What Works and Doesn't Work, A Deep Decoder for Neural Machine Translation

Anonymous ACL submission

Abstract

Deep learning has demonstrated performance advantages in a wide range of natural language processing tasks, including neural machine translation (NMT). Transformer NMT models are typically strengthened by deeper encoder layers, but deepening their decoder layers usually results in failure. In this paper, we first identify the cause of the failure of the deep decoder in the Transformer model. Inspired by this discovery, we then propose approaches to improving it, with respect to model structure and model training, to make the deep decoder practical in NMT. Specifically, with respect to model structure, we propose a cross-attention drop mechanism to allow the decoder layers to perform their own different roles, to reduce the difficulty of deep-decoder learning. For model training, we propose a collapse reducing training approach to improve the stability and effectiveness of deep-decoder training. We experimentally evaluated our proposed Transformer NMT model structure modification and novel training methods on several popular machine translation benchmarks. The results showed that deepening the NMT model by increasing the number of decoder layers successfully prevented the deepened decoder from degrading to an unconditional language model. In contrast to prior work on deepening an NMT model on the encoder, our method can deepen the model on both the encoder and decoder at the same time, resulting in a deeper model and improved performance.

1 Introduction

With the help of the deep neural network, the feature extraction capability of models has been substantially enhanced (Schmidhuber, 2015; LeCun et al., 2015). Deep neural network models are also popular for natural language processing (NLP) tasks. The most typical deep neural network model in NLP is based on the convolutional neural network (CNN) (Gehring et al., 2017) and Transformer (Vaswani et al., 2017) structures, and the deep pretrained Transformer language model has begun to dominate NLP. The deep neural network model has also attracted substantial interest in neural machine translation (NMT), for both theoretical research (Wang et al., 2019; Li et al., 2020, 2021; Kong et al., 2021) and competition evaluation (Zhang et al., 2020; Wu et al., 2020b,a; Meng et al., 2020). Because it has been demonstrated that deep neural network models can benefit from an enriched representation, deep NMT models also show advantages with respect to translation performance (Wu et al., 2019; Wei et al., 2020).

Although deep models have been extensively studied in machine translation and are frequently used to improve translation performance, almost all work on deepening models has focused on increasing the number of encoder layers; there has been very little research on deepening the decoder. Through preliminary experiments on varying the number of decoder layers in the Transformer NMT model, we observed that, when the decoder is deepened beyond a certain number of layers, the translation performance of the overall model fails to improve; moreover, it declines rapidly to near zero. This demonstrates that there are flaws in the current structure or training method, and the deep-decoder NMT model cannot be trained.

By analyzing the training process of the deep-decoder model, we found that the training perplexity of the model was relatively low, but the translation performance of the obtained model was much worse than that of a shallow model. Inspired by this phenomenon, we hypothesize that, as the decoder deepens, the model may increasingly ignore the source inputs and degenerate to an unconditional language model, even though a low perplexity can be obtained on the training set. In this case, the purpose of translation learning is not achieved, and thus the model training fails.

According to our hypotheses, preventing the decoder from degenerating to an unconditional language model is the key to overcoming the failure of deep-decoder NMT model training. Consequently, we propose two aspects of model improvement: model structure and model training. In model structure, the only difference between the decoder of the NMT model and that of the unconditional language model is cross-attention; therefore, we focus mainly on this structure. In model training, we aim to make the decoder output distant from the output of the unconditional language model to avoid fitting the target sentences while ignoring the source inputs in the training dataset.

Specifically, we propose a cross-attention drop (CAD) mechanism for the deep-decoder layer structure. The original intention of this mechanism is that we suspected that the degeneration of the deep decoder to an unconditional language model was caused by the training difficulties resulting from too many cross-attentions. Because the purpose of cross-attention is to force the decoder layer to obtain features from the source representation, the different layers in the deep decoder should perform distinct roles. However, the conventional deep decoder requires each layer to extract source features similarly, thus increasing the training difficulty. As a result, to minimize training loss, the model chooses to memorize the training target sentences directly and ignore the source inputs. In this mechanism, we drop the cross-attention in some decoder layers to lower the overall training difficulty, thereby preventing the failure of deep-decoder training. In addition to structural changes, we also propose a decoder dropout regularization (DDR) loss and anti-LM-degradation (ALD) loss for joint model optimization, based on contrastive learning; these increase the stability of deep-decoder NMT model training and avoid degeneration to an unconditional language model.

Our experiments were conducted mainly on two popular machine translation benchmarks: WMT14 English-to-German and English-to-French. The results of the experimental exploration of decoders with different depths show that a successfully trained depth decoder greatly benefits the overall translation performance and can work with the deep encoder to achieve higher translation performance. Moreover, the novel training approaches that we propose both increase the stability of the training of the deep-decoder model and enable additional improvements.

2 Related Work

Since the emergence of the Transformer-based model (Vaswani et al., 2017), the deep model has become the mainstream baseline model for machine translation. The Transformer NMT model employs a deeper architecture than the RNN-based model, with six encoder layers and six decoder layers. During the same time period, Gehring et al. (2017) introduced an encoder–decoder architecture wholly based on CNNs, which increased both the number of encoder layers and the number of decoder layers to 20.

