}) \triangleq g_{\theta_c}(\boldsymbol{s}^l_{n,j}, \boldsymbol{s}^l_{n,j}, \boldsymbol{c}_{n,j-1}) \approx g_{\theta_c}(\boldsymbol{s}^l_{n,j}, \boldsymbol{s}^l_{n,j}, \boldsymbol{s}_{n-1,j-1}). \tag{8}$$

We do an approximate here to avoid reducing training efficiency. The $g_{\theta_c}(\cdot)$ is computed by:

$$g_{\theta_c}(\boldsymbol{s}^l_{n,j}, \boldsymbol{s}^l_{n,j}, \boldsymbol{s}_{n-1,j-1}) = \frac{\exp\left[\tau(\boldsymbol{s}_{n-1,j-1}, \boldsymbol{s}^l_{n,j})\right]}{\exp\left[\tau(\boldsymbol{s}_{n-1,j-1}, \boldsymbol{s}^l_{n,j})\right] + \exp\left[\tau(\boldsymbol{s}_{n-1,j-1}, \boldsymbol{s}^t_{n,j})\right]}, \tag{9}$$

where $\tau(\cdot)$ is a feed-forward network. We concatenate the two vectors, *i.e.*, $[\boldsymbol{s}_{n-1,j-1}, \boldsymbol{s}^l_{n,j}]$, as the input. We expect the dual-channel mechanism to retain the capabilities of both models and the decoder to dynamically fetch the representation from them. An illustration is shown in Figure 2.

Fine-tuning the model with the NMT training objective will lead to the catastrophic forgetting problem. Specifically, we notice that the representation from one channel will become similar to another during the fine-tuning stage. This representation degeneration phenomenon will be detrimental to the NMT. As shown in the Figure 3, we adopt two constraints, named *content consistency* and *relation consistency*, to avoid the these problems. The details are shown below.

Content Consistency Given the parallel sentence pair $\{\boldsymbol{x}, \boldsymbol{y}\}$, we feed $\boldsymbol{x}$ to go through the NMT model $f_{\theta^{\text{mt}}}(\cdot)$ and $\phi(\boldsymbol{y})$ to the pre-trained model $f_{\theta^{\text{pm}}}(\cdot)$. to obtain two distributions of the two models, *i.e.*, $P_{\theta_{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})$ and $P_{\theta_{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))$. Then, we regularize the model predictions by minimizing the Kullback-Leibler divergence ($D_{\text{KL}}(\cdot)$) between these two output distributions for the output distribution $P_\theta(\boldsymbol{y}|\boldsymbol{x})$, which is:

$$\begin{aligned}\mathcal{L}_{\mathcal{C}}(\theta) &= \frac{1}{2}\mathcal{L}_{\mathcal{CL}}(\theta) + \frac{1}{2}\mathcal{L}_{\mathcal{CT}}(\theta) = \frac{1}{2}(D_{\text{KL}}(P_{\theta^{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))\|P_\theta(\boldsymbol{y}|\boldsymbol{x})) + D_{\text{KL}}(P_{\theta^{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})\|P_\theta(\boldsymbol{y}|\boldsymbol{x})))\\ &= \frac{1}{2}\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\boldsymbol{B}}\left\{\mathbb{E}_{P_{\theta^{\text{pm}}}}\left[\log\frac{P_\theta(\boldsymbol{y}|\boldsymbol{x})}{P_{\theta^{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))}\right] + \mathbb{E}_{P_{\theta^{\text{mt}}}}\left[\log\frac{P_\theta(\boldsymbol{y}|\boldsymbol{x})}{P_{\theta^{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})}\right]\right\}.\end{aligned} \tag{10}$$

The goal of this training function is that the representations after training are able to preserve the semantic information present before training.

