DeepNet: Scaling Transformers to 1,000 Layers

Hongyu Wang[*] **Shuming Ma**[*] **Li Dong** **Shaohan Huang** **Dongdong Zhang** **Furu Wei**[†]
Microsoft Research
https://github.com/microsoft/unilm

Abstract

In this paper, we propose a simple yet effective method to stabilize extremely deep Transformers. Specifically, we introduce a new normalization function (DEEPNORM) to modify the residual connection in Transformer, accompanying with theoretically derived initialization. In-depth theoretical analysis shows that model updates can be bounded in a stable way. The proposed method combines the best of two worlds, i.e., good performance of Post-LN and stable training of Pre-LN, making DEEPNORM a preferred alternative. We successfully scale Transformers up to 1,000 layers (i.e., 2,500 attention and feed-forward network sublayers) without difficulty, which is one order of magnitude deeper than previous deep Transformers. Remarkably, on a multilingual benchmark with 7,482 translation directions, our 200-layer model with 3.2B parameters significantly outperforms the 48-layer state-of-the-art model with 12B parameters by 5 BLEU points, which indicates a promising scaling direction.
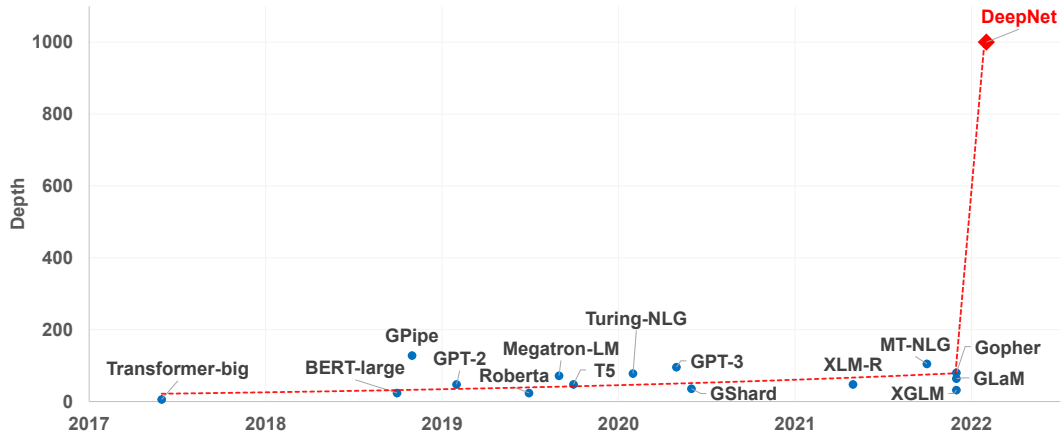
Figure 1: Trend of Transformer depths of state-of-the-art NLP models over time.

1 Introduction

Recent years have witnessed a trend towards large-scale Transformer (Vaswani et al., 2017) models. The capacity has substantially increased from millions of parameters (Devlin et al., 2019; Conneau et al., 2020) to billions (Radford et al., 2019; Brown et al., 2020; Huang et al., 2019; Raffel et al., 2020; Lepikhin et al., 2021; Rae et al., 2021; Lin et al., 2021; Smith et al., 2022), and even trillions (Du et al., 2021). Large-scale models yield state-of-the-art performance on a wide range of tasks, and show impressive abilities in few-shot and zero-shot learning. Despite an enormous number of parameters, their depths (as shown in Figure 1) are limited by the training instability of Transformers.

Nguyen and Salazar (2019) find that pre-norm residual connections (Pre-LN) improve the stability of Transformers based on post-norm connections (Post-LN). However, the gradients of Pre-LN at bottom layers tend to be larger than at top layers (Shleifer et al., 2021), leading to a degradation in performance compared with Post-LN. In order to alleviate the above issue, there have been efforts on improving the optimization of deep Transformer by means of better initialization (Zhang et al., 2019a;b; Huang et al., 2020), or better architecture (Wang et al., 2019; Liu et al., 2020; Bachlechner et al., 2020; Shleifer et al., 2021). These approaches can stabilize a Transformer model with up to hundreds of layers. Yet, none of previous methods has been successfully scaled to 1,000 layers.

Our aim is to improve the training stability of Transformers and scale the model depth by orders of magnitude. To this end, we study the cause of unstable optimization, finding the exploding model update is responsible for the instability. Motivated by the above observation, we introduce a new normalization function (DEEPNORM) at residual connections (He et al., 2016), which has theoretical justification of bounding the model update by a constant. The proposed method is simple yet effective, with just lines of code change. The approach improves the stability of Transformers so that we are able to scale model depth to more than 1,000 layers. Moreover, experimental results show that DEEPNORM combines the best of two worlds, i.e., good performance of Post-LN and stable training of Pre-LN. The proposed method can be a preferred alternative of Transformers, not only for extremely deep (such as >1000 layers) models, but also for existing large models. Notably, our 200-layer model with 3.2B parameters achieves 5 BLEU improvement on a massively multilingual machine translation benchmark compared to state-of-the-art model (Fan et al., 2021) with 48 layers and 12B model size.