Because greater model capacity has the potential to contribute significantly to quality improvement, deepening a model is regarded as a good method of boosting the capacity of the model with the same architecture. It has been shown that more expressive features are extracted (Mhaskar et al., 2016; Telgarsky, 2016; Eldan and Shamir, 2016), which has resulted in improved performance for vision tasks (He et al., 2016; Srivastava et al., 2015) over the past few years. In Transformer NMT models, there have also been numerous studies on deepening the model for better performance. Bapna et al. (2018) took the first step toward training extraordinarily deep models by deepening the encoders for translation, but discovered that simply increasing the encoder depth of a basic Transformer model was insufficient. Because of the difficulty of training, models utterly fail to learn. Transparent attention has also been proposed to regulate deep-encoder gradients; this eases the optimization of deeper models and results in consistent gains with a 16-layer Transformer encoder.

Following research on deepening the encoder to obtain a deep NMT model, as in (Bapna et al., 2018), Wu et al. (2019) proposed a two-stage training strategy with three special model structural designs for constructing deep NMT models with eight encoder layers. Wang et al. (2019) proposed a dynamic linear combination mechanism and successfully trained a Transformer model with a 30-layer encoder, with the proposed mechanism shortening the path from upper-level layers to lower-level layers to prevent the gradient from vanishing or exploding. Zhang et al. (2019) proposed a depth-scale initialization for improving norm preservation and a merged attention sublayer that integrates a simplified average-based self-attention sublayer into the cross-attention module. Fan et al. (2019) employed a layer-drop mechanism to train a 12-6

Transformer NMT model and pruned subnetworks during inference without fine-tuning. More recently, Wei et al. (2020) proposed to attend the decoder to multigranular source information with different space-scales, thereby boosting the training of very deep encoders without special training strategies. Li et al. (2020) developed a shallow-to-deep training strategy and employed sparse connections across blocks to successfully train a 48-layer encoder model. Kong et al. (2021) studied using deep-encoder and shallow-decoder models to improve decoding speed while maintaining high translation quality. Most of these related studies focused on deepening the encoder for deep NMT models, whereas there have been very few studies on deepening the decoder. Herein lies the most significant dissimilarity between our work and this related work.

3 Our Method

Given bilingual parallel sentences $\langle \mathbf{X}, \mathbf{Y} \rangle$, the NMT model learns a set of parameters $\mathbf{\Theta}$ by maximizing the likelihood $\mathcal{J}(\mathbf{Y}|\mathbf{X}, \Theta)$, which is represented as the product of the conditional probabilities of all target words:

$$\begin{aligned}\mathcal{J}_{\text{NLL}}(\mathbf{Y}|\mathbf{X};\mathbf{\Theta}) &= \prod_{i=1}^{|\mathbf{Y}|} P(\mathrm{Y}_i|\mathbf{Y}_{<i},\mathbf{X};\mathbf{\Theta}) \\ &= -\sum_{i=1}^{|\mathbf{Y}|} \log P(\mathbf{Y}_i|\mathbf{Y}_{<i},\mathbf{X};\mathbf{\Theta}),\end{aligned}$$

where $|\mathbf{Y}|$ represents the sequence length of $\mathbf{Y}$, Y_i represents the i-th token of sequence $\mathbf{Y}$, and $\mathbf{Y}_{<i}$ represents all the tokens before the i-th token. Encoder–decoder architectures are commonly employed in NMT to model the translation conditional probabilities $P(\mathbf{Y}|\mathbf{X};\mathbf{\Theta})$, where the encoder and decoder can be implemented as RNNs (Wu et al., 2016), CNNs (Gehring et al., 2017), or self-attention (Vaswani et al., 2017). In this study, we used the most recent Transformer NMT model, based on a self-attention structure, as our baseline.

3.1 Transformer NMT Model

The encoder and decoder in the Transformer NMT model both consist of stacked multiple layers, with each layer composed of attention networks. The following is the basic form of an attention network:

$$\begin{aligned}\textsc{Attn}(\mathbf{H}_{\mathrm{Q}},\mathbf{H}_{\mathrm{KV}}) &= \mathbf{W}_{\mathrm{O}}\left[\mathrm{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}\right], \\ \mathbf{Q},\mathbf{K},\mathbf{V} &= \mathbf{W}_{\mathrm{Q}}\mathbf{H}_{\mathrm{Q}},\mathbf{W}_{\mathrm{K}}\mathbf{H}_{\mathrm{KV}},\mathbf{W}_{\mathrm{V}}\mathbf{H}_{\mathrm{KV}},\end{aligned}$$

where $\mathbf{W}_{\mathrm{Q}}, \mathbf{W}_{\mathrm{K}}, \mathbf{W}_{\mathrm{V}}$, and $\mathbf{W}_{\mathrm{O}}$ are weight parameters, d is the hidden dimension, and $\mathbf{H}_{\mathrm{Q}}$ and $\mathbf{H}_{\mathrm{KV}}$ are two input vectors for attention, with $\mathbf{H}_{\mathrm{Q}}$ serving as a query and $\mathbf{H}_{\mathrm{KV}}$ serving as key and value. When $\mathbf{H}_{\mathrm{Q}}$ and $\mathbf{H}_{\mathrm{KV}}$ are input into the same vector, the attention becomes self-attention: $\textsc{SelfAttn}(\mathbf{H}_{\mathrm{QKV}}) = \textsc{Attn}(\mathbf{H}_{\mathrm{QKV}}, \mathbf{H}_{\mathrm{QKV}})$. To improve feature extraction capabilities, Vaswani et al. (2017) advocated using a multihead mechanism to enhance the original attention; we omit this here for simplicity.