Relation Consistency On the other hand, the representations from the two channels can be seen as two low-dimension manifolds in the semantic space. To preserve their characteristics, we let the relative distance of the representations from the two channels be the same. The detail of the loss function is as follows:

$$\mathcal{L}_{\mathcal{R}}(\theta) = -\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\boldsymbol{B}}\left\{\frac{1}{J}\sum_{j=1}^{J}\left[||\textsc{Dis}(\text{s}^j_{P_{\theta^{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})}, \text{s}^j_{P_{\theta^{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))}) - \textsc{Dis}(\text{s}^{t,j}_{P_\theta(\boldsymbol{y}|\boldsymbol{x})}, \text{s}^{l,j}_{P_\theta(\boldsymbol{y}|\boldsymbol{x})})||_2^2\right]\right\}, \tag{11}$$

where $\textsc{Dis}(\cdot)$ is the distance function, for which we use the cosine similarity.

Algorithm 1 The overall training process of the proposed DCRT.

Input: Parallel set $\boldsymbol{B}$; Seq2seq pre-trained model with parameter $\theta^{\text{pm}} = \{\theta_e^{\text{pm}}, \theta_s^{\text{pm}}, \theta_x^{\text{pm}}\}$; Epoch E
Output: The NMT model $f_\theta(\cdot)$ with parameter $\theta = \{\theta_e, \theta_s, \theta_l, \theta_t, \theta_c\}$

Initialize the parameter θ^{mt} of the NMT model $f_{\theta^{\text{mt}}}(\cdot)$ by the θ^{pm}
for $e = 0$ to E **do**
  for $\{\boldsymbol{x}, \boldsymbol{y}\}$ in $\boldsymbol{B}$ **do**
    Optimize θ_x^{mt} by the Equation 3 ▷ Freezing $\{\theta_e^{\text{mt}}, \theta_s^{\text{mt}}\}$
  end for
end for
Randomly initialize the parameter θ of the DCRT model ▷ Equation 5-Equation 9
Reinitialize the corresponding parameters of θ by the θ^{mt} and θ^{pm}: $\theta_e^{\text{pm}} \rightarrow \theta_e$, $\theta_s^{\text{pm}} \rightarrow \theta_s$, $\theta_x^{\text{pm}} \rightarrow \theta_l$, $\theta_x^{\text{mt}} \rightarrow \theta_t$
for $e = 0$ to E **do**
  for $\{\boldsymbol{x}, \boldsymbol{y}\}$ in $\boldsymbol{B}$ **do**
    Compute the probability $P_{\theta^{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))$ and $\mathbf{S}_{P_{\theta^{\text{pm}}}(\boldsymbol{y}|\phi(\boldsymbol{y}))}$ by the S2SPTM $f_{\theta^{\text{pm}}}(\cdot)$
    Compute the probability $P_{\theta^{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})$ and $\mathbf{S}_{P_{\theta^{\text{mt}}}(\boldsymbol{y}|\boldsymbol{x})}$ by the NMT model $f_{\theta^{\text{mt}}}(\cdot)$
    Optimize $f_\theta(\cdot)$ by the Equation 13 ▷ Freezing $\{\theta_e, \theta_s\}$
  end for
end for
return $f_\theta(\cdot)$

Training The overall training function can be formalized as:

$$\mathcal{L}(\theta; \boldsymbol{B}) = \mathcal{L}_{\text{N}}(\theta; \boldsymbol{B}) + \alpha \mathcal{L}_c(\theta; \boldsymbol{B}) + \beta \mathcal{L}_{\text{R}}(\theta; \boldsymbol{B}), \tag{12}$$

where the coefficient α and β are the hyper-parameters to control loss function, which we set as 1 and 0.5, respectively. The loss of the training function is:

$$\begin{aligned} \nabla\mathcal{L}(\theta) &= \nabla\mathcal{L}_{\mathcal{N}}(\theta) + \frac{\alpha}{2}\nabla\mathcal{L}_{\mathcal{CT}}(\theta_t) + \frac{\alpha}{2}\nabla\mathcal{L}_{\mathcal{CL}}(\theta_l) + \beta\nabla\mathcal{L}_{\mathcal{R}}(\theta_t, \theta_l) \\ &= \nabla\mathcal{L}_{\mathcal{N}}(\theta) + \underbrace{\frac{\alpha}{2}\nabla\mathcal{L}_{\mathcal{CT}}(\theta_t) + \beta\nabla\mathcal{L}_{\mathcal{R}}(\theta_t)}_{\textit{Regularization for TM}} + \underbrace{\frac{\alpha}{2}\nabla\mathcal{L}_{\mathcal{CL}}(\theta_l) + \beta\nabla\mathcal{L}_{\mathcal{R}}(\theta_l)}_{\textit{Regularization for LM}}. \end{aligned} \tag{13}$$