2 TL;DR for Practitioners

```
def deepnorm(x):
    return LayerNorm(x * α + f(x))

def deepnorm_init(w):
    if w is ['ffn', 'v_proj', 'out_proj']:
        nn.init.xavier_normal_(w, gain=β)
    elif w is ['q_proj', 'k_proj']:
        nn.init.xavier_normal_(w, gain=1)
```

Architectures	Encoder		Decoder	
	α	β	α	β
Encoder-only (e.g., BERT)	$(2N)^{\frac{1}{4}}$	$(8N)^{-\frac{1}{4}}$	-	-
Decoder-only (e.g., GPT)	-	-	$(2M)^{\frac{1}{4}}$	$(8M)^{-\frac{1}{4}}$
Encoder-decoder (e.g., NMT, T5)	$0.81(N^4M)^{\frac{1}{16}}$	$0.87(N^4M)^{-\frac{1}{16}}$	$(3M)^{\frac{1}{4}}$	$(12M)^{-\frac{1}{4}}$

Figure 2: (a) Pseudocode for DEEPNORM. We take Xavier initialization (Glorot and Bengio, 2010) as an example, and it can be replaced with other standard initialization. Notice that α is a constant. (b) Parameters of DEEPNORM for different architectures (N-layer encoder, M-layer decoder).

As shown in Figure 2, it is simple to implement our method based on Transformers with Post-LN. Compared to Post-LN, DEEPNORM up-scales the residual connection before performing layer normalization. Besides, we down-scale the parameters during initialization. Notably, we only scale the weights of feed-forward networks, as well as the value projection and the output projection of attention layers. Moreover, the scales of residual connection and initialization are dependent on the architecture (Figure 2). We provide more details in Section 4.3.

3 Instability of Deep Transformer

We study the causes of the instability for deep Transformers. Our analysis begins with the observation: better initialization methods stabilize the training of Transformer. This has also been verified by previous work (Zhang et al., 2019a; Huang et al., 2020; Xu et al., 2021). Therefore, we study the
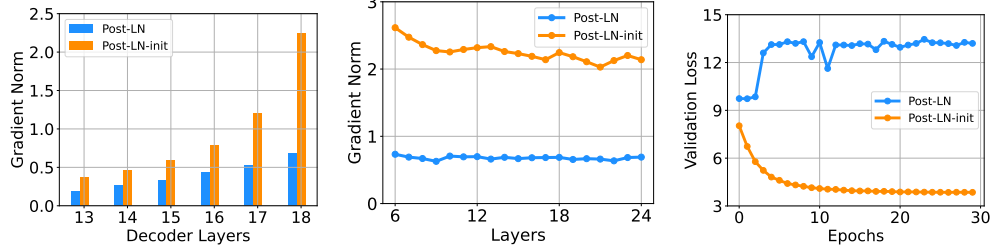
Figure 3: (a) Gradient norm in the top layers of 18L-18L models. (b) Gradient norm in the last layer of the models with depths varying from 6L-6L to 24L-24L. (c) Validation loss curves of 18L-18L models.

(a) Accumulated model update   (b) Input from FFN to LN   (c) Input from attention to LN
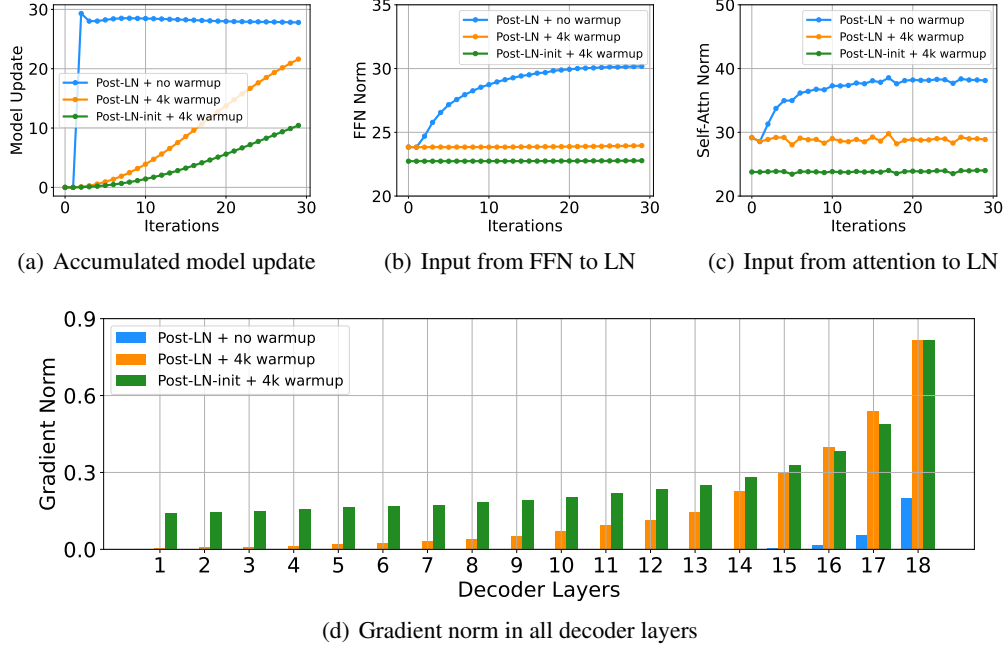