In the encoder, $\mathcal{L}_e$ identical layers are stacked, and each layer has a self-attention sublayer and a pointwise feedforward sublayer. Layer normalization (Ba et al., 2016) and skip residual connection (He et al., 2016) are employed for each sublayer's input and output. The process in the l-th encoder layer can be formalized as follows:

$$\begin{aligned}\hat{\mathbf{H}}_e^l &= \textsc{Ln}\left(\textsc{SelfAttn}(\mathbf{H}_e^{l-1}) + \mathbf{H}_e^{l-1}\right), \\ \mathbf{H}_e^l &= \textsc{Ln}\left(\textsc{Ffn}(\hat{\mathbf{H}}_e^l) + \hat{\mathbf{H}}_e^l\right),\end{aligned}$$

where $\mathbf{H}_e^{l-1}$ denotes the output of the (l-1)-th layer in the encoder, $\textsc{Ffn}(\cdot)$ is the pointwise feedforward sublayer with a two-layer feedforward network and ReLU activation function, and $\mathbf{H}_e^0 = \textsc{Emb}(\mathbf{X})$ denotes the initial representation from the embedding layer.

The decoder consists of $\mathcal{L}_d$ identical layers. As in the encoder, the self-attention network is used to extract features from the target sequence in each layer; however, in addition, a cross-attention is used to extract features from the source sequence. The process of the l-th layer in the decoder can be formalized as follows:

$$\begin{aligned}\hat{\mathbf{H}}_d^l &= \textsc{Ln}\left(\textsc{SelfAttn}(\textsc{CasualMask}(\mathbf{H}_d)) + \mathbf{H}_d^{l-1}\right), \\ \tilde{\mathbf{H}}_d^l &= \textsc{Ln}\left(\textsc{CrossAttn}(\hat{\mathbf{H}}_d^l, \mathbf{H}_e^{\mathcal{L}_e}) + \hat{\mathbf{H}}_d^l\right), \\ \mathbf{H}_d^l &= \textsc{Ln}\left(\textsc{Ffn}(\tilde{\mathbf{H}}_d^l) + \tilde{\mathbf{H}}_d^l\right).\end{aligned}$$

where $\mathbf{H}_d^0 = \textsc{Emb}(\mathbf{Y})$, $\textsc{CausalMask}(\cdot)$ represents the causal mask mechanism (to make any i-th token unable to see future tokens, thereby maintaining unidirectional translation), $\textsc{CrossAttn}(\cdot)$ is the same as $\textsc{Attn}(\cdot)$ in implementation, in which the hidden state on the decoder is input as the query, and the hidden state on the encoder is input as the key and value. The output target sequence is predicted on the output hidden state $\mathbf{H}_d^{\mathcal{L}_d}$ from the top layer of the decoder:

$$P(\mathbf{Y}|\mathbf{X};\mathbf{\Theta}) = \mathrm{Softmax}(\mathbf{W}_{\mathrm{D}}\mathbf{H}_d^{\mathcal{L}_d}),$$

where $\mathbf{W}_{\mathrm{D}}$ is the projection weight parameter, which maps the hidden state to the probability in the vocabulary space.

3.2 Deep Decoder Collapse

In theory, we can construct a deeper Transformer NMT model by stacking more decoder layers in addition to more encoder layers. To illustrate the challenge of simply increasing the number of decoder layers for a deep NMT model, we conducted a preliminary experiment using the WMT14 En→De translation task.

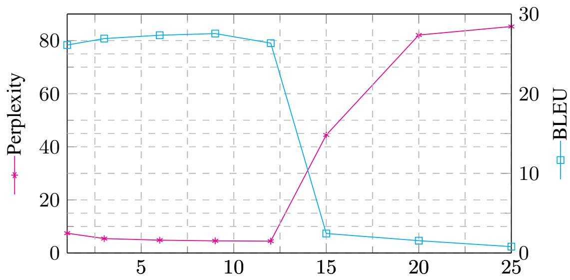

Figure 1: Training perplexity vs. decoder depth and BLEU score vs. decoder depth on WMT14 En→De translation task.

Figure 1 shows the relationship between training perplexity and BLEU score on the test set with different decoder depths after 200K training steps. Except for the number of decoder layers, other hyperparameters were kept consistent with those used in the Transformer-based model setting. The figure shows that, as the number of decoder layers increased, the training perplexity fell gradually and then increased, whereas the BLEU score increased at first and eventually declined to a very low level. This phenomenon is referred to as deep-decoder collapse. The perplexity on the training set appeared to decrease but the translation performance was very poor; we hypothesize that this phenomenon was caused by the model ignoring the source inputs, leading the decoder to degenerate to an unconditional language model. To verify our hypothesis, we made improvements in two respects: model structure and model training.