Compared to the vanilla training function (Equation 3), the proposed method has additional terms to regularize the parameters further. Here, we divide them into two parts. The first part is used to regularize the parameters θ_t for the translation model, and the second is used to regularize the parameters θ_l for language model. DCRT can leverage this kind of regularization to avoid the catastrophic forgetting problem.

3 EXPERIMENT

3.1 IMPLEMENTATION DETAIL

Data-set We first conduct experiments on the three widely-used WMT translation tasks: WMT14 English→German (EN→DE), WMT14 English→French (EN→FR) and WMT18 Chinese→English (ZH→EN), The training data sizes of EN→DE, EN→FR and ZH→EN are 4.5M, 36M and 21M, respectively. Both EN→DE and EN→FR tasks, we use the `newstest2013` as the dev set and the `newstest2014` as the test set. On the ZH→EN task, we use the `newsdev2017` as the dev set and the `newstest2018` as the test set.

Moreover, we evaluate the proposed DCRT on several low-resource machine translation tasks, including WMT16 English↔Romanian (EN↔RO), IWSLT14 English↔German (EN↔DE), IWSLT17 English→French (EN→FR) and English→Chinese (EN→ZH) tasks. The sizes of the training set for EN↔RO, EN↔DE, EN→FR and EN→ZH are 60K, 160k, 236k and 235k, respectively. On the EN↔RO task, we use the `newstest2015` as the dev set and the `newstest2016` as the test set. Following the Zhu et al. (2020), on the EN↔DE task, we split 7k sentence pairs from the training dataset as the dev set and concatenate the dev2010, dev2012, tst2010, tst2011 and tst2012 as the test set. On the EN→FR and EN→ZH tasks, we use the official dev/test sets of the corresponding years.

Table 1: The comparison of the proposed DCRT and previous approaches. #PARAM is the number of parameters. "T/F" indicates the number of trainable and frozen parameters. "$*$" means the model is implemented by ourselves. The BLEU score is computed by the *multi-bleu.perl*, and the results computed by the sacreBLEU are shown in Appendix A.

Model	#PARAM(T/F)	EN→DE	EN→FR
Transformer-*big** (Vaswani et al., 2017)	207M/207M	28.73	41.70
Depth Growing NMT (Wu et al., 2019)	268M/268M	29.92	43.27
SD-Transformer (Li et al., 2020)	437M/437M	30.46	43.29
Transformer *w/* mBERT* (Devlin et al., 2019)	164M/164M	29.19	41.46
Transformer *w/* mBART* (Liu et al., 2020)	680M/680M	29.87	43.03
Transformer *w/* XLM-R (Conneau et al., 2020)	700M/700M	30.9	43.8
VECO (Luo et al., 2021)	662M/662M	31.7	44.5
HICTL (Wei et al., 2021)	700M/700M	31.74	43.95
XLM (Lample & Conneau, 2019)	570M/570M	28.8	–
mRASP (Lin et al., 2020)	375M/375M	–	44.3
Transformer *w/* mBERT* (Devlin et al., 2019)	54M/164M	28.17	40.03
Deep Fusion (Weng et al., 2022)	171M/171M	31.59	44.21
CTNMT (Yang et al., 2019)	–	30.10	42.30
BERT-fused NMT (Zhu et al., 2020)	–	30.75	43.78
mGPT *w/* Prompt Tuning (Zhang et al., 2021)	131K/560M	5.9	–
mGPT *w/* Pre-fix Tuning (Zhang et al., 2021)	26M/560M	17.5	–
Multi-Stage Prompting (Tan et al., 2022)	19M/560M	21.2	–
CROSSATT Tuning* (Gheini et al., 2021)	40M/680M	29.51	42.83
DCRT *w/* mBART	91M/731M	**32.56**	**45.53**

Setting We implement all experiments on the *fairseq* toolkit.[1] We adopt the Transformer-*big* as the baseline for the rich-resource tasks and Transformer-*base* for the low-resource tasks. We use mBART25 as the default sequence to sequence pre-trained model in our experiment Liu et al. (2020). We apply the byte pair encoding (BPE) (Sennrich et al., 2016b) to all language pairs and limit vocabulary size to 32K. We set label smoothing as 0.1 and dropout rate as 0.1. The Adam is adopted as the optimizer, and the $\beta 1/\beta 2$ is set as $0.9/0.98$ for the *base* setting and $0.9/0.998$ for the *big* setting. We set the initial learning rate as 1e-3 and use the default warm-up strategy with 4000 steps. In addition, for all pre-trained encoders, we append a 6 layers decoder with the same hidden size to compose the NMT model. We set the learning rate as 1e-5 for models initialized by pre-trained encoders and 2e-5 for models initialized by seq2seq pre-trained models. We use the polynomial decay and set the label smoothing rate as 0.2. We use 8 A100 GPUs to train the rich-resource tasks and one A100 GPU to train the low-resource tasks.

We use beam search as the decoding algorithm. For the WMT EN→DE, we set the beam size as 4 and the length penalty as 0.6. For other tasks, we set the beam size as 5 and the length penalty as 1.0. For a fair comparison, we calculate the *case-sensitive tokenized* BLEU with the *multi-bleu.perl* script for the WMT tasks and IWSLT EN↔DE task, and use the *sacreBLEU*[2][3] to calculate *case-sensitive* BLEU Papineni et al. (2002) for other tasks.[4]

3.2 RESULTS AND DISCUSSION

Rich-resource Translation Tasks The results on the WMT14 EN→DE and EN→FR tasks are summarized in the Table 1. We implement the Transformer and report two advanced variations proposed by Wu et al. (2019) and Li et al. (2020) as baselines. The Transformer baseline gets 28.73

[1]https://github.com/pytorch/fairseq

[2]https://github.com/mjpost/sacreBLEU

[3]BLEU+case.mixed+lang.${Src}-${Trg}+num- refs.1+smooth.exp+test.${Task}+tok.13a+version.1.5.1

[4]We also give the results computed by the sacreBLEU for the WMT tasks in Appendix A.

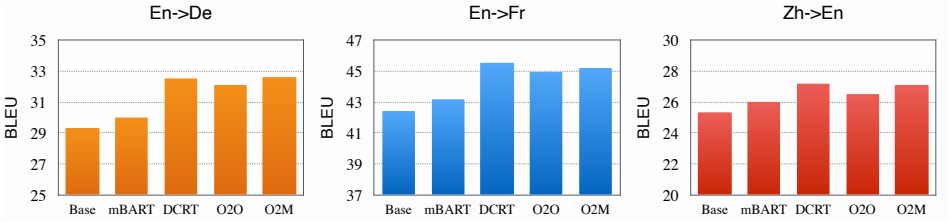

Figure 4: A comparison of the different dual-channel structures. "O2O" means that the model has one language channel and one translation channel. "O2M" means that the model has one language channel and multiple translation channels.

Table 2: The results of the proposed DCRT on the low-resource translation tasks. #PARAM is the number of trainable parameters.