3.3 Cross-attention Drop

The sole fundamental difference between the decoder in Transformer NMT and the pure unconditional language model, such as GPT2, is the cross-attention in Eq. (3.1). The cross-attention forces the target representation to include features from the source's representation, rather than relying only on the visible target tokens. Although the presence of cross-attention intuitively prevents the decoder from degenerating to an unconditional language model, we argue that it is the presence of cross-attention that makes the learning more difficult. This is because each layer in the deep decoder plays a more distinct role than in a shallow decoder but each layer is forced to extract features from the source representation. Thus, the decoder may abandon the cross-attention and act as an unconditional language model, to achieve a lower training loss.

We propose a drop-net technique to ensure that the features output by self-attention and the encoder are fully exploited. This technique, inspired by dropout (Srivastava et al., 2014) and drop-path (Larsson et al., 2017), can be employed to regularize the network training. Specifically, for the l-th decoder layer, given a drop-net rate of p^l_{net}, we randomly sample a variable $U^l \in [0,1]$, and the calculation of $\tilde{\mathbf{H}}^l_d$ in Eq. (3.1) becomes:

$$\tilde{\mathbf{H}}^l_{d,\text{drop-net}} = \textsc{Ln}\big(\mathbb{1}(U^l > p^l_{\text{net}}) \cdot \hat{\mathbf{H}}^l_d + \mathbb{1}(U^l > 1 - p^l_{\text{net}}) \cdot (\textsc{CrossAttn}(\hat{\mathbf{H}}^l_d, \mathbf{H}^{L_e}_e) + \hat{\mathbf{H}}^l_d)\big).$$

where $\mathbb{1}(\cdot)$ is an indicator function. For layer l, with probability p^l_{net}, only self-attention is used; with probability $(1 - p^l_{\text{net}})$, both of the two attentions are used. During the inference stage, both attentions are used for the $\tilde{\mathbf{H}}^l_d$ calculation. For the simplicity of implementation, we adopted a same fixed p_{net} for layers $1 \le l \le \mathcal{L}_{dr}$ (i.e. $p^l_{\text{net}} = p_{\text{net}}, 1 \le l \le \mathcal{L}_{dr}$), while set $p^l_{\text{net}} = 1.0$ for layers $l > \mathcal{L}_{dr}$. We denote $\mathcal{L}_{dr}$ as the drop depth and p_{net} as the drop ratio.

3.4 Collapse Reducing Training

In addition to the model structure, we introduced two extra losses into model training: one for stable optimization and another to minimize the risk of the decoder degenerating to an unconditional language model. These are the DDR loss and ALD loss, both of which are inspired by the concept of contrastive learning.

Because of the use of dropout and drop-net in the decoder, we propose a simple regularization loss, DDR loss, which is based on the randomness of the model structure. The purpose of this loss, which is inspired by R-drop (Liang et al., 2021), is to regularize the output predictions from different substructures of the deep decoder and increase the stability of the optimization. Specifically, because

Systems	Enc.	Dec.	Ratio	WMT14 En→De Params	Time	BLEU	sacreBLEU	WMT14 En→Fr Params	Time	BLEU	sacreBLEU
(Vaswani et al., 2017) (BIG)	6	6	1.0	213M	N/A	28.40	N/A	222M	N/A	41.00	N/A
(Shaw et al. 2018) (BIG)	6	6	1.0	210M	N/A	29.20	N/A	222M	N/A	41.30	N/A
(Ott et al., 2018) (BIG)	6	6	1.0	210M	N/A	29.30	28.6	222M	N/A	43.20	41.4
(Wu et al., 2019) (BIG)	8	8	1.0	270M	N/A	29.92	N/A	281M	N/A	43.27	N/A
(Wang et al., 2019) (BIG, DEEPE)	30	6	5.0	137M	N/A	29.30	N/A	N/A	N/A	N/A	N/A
(Wei et al., 2020) (BASE, DEEPE)	48	6	8.0	272M	N/A	30.19	N/A	N/A	N/A	N/A	N/A
(Wei et al., 2020) (BIG, DEEPE)	18	6	3.0	512M	N/A	30.56	N/A	N/A	N/A	N/A	N/A
(Li et al., 2020) (BASE, DEEPE)	24	6	4.0	118M	6.16	29.02	27.9	124M	33.81	42.42	40.6
(Li et al., 2020) (BASE, DEEPE)	48	6	8.0	194M	10.65	29.60	28.5	199M	55.35	42.82	41.0
(Li et al., 2020) (BIG, DEEPE)	24	6	4.0	437M	18.31	29.93	28.7	N/A	N/A	N/A	N/A
BASE (Pre-Norm)	6	6	1.0	63M	4.79	27.05	26.0	65M	27.11	41.00	39.2
DEEPE	24	6	4.0	118M	8.66	28.95	27.8	119M	48.43	42.40	40.6
DEEPE	48	6	8.0	194M	16.38	29.44	28.3	195M	90.85	42.75	41.0
DEEP	15	15	1.0	123M	9.82	0.55	0.2	124M	49.96	0.93	0.3
DEEP+CAD+CRT	15	15	1.0	123M	10.52	29.09	28.1	124M	50.13	42.86	41.0
DEEP	27	27	1.0	199M	16.56	0.31	0.1	200M	78.82	0.65	0.1
DEEP+CAD+CRT	27	27	1.0	199M	17.92	**30.31**	**28.8**	200M	79.96	**43.57**	**41.6**
BIG (Pre-Norm)	6	6	1.0	210M	36.05	28.79	27.7	212M	97.51	42.40	40.6
DEEPE	24	6	4.0	437M	42.41	29.90	28.7	439M	102.14	43.11	40.9
DEEP	15	15	1.0	448M	45.32	0.40	0.2	449M	108.02	0.71	0.2
DEEP+CAD+CRT	15	15	1.0	448M	46.52	**30.69**	**29.0**	449M	110.5	**43.95**	**41.9**