Model	#PARAM	EN→RO	RO→EN	EN→DE	DE→EN	EN→FR	EN→ZH
TRANS	6*65M	32.51	32.67	28.34	34.81	35.73	25.94
TRANS_{M}	107M	30.93	31.72	27.79	34.69	34.92	25.57
XLM-R	700M	35.60	35.80	30.58	35.66	38.14	27.49
mBART	680M	37.43	37.21	30.72	36.79	38.86	28.63
BERT-fused	–	–	–	30.45	36.11	38.70	28.20
DCRT_{O2O}	91M	38.04	38.46	**31.47**	37.51	39.89	29.61
DCRT_{O2M}	346M	**38.52**	**39.02**	31.35	**37.65**	**40.23**	**29.87**

and 41.70 BLEU scores on the EN→DE and EN→FR, respectively. Compared to the Depth Growing NMT and SD-Transformer, we can see that the increase in trainable parameters improves model performance. We sort several related studies out according to the strategy of using pre-trained models. The first is the standard fine-tuning paradigm, which only achieves similar results to the baseline. In the second category, we use the cross-lingual pre-trained models trained with large-scale parallel data sets to initialize the NMT model. In this setting, the translation performance significantly improves (2∼3 BLEU gains). It means that learning the *translation model* in the pre-training or fine-tuning stage is necessary.

The motivation of the third one is to integrate the pre-trained models into the NMT model as a knowledge base. In this situation, the NMT model focuses on learning to translate with the pre-trained contextual representation. These approaches can work well for leveraging the capabilities of pre-trained encoders (1.5∼2.5 BLEU gains). However, they ignore the generation ability from the pre-trained models, which is equally important for machine translation. The final one includes non-parametric and few-parametric (PEFT) methods. The prompting methods with mGPT are not work in NMT. Only fine-tuning the cross-attention network (∼40M) of mBART achieves similar results to tuning the whole model (0.3∼0.4 BLEU drops). Our model gets 32.56 and 45.53 BLEU scores on the EN→DE and EN→FR, respectively. The empirical results prove that preserving the generation ability besides learning translation can improve translation performance effectively. The DCRT gets better quality by only tuning about 91M parameters which are largely less than previous work. It is worth mentioning that more than one-third of the parameters in the mBART is the embedding layer (250M+), which is used to cover more languages. In fact, only 10∼20% of word vectors are used for a given translation task.

One Model Covers Multiple Translation Tasks Then, we combine all high-resource translation tasks into one model to fully use the multilingual pre-trained model. We first put all the tasks into the DCRT model same as above (named DCRT_{O2O}). Then, we make all tasks share the same language channel and have an exclusive translation channel (named DCRT_{O2M}). In other words, if we have three translation tasks, the DCRT_{O2M} will have one language channel and three translation channels. The results are shown in Figure 4, DCRT_{O2O} has a slight descend compared to using one model per task. The DCRT_{O2M} gets the comparable results to the DCRT. In the O2M setting, we can achieve a high-quality multilingual translation model with a few extra parameters, effectively reducing computation and storage consumption.

Low-resource Translation Tasks Following Liu et al. (2020) and Zhu et al. (2020), we evaluate the proposed DCRT with O2O and O2M settings on the six low-resource tasks, which are shown in Table 2. Further, we also implement a multilingual NMT model (TRANS_M) as a comparison. Compared to the Transformer baseline (TRANS), the TRANS_M drops 0.1-1.7 BLEU, while the DCRT_O2O get 2.5$\sim$6.5 BLEU gains. Compared to the mBART, the DCRT_O2O, which reduces 87% trainable parameters, gets 0.61$\sim$1.25 BLEU gain. Then, DCRT_O2M can achieve better results on five of six tasks, in which the trainable parameter size will grow linearly with the number of tasks. We notice that the EN$\rightarrow$RO and RO$\rightarrow$EN have more improvements when they have the individual translation channel. The reason may be that the data of WMT is from the news domain, while IWSLT is from the TED talks. We will investigate the influence of the domain discrepancy in the future.

Semi-supervised Translation We study whether the proposed approach can work with back-translation (BT) (Sennrich et al., 2016a), which is the most widely used data-augmentation method in NMT. We make experiments on the WMT En$\rightarrow$De and Ro$\rightarrow$En tasks. Specifically, for the En$\rightarrow$De, we use 24M synthetic data from Caswell et al. (2019). For the Ro$\rightarrow$En, we use 2M back-translated data from Sennrich et al. (2016a).

Table 3: The results of the proposed DCRT with back-translated dataset.

Model	EN$\rightarrow$DE	RO$\rightarrow$EN
Transformer	28.73	32.67
w/ BT	30.44	38.65
mBART	29.87	37.21
w/ BT	30.94	39.15
DCRT	32.56	39.11
w/ BT	33.43	40.31

The results are shown in Table 3. Our model with BT gets 0.87/1.20 gains compared to only using parallel data. Moreover, DCRT with BT achieves 2.99/1.66 gains compared to the Transformer baseline and 2.49/1.16 gains compared to the mBART. The results demonstrate that DCRT could work with BT to achieve better performance.

Ablation Study To further investigate the effect of each module in the DCRT, we make an ablation study in this section. The results are shown in Table 4. On the one hand, we look at the change in BLEU by removing each training objective. When ablating $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{R}$, the BLEU drops 2.01 and 1.33, respectively. The BLEU drops 2.26 when removing both $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{R}$. The experimental results show that the two objectives are indispensable to avoid representation degradation, and removing either one will lead to a great reduction.

Table 4: The ablation study of the DCRT.

Model	EN$\rightarrow$DE	Δ
DCRT	32.56	–
Remove the $\mathcal{L}_\mathcal{C}$	30.55	-2.01
Remove the $\mathcal{L}_\mathcal{R}$	30.23	-1.33
Remove both $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{R}$	30.30	-2.26
Update all parameters	30.81	-1.75
Random initialization	29.24	-3.32
Freeze the θ^t	31.66	-0.90
Freeze the θ^l	31.98	-0.58
Freeze the θ^l and θ^t	31.04	-1.52

On the other hand, we freeze the parameters of different modules in the model. When fine-tuning all parameters, the BLEU drops 1.75. As a comparison, when initializing all parameters randomly, the BLEU drops 3.32. The results show that even if all parameters can be updated, the model also benefits from the S2SPTM. Then, when freezing θ_t and θ_l, the BLEU drops 0.9 and 0.58, respectively. When freezing both of θ_t and θ_l, the BLEU drops 1.52. The results suggest that tuning these parameters makes the representation more suit for NMT.

The influence of the $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{R}$ For each setting, we sample a checkpoint at each 5000 training steps. Then, we compute the probability from the combination function, BLEU score, and the cosine distance between $\mathbf{S}^l$ and $\mathbf{S}^t$ on the test set on the WMT EN$\rightarrow$DE task. The results are shown in Figure 5. When removing the $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{R}$ (*w/o* Regu), the probability will be biased to the translation channel, and the similarity of the representations of the two channels increases rapidly. We think the DCRT will likely degenerate to only fine-tuning the cross-attention network (Gheini et al., 2021) when there is no mitigation of catastrophic forgetting.

4 RELATED WORK

NMT with Pre-trained Models Using pre-trained models to improve the performance of NMT is an attractive research direction worth studying. Liu et al. (2020); Lample & Conneau (2019) proposed

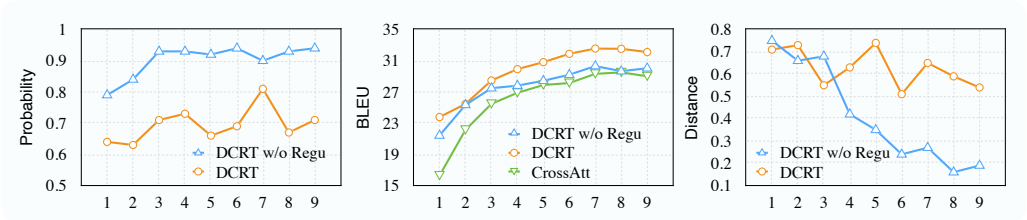

Figure 5: The changes of the representations from the two channels with and without $\mathcal{L}_{\mathcal{C}}$ and $\mathcal{L}_{\mathcal{R}}$.

to fine-tune multi-lingual pre-trained models NMT. However, the fine-tuning method does not work well in rich-resource translation tasks Zhu et al. (2020). Zhu et al. (2020); Yang et al. (2019); Weng et al. (2022) proposed to fuse pre-trained encoders into NMT. Weng et al. (2020) proposed to use knowledge distillation to transfer knowledge from pre-trained models to NMT. Brown et al. (2020) proposed that the generative language model (GPT-3) can use to translate by the prompting method (or in-context learning). Our approach focuses on achieving the complementary representations of the sequence to sequence pre-trained models and NMT.