Table 1: Number of model parameters, training time (hours), BLEU scores (%), and sacreBLEU scores (%) of translation models on WMT14 En→De and En→Fr tasks. We use BASE and BIG to represent the different parameter settings of the NMT model, DEEP represents the deep NMT model, and DEEPE specifically refers to the deep NMT model with a deep encoder.

the same source representation and target tokens are input twice, the two predicted distributions P_1 and P_2 are forced to be mutually consistent. The probability forms of two separate passes for the decoder only are written as $P_1(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)$ and $P_2(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)$, in which $\mathbf{\Theta}_d$ denotes the parameters of the decoder. The similarity loss of the two prediction distributions is implemented as the minimization of the bidirectional Kullback–Leibler (KL) divergence between the two distributions:

$$\begin{aligned}\mathcal{J}_{\mathrm{DDR}} = \frac{1}{2}\big(&\\ &\mathcal{D}_{\mathrm{KL}}(P_1(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)||P_2(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)+\\ &\mathcal{D}_{\mathrm{KL}}(P_2(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)||P_1(\mathrm{Y}_i|\mathbf{Y}_{<i}, \mathbf{H}_e^{L_e}; \mathbf{\Theta}_d)\big),\end{aligned}$$

where $\mathcal{D}_{\mathrm{KL}}(p||q)$ denotes the logarithmic difference between probabilities p and q. A decoder with drop-net and dropout can converge stably by contrastive learning from the two passes' output distributions of the same input.

With the DDR loss, regularization training is applied to the deep decoder with dropout and drop-net to help the decoder converge; however, the risk of the model degenerating to an unconditional language model remains. To solve this problem, we propose the ALD loss, the primary purpose of which is to allow the model to be aware that the amount of source information used determines the effect on the decoder output, when performing contrastive learning. That is, the output with more source information used should be more similar to the output using full source information than the output with less source information used.

The traditional definition of contrastive learning assumes a set of paired examples, $\mathcal{D} = \{(z_i, z_i^+)\}_{i=1}^M$, where z_i and z_i^+ are semantically related. In contrastive learning, z_i^+ is used as a positive instance of z_i, and other in-batch examples are used as the negative instances. Specifically, the loss of contrastive learning is realized as a cross-entropy loss, and can be represented as follows:

$$\mathcal{J}_{\mathrm{CL}} = -\log \frac{e^{\mathrm{sim}(\mathcal{G}(z_i), \mathcal{G}(z_i^+))/\tau}}{\sum_{j=1}^N e^{\mathrm{sim}(\mathcal{G}(z_i), \mathcal{G}(z_j))/\tau}},$$

where N is the size of a mini-batch, $\mathcal{G}(\cdot)$ denotes a function that transforms a sequence input into a final single-vector representation, $\mathrm{sim}(\mathbf{v}_1, \mathbf{v}_2)$ denotes the cosine similarity $\frac{\mathbf{v}_1^\top \mathbf{v}_2}{\|\mathbf{v}_1\| \cdot \|\mathbf{v}_2\|}$, and τ is a softmax temperature hyperparameter. In SimCSE (Pan et al., 2021), the $\mathcal{G}(\cdot)$ function is implemented as the model with an additional pooling layer that obtains the sentence representation. Because the presence of dropout in the model results in different outputs for the same input, the input is treated as a positive instance of z_i itself.

In ALD loss, our purpose is entirely different from the above. We consider using more source inputs as positive instances and fewer as negative instances of z_i, with all source inputs. Specifically, for the translation pair $\langle \mathbf{X}, \mathbf{Y} \rangle$, we randomly sample a ratio $\gamma \in [0, p_{\text{ALD}}), 0 < p_{\text{ALD}} < 0.5$, replace the token in $\mathbf{X}$ with UNK in the ratio γ to obtain $\mathbf{X}^+$, and replace the X in the ratio $(1-\gamma)$ with UNK to obtain $\mathbf{X}^-$.