Parameter Efficient Fine-tuning in NMT Parameter efficient fine-tuning (PEFT) is an effective transfer mechanism, which can reduce the substantial training costs (Houlsby et al., 2019; He et al., 2022; Zaken et al., 2022). Furthermore, Liu et al. (2022) demonstrated that PEFT is cheaper and better than in-context learning in many NLP tasks. In NMT, Bapna & Firat (2019) proposed injecting task-specific adapter layers into the pre-trained NMT model for domain adaption and multilingual tasks. Guo et al. (2020) proposed to fine-tune the BERT with an adapter module for parallel sequence decoding. Gheini et al. (2021) proposed to fine-tune the cross-lingual network of the S2SPTM for NMT. According to the characteristics of the S2SPTM and NMT, we proposed the DCRT to amplify the advantage of PEFT in NMT further.

Multi-channel Framework in NMT The multi-channel framework has been widely used in NMT. Zhang & Zong (2016) proposed using an additional encoder to model source contextual knowledge with monolingual data. Song et al. (2018) adopted a dual-channel model based on the different structures, *i.e.*, a self-attention module, and a CNN module, to model contextual representation in different aspects. Xiong et al. (2018) extended the dual-channel framework to multi-channel for fetching information on different levels according to the linguistic structure. On the other hand, the mixture-of-experts (MoE) method is an expansion of the multi-channel framework (Shi et al., 2019; Masoudnia & Ebrahimpour, 2014). Dai et al. (2022) proposed to use MoE structure in NMT. Then, NLLB Team et al. (2022) gave a more general and effective version. Furthermore, the multi-channel structure is widely-used in multi-modal MT, in which the different channels extract the inputs from different modals. For example, Huang et al. (2016); Fang & Feng (2022) used a dual-channel structure to encode image and text separately, then feed them to the decoder for translation. Inspired by them, we use a dual-channel structure to obtain pre-trained representations, effectively avoiding the problems caused by simply fine-tuning the pre-trained model.

5 Conclusion

In this work, we present a novel dual-channel recombination framework (DCRT) to exploit the sequence-to-sequence pre-trained models in NMT. Specifically, we incorporate two cross-attention networks into the pre-trained seq2seq model to fetch the contextual information. On the other hand, to avoid catastrophic forgetting and representation degeneration problems, we require them to learn the *translation* and *language* models with the content consistency and the relation consistency training objectives. Extensive experiments from different settings show that the proposed approach effectively utilizes the pre-trained mBART to improve translation quality by only fine-tuning a small number of parameters. In future work, we will investigate whether the DCRT can combine with other advanced sequence-to-sequence pre-trained models.

REFERENCES

Ankur Bapna and Orhan Firat. Simple, scalable adaptation for neural machine translation. In *EMNLP-IJCNLP*, pp. 1538–1548, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NIPS*, 33:1877–1901, 2020.

Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. *arXiv:1906.06442*, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, pp. 8440–8451, 2020.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. Stablemoe: Stable routing strategy for mixture of experts. In *ACL*, pp. 7085–7095, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pp. 4171–4186, 2019.

Qingkai Fang and Yang Feng. Neural machine translation with phrase-level universal visual representations. *arXiv: 2203.10299*, 2022.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-attention is all you need: Adapting pretrained transformers for machine translation. In *EMNLP*, pp. 1754–1765, 2021.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv:1312.6211*, 2013.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. Incorporating bert into parallel sequence decoding with adapters. In *NeurIPS*, pp. 10843–10854, 2020.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE TPAMI*, 2022.

Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient fine-tuning for vision transformers. *arXiv preprint arXiv:2203.16329*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pp. 2790–2799. PMLR, 2019.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 639–645, 2016.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv:1901.07291*, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv:1910.13461*, 2019.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. Shallow-to-deep training for neural machine translation. In *EMNLP*, pp. 995–1005, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv:2101.00190*, 2021.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. Pre-training multilingual neural machine translation by leveraging alignment information. In *EMNLP*, pp. 2649–2663, 2020.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*, 2022.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*, 2020.

Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si. Veco: Variable and flexible cross-lingual pre-training for language understanding and generation. In *ACL*, pp. 3980–3994, 2021.

Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pp. 109–165. Elsevier, 1989.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. *arXiv*, 2022.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, pp. 311–318, 2002.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. Monolingual adapters for zero-shot neural machine translation. In *EMNLP*, pp. 4465–4470, 2020.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. Prophetnet: Predicting future n-gram for sequence-to-sequencepre-training. In *Findings of EMNLP*, pp. 2401–2410, 2020.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pp. 1872–1897, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *CoRR*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. In *OpenAI blog*, volume 1, pp. 9, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, pp. 86–96, 2016a.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, pp. 1715–1725, 2016b.

Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multimodal deep generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaitao Song, Xu Tan, Di He, Jianfeng Lu, Tao Qin, and Tie-Yan Liu. Double path networks for sequence to sequence learning. In *COLING*, pp. 3064–3074, 2018.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*, pp. 5926–5936, 2019.

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. Msp: Multi-stage prompting for making pre-trained language models better translators. *arXiv:2110.06609*, 2021.

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. Msp: Multi-stage prompting for making pre-trained language models better translators. In *ACL*, pp. 6131–6142, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, volume 30, 2017.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. On learning universal representations across languages. In *ICLR*, 2021.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. Acquiring knowledge from pre-trained model to neural machine translation. In *AAAI*, pp. 9266–9273, 2020.

Rongxiang Weng, Heng Yu, Weihua Luo, and Min Zhang. Deep fusing pre-trained models into neural machine translation. In *AAAI*, 2022.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Depth growing for neural machine translation. In *ACL*, pp. 5558–5563, 2019.

Hao Xiong, Zhongjun He, Xiaoguang Hu, and Hua Wu. Multi-channel encoder for neural machine translation. In *AAAI*, volume 32, 2018.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. Towards making the most of bert in neural machine translation. In *EMNLP*, pp. 9378–9385, 2019.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, pp. 1–9, 2022.

Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*, pp. 1535–1545, 2016.

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, et al. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224, 2021.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *AAAI*, pp. 11765–11773, 2022.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. In *arXiv:2002.06823*, 2020.

A Results on Rich-resource Translation

The results computed by sacreBLEU are shown in Table 5. Our approach gets 31.79 and 42.89 BLEU on the WMT En→De and En→Fr tasks, respectively. Compared to fine-tuning the mBART, our method achieves 2.87 and 2.71 BLEU gains, which outperform all previous work.

Table 5: The comparison of the proposed DCRT and previous approaches. #PARAM is the number of parameters, and "T/F" indicates the tuneable and frozen parameters. "$*$" means the model is implemented by ourselves.

Model	#PARAM(T/F)	EN→DE	EN→FR
Transformer-*big** (Vaswani et al., 2017)	207M/207M	28.42	40.53
Depth Growing NMT (Wu et al., 2019)	268M/268M	29.5	41.8
Transformer *w/* mBERT* (Devlin et al., 2019)	164M/164M	28.51	40.18
Transformer *w/* mBART (Liu et al., 2020)	680M/680M	28.92	41.1
Transformer *w/* XLM-R (Conneau et al., 2020)	700M/700M	39.9	41.2
CROSSATT Tuning* (Gheini et al., 2021)	40M/680M	28.71	40.80
VECO (Luo et al., 2021)	662M/662M	30.6	42.0
mRASP (Lin et al., 2020)	375M/375M	–	41.7
DCRT *w/* mBART	91M/731M	**31.79**	**42.89**