$$\mathcal{J}_{\text{ALD}} = -\log \frac{e^{\text{sim}(\mathcal{G}(\mathbf{X},\mathbf{Y}),\mathcal{G}(\mathbf{X}^+,\mathbf{Y}))/\tau}}{\sum_{*\in[+,-]} e^{\text{sim}(\mathcal{G}(\mathbf{X},\mathbf{Y}),\mathcal{G}(\mathbf{X}^*,\mathbf{Y}))/\tau}},$$

where $G(\cdot,\cdot)$ denotes average pooling output on the hidden state from the top layer of the decoder (i.e., $\mathcal{G}(\mathbf{X}, \mathbf{Y}) = \text{AvgPool}(\mathbf{H}_d^{\mathcal{L}_d})$). When using ALD loss, if the decoder ignores the source inputs and degenerates to an unconditional language model, the source inputs will have very little impact on the output: $\mathcal{G}(\mathbf{X}, \mathbf{Y})$, $\mathcal{G}(\mathbf{X}^+, \mathbf{Y})$, and $\mathcal{G}(\mathbf{X}^-, \mathbf{Y})$ will all be similar, resulting in confusion for the contrastive learning.

4 Experiment

4.1 Setup

Dataset To compare with previous work, we conducted experiments on two classical machine translation datasets: WMT14 English-to-German (En→De) and English-to-French (En→Fr). The corpus sizes are 4.5M and 36M for the En→De and En→Fr datasets, respectively. Following common practice, we concatenated *newstest2012* and *newstest2013* as the validation set and used *newstest2014* as the test set. We employed `tokenizer.pl` in Moses (Koehn et al., 2007) to tokenize En, De, and Fr sentences, and then used BPE (Sennrich et al., 2016) to split the words into subwords. A joint BPE strategy with 40K merge operations between source and target languages was adopted to construct the vocabulary.

Configuration We adopted the most widely used Transformer (Vaswani et al., 2017) network as our research basis. Two typical parameter settings are often used to fulfill various needs: Transformer BASE and Transformer BIG. Both settings employ a six-layer encoder and a six-layer decoder. The differences between them are the embedding width, feedforward network size, and number of attention heads, which are 512/1024/8 for BASE and 1024/4096/16 for BIG. We used `multi-bleu.perl` and detokenized sacreBLEU[1] to evaluate the translation performance on test sets, for fair comparison with previous work. Other hyperparameter settings for model training were consistent with (Vaswani et al., 2017). The number of training steps was 200K for En→De models and 400K for En→Fr models, the batch size was 4096 tokens per GPU, and the models were trained on eight NVIDIA V100 GPUs.

4.2 Main Results

Table 1 shows the results of our model on the WMT14 En→De and En→Fr translation tasks. To make it easier to compare the results of NMT models with the same depth, we set the total number of layers of the model to be as consistent as possible with that used in related work. Because the encoder is responsible for encoding the source language, and the decoder is in charge of encoding the target language, and the depth of the model affects its abstraction ability, we argue that the encoder should have a depth similar to that of the decoder. Therefore, we employed the same number of layers for the encoder and decoder in the NMT model.

On the basis of the baseline model, the results for the deepened models (denoted by DEEP) suggest that the training encountered failures, and deeper models achieved worse results. When we applied the CAD and CRT approaches to the Deep models, the training failure problem was resolved: the full model both achieved better results than the corresponding baselines and obtained performance superior to that of the model with a deep encoder only. This demonstrates that a deeper model has performance advantages, and our proposed CAD and CRT methods alleviate the problem of deep-decoder collapse. In addition, it reveals that the architecture with balanced encoder and decoder outperforms the architecture with only a deep encoder. We also conducted experiments to deepen the NMT models under the BIG parameter setting, and the performance phenomenon was similar to that observed under the BASE parameter setting.

Compared with (Wang et al., 2019), our model achieved similar results but with fewer layers (30), and did not require a special model structure design. Our models achieved a better translation effect with fewer parameters compared with the results of (Wei et al., 2020), demonstrating that our proposed method is simple and very effective. In comparison with (Li et al., 2020), our models performed simi-

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

larly in En→De translation under the BASE setting, and demonstrated better performance in En→Fr. We believe that this is a consequence of the larger quantity of training data in En→Fr, which allows the decoder to be more fully trained. We obtained generally better results in the BIG setting, whereas Li et al. (2020)'s results were comparable to those of our DEEPE baseline.

4.3 Further Exploration

Effects of Drop Depth and Drop Ratio. As explained in model part, we propose the CAD approach for the deep NMT model structure. To investigate the impact of the drop depth and drop ratio on final translation performance, we conducted experiments on the WMT14 En→De task using the BASE, DEEP-54L model with both CAD and ALD techniques; the experimental results are presented in Figure 2. We found that, when the drop depth was very small for a 27-layer decoder, the model also suffered from the problem of deep-decoder collapse, and the translation performance was very poor. When we increased the drop depth, the translation performance improved progressively, reaching a peak at the 21st layer, confirming our hypothesis that cross-attention is a contributing cause to the problem of deep-decoder collapse.

As the drop depth was increased further, performance suffered, even though there was no training failure. This demonstrates that cross-attention is also an important component of the translation model, and insufficient cross-attention also prevents the model from extracting adequate source information. Furthermore, we compared several drop ratios and observed that, with a small drop depth, $p_{\text{net}} = 1.0$ indicates that all cross-attention drops in the corresponding layer will have a superior final effect. Conversely, with a greater drop depth, a smaller p_{net}—which retains some of the cross-attention—will achieve better results.

Hyperparameters in ALD Loss. To analyze the effect of the hyperparameters—softmax temperature τ and sampling threshold p_{ALD}—in the ALD loss, we conducted experiments on the WMT14 En→De task with the BASE, DEEP-30L model. The results obtained are presented in Figure 3, which shows that increasing the sampling threshold improves the BLEU score. This is because a larger p_{ALD} for UNK replacement can yield a greater diversity of negative examples, which is beneficial for contrastive learning. However, if p_{ALD} is further increased, the difference between positive and negative examples decreases, which has a detrimental impact on the final translation performance. Compared with the sampling threshold p_{ALD}, the temperature τ has a relatively small effect. The experimental results reveal that the BLEU score with $\tau = 0.05$ is slightly lower than that with $\tau = 0.1$. We believe that, when the value of the temperature parameter is too small, the ALD loss is too large, thus affecting the model's convergence.

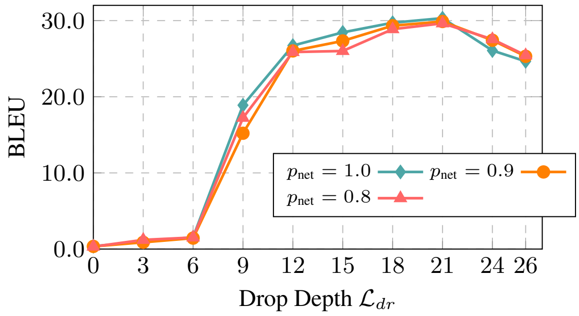

Figure 2: Influence of different drop ratios and depths on translation performance of deep NMT model.

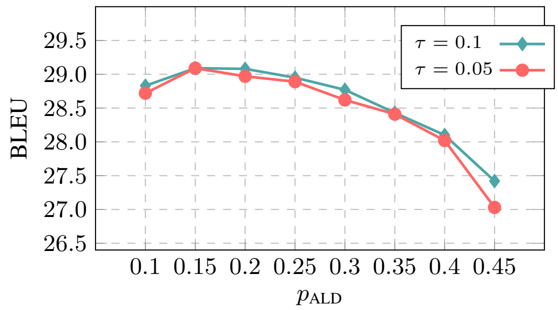

Figure 3: Influence of sampling threshold p_{ALD} and temperature parameter τ on translation performance in ALD loss.

Effects of Encoder Depth and Decoder Depth. Because our method allows for a deep encoder and decoder, we investigated the effect of encoder and decoder depth on translation performance. We selected the BASE, DEEP-30L model as the basis and conducted experiments on the WMT14 En→De translation task, changing only the depth of the encoder or decoder. The results are illustrated in Figure 4. When the encoder depth was 1, the translation performance was significantly poorer than when the decoder depth was 1, indicating that the encoder has a more obvious performance limit at this shallow level. This is because the encoder is directly responsible for the extraction of the source representation, and a shallow encoder cannot ex-

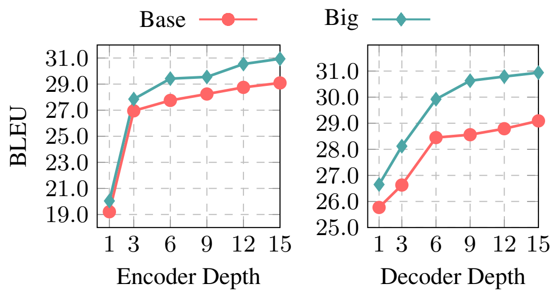

Figure 4: Effects of different encoder and decoder depths when using CAD and CRT methods.

Enc.	Dec.	BLEU	sacreBLEU
24	6	28.95	27.8
6	24	28.21	27.0
15	15	29.09	28.1

Table 2: Performance of deep NMT models with different combinations of encoder and decoder depth.

tract enough source information. This suggests that, if resources are restricted and the number of layers needs to be decreased to obtain a smaller model, it is more effective to reduce the number of decoder layers; this finding is compatible with Kasai et al. (2021)'s conclusion. In addition, increasing the depth of both the encoder and the decoder improves the model's translation performance, implying that increasing the number of decoder layers is effective in a deep NMT model.

The balance between the number of encoder layers and the number of decoder layers in a deep model is another important consideration. To investigate this, we compared translation performance in three typical cases on WMT14 En→De with the total number of encoder and decoder layers set to 30. As shown in Table 2, the model with an equal number of encoder and decoder layers achieved the best results, outperforming the pure deep-encoder and deep-decoder models.

5 Ablation Study

We conducted ablation studies on the modifications that we made to both the model structure and training to investigate their respective effects on the translation performance. The ablation research was conducted on the WMT14 En→De task, as before, and the model employed was the BASE, DEEP-30L-Full model. We began by adding extra R-Drop, DDR, ALD, and CAD techniques to its baseline model (BASE, DEEP-30L). The results in Table 3 show that the baseline training was unsatisfactory,

System	BLEU	sacreBLEU
BASE, DEEP-30L	0.55	0.2
+R-Drop	0.97	0.5
+DDR	1.01	0.4
+ALD	1.45	0.7
+CAD	28.35	27.2
BASE, DEEP-30L-Full	29.09	28.1
-CAD	1.39	0.7
-DDR	28.77	27.6
-ALD	28.52	27.4

Table 3: Ablation studies on model structures and training approaches.

even with the addition of the better training methods (R-Drop, DDR, and ALD). However, when we dropped cross-attention after applying CAD, the model training became normal, indicating that the model structure has a significant impact on its performance. When we compared the results of BASE, DEEP-30L+CAD with those of BASE, DEEP-30L-Full, we found that the training methods DDR and CAD were beneficial to improving performance, demonstrating their effectiveness.

We also conducted ablation evaluation of the model structure and training method on the entire model. According to the results, CAD had the greatest influence on the translation performance, which is consistent with the conclusion stated above, based on the results in Table 3. Additionally, when comparing DDR and ALD, we found that ALD had a greater influence on translation because it directly mimics the deep-decoder collapse problem, whereas DDR is mostly employed to increase the stability of the training of the drop-net mechanism in CAD, by incorporating regularization.

6 Conclusion

In this paper, we investigated the problem of deep-decoder collapse in NMT when the decoder is deepened. We introduced a CAD mechanism, DDR loss, and ALD loss to solve this problem. Using this model, we demonstrated that a deep model with balanced numbers of encoder and decoder layers outperforms either encoder deepen only or decoder deepen only NMT models. Our model outperformed previous similar models on the WMT14 En→De and En→Fr tasks, confirming the effectiveness of our approach. For future work, we intend to incorporate methods from related work on deep NMT to further improve the performance of our translation model.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *EMNLP*, pages 3028–3033.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607.

Ronen Eldan and Ohad Shamir. 2016. The power of depth for feedforward neural networks. In *COLT*, volume 49, pages 907–940.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. In *ICLR*.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*, volume 70, pages 1243–1252.

John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180.

Xiang Kong, Adithya Renduchintala, James Cross, Yuqing Tang, Jiatao Gu, and Xian Li. 2021. Multilingual neural machine translation with deep encoder and multiple shallow decoders. In *EACL*, pages 1613–1624.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2017. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nat.*, 521(7553):436–444.

Bei Li, Ziyang Wang, Hui Liu, Quan Du, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2021. Learning light-weight translation models from deep transformer. In *AAAI*, pages 13217–13225.

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

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. WeChat neural machine translation systems for WMT20. In *WMT*, pages 239–247.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. 2016. Learning functions: when is deep better than shallow. *arXiv preprint arXiv:1603.00988*.

Mengqi Miao, Fandong Meng, Yijin Liu, Xiao-Hua Zhou, and Jie Zhou. 2021. Prevent the language model from being overconfident in neural machine translation. In *ACL*, pages 3456–3468.

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*, pages 6706–6716.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*, pages 1–9.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *ACL-IJCNLP*, pages 244–258.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

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

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL*, pages 464–468.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Matus Telgarsky. 2016. benefits of depth in neural networks. In *COLT*, volume 49, pages 1517–1539.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *ECCV*, volume 12356, pages 776–794.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *ACL*, pages 1810–1822.

Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. 2020. Multiscale collaborative deep models for neural machine translation. In *ACL*, pages 414–426.

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

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2020a. The volctrans machine translation system for WMT20. In *WMT*, pages 305–312.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the WMT20 news translation task. In *WMT*, pages 313–319.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020c. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *EMNLP-IJCNLP*, pages 898–909.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans machine translation systems for WMT20. In *WMT*, pages 338–345.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6001–6011.

A Contrastive Learning in NLP

Contrastive learning (Hadsell et al., 2006) is an effective approach to learning and is usually used for unsupervised learning because of its unique characteristics. It has achieved significant success in various computer vision tasks (Misra and van der Maaten, 2020; Zhuang et al., 2019; Tian et al., 2020; He et al., 2020; Chen et al., 2020). Gao et al. (2021) introduced a simple contrastive learning framework for unsupervised learning of sentence embedding, which performed as well as previous supervised approaches. Wu et al. (2020c) employed multiple sentence-level augmentation strategies—such as word and span deletion, reordering, and substitution—with a sentence-level contrastive learning objective to pretrain a language model for a noise-invariant sentence representation. Fang et al. (2020) pretrained language representation models using contrastive self-supervised learning at the sentence level by predicting whether two back-translated sentences originate from the same sentence. In (Giorgi et al., 2020), a universal sentence embedding encoder was trained to minimize the distance between the embeddings of textual segments randomly sampled from nearby locations in the same document by a self-supervised contrastive objective. Pan et al. (2021) demonstrated the effectiveness of contrastive learning in NMT, particularly for the zero-shot machine translation situation. Current contrastive learning for NMT primarily employs cross-lingual representation similarity, whereas we aim to prevent the outputs of the deep decoder and the unconditional language model from becoming too similar, thus preventing degradation. Part of our method is similar to (Miao et al., 2021) in purpose, but it is designed to avoid the NMT model from over-confident, while ours is to tackle the problem of the deep decoder collapsing into an unconditional language model.