(d) Gradient norm in all decoder layers

Figure 4: Visualization of the model update, the average input of LNs, and the gradients for the 18L-18L models at the early stage of training.

training process of Post-LN with or without proper initialization. With better initialization, we down-scale the weights of l-th layer by $k_l = N - l + 1, l \in [1, N]$ after performing Xavier initialization. For example, the output projection W_o^l of FFN in l-th layer is initialized as:

$$W_o^l \backsim \mathcal{N}\left(0, \frac{1}{k_l^2 d'}\right),$$

where d' is an average of input and output dimensions. We name this model Post-LN-init. Notice that different from the prior work (Zhang et al., 2019a), we narrow the scale of lower layers instead of the higher layers. We believe that it helps to separate the effect of the gradient scale from the model update. Besides, Post-LN-init has the same architecture as Post-LN, which eliminates the impact from the architecture.

We train 18L-18L Post-LN and 18L-18L Post-LN-init on the IWSLT-14 De-En machine translation dataset. Figure 3 visualizes their gradients and validation loss curves. As shown in Figure 3(c), Post-LN-init converged while Post-LN did not. Post-LN-init has an even larger gradient norm in the last several layers, although its weights have been scaled down. Furthermore, we visualize the

gradient norm of the last decoder layer with varying model depth from 6L-6L to 24L-24L. Figure 3 shows that the gradient norm of Post-LN-init in the last layer is still much larger than that of Post-LN, regardless of model depth. It concludes that the exploding gradients in deep layers should not be the root cause of instability of Post-LN, while the scale of model update tends to account for it.

Then we demonstrate that the instability of Post-LN comes from a chain of several issues, including gradient vanishing as well as too large model updates. As shown in Figure 4(a), we first visualize the norm of model update $||\Delta F||$ at the early stage of training:

$$||\Delta F|| = ||F(x, \theta_i) - F(x, \theta_0)||,$$

where x and θ_i denotes input, and model parameters after i-th updates. Post-LN has an exploding update at the very beginning of training, and then nearly no update shortly. It indicates that the model has been stuck in a spurious local optima. Both warm-up and better initialization help alleviate this issue, enabling the model to update smoothly. When the update explodes, the inputs to LN become large (see Figure 4(b) and Figure 4(c)). According to the theoretical analysis from Xiong et al. (2020), the magnitude of gradient through LN is inversely proportional to the magnitude of its input:

$$||\frac{\partial LN(x)}{\partial x}|| = \mathcal{O}(\frac{\sqrt{d}}{||x||}).$$

Figure 4(b) and Figure 4(c) show that $||x||$ is significantly larger than $\sqrt{d}$ ($d = 512$) without warm-up or proper initialization. This explains the gradient vanishing problem occurred in the training of Post-LN (see Figure 4(d)).

Above all, the instability starts from the large model update at the beginning of training. It renders the model trapped in a bad local optima, which in turn increases the magnitude of inputs to each LN. As training continues, the gradient through LN becomes increasingly small, thus resulting in severe gradient vanishing. The vanishing gradients make it difficult to escape from the local optima, and further destabilize the optimization. On the contrary, Post-LN-init has relatively small updates, and the inputs to LN are stable. This relieves suffering from gradient vanishing, making optimization more stable.

4 DEEPNET: Extremely Deep Transformers

In this section, we introduce our extremely deep Transformers named DEEPNET. It can stabilize the optimization by mitigating the exploding model update problem. We first provide the estimation of the expected magnitude of DEEPNET's model update. Then we provide the theoretical analysis to show that its updates can be bounded by a constant with our proposed DEEPNORM.

4.1 Architecture

DEEPNET is based on the Transformer architecture. Compared to the vanilla Transformer, it uses our new DEEPNORM, instead of Post-LN, for each sub-layer. The formulation of DEEPNORM can be written as:

$$x_{l+1} = LN(\alpha x_l + G_l(x_l, \theta_l)),$$

where α is a constant, and $G_l(x_l, \theta_l)$ is the function of the l-th Transformer sub-layer (i.e., attention or feed-forward network) with parameters θ_l. Besides, DEEPNET scales the weights θ_l inside residual branches by β. Notably, both α and β are constants that only depend on the architecture, and we provide the derivation in Section 4.3.

4.2 Expected Magnitude of Model Update

Attention is an important part of Transformer. Without loss of generality, we study the 1-head case. Let $Q, K, V \in \mathbf{R}^{n \times d}$ denote the query, key, value, respectively. $W^Q, W^K, W^V \in \mathbf{R}^{d \times d_k}$ are the input projection matrices, and $W^O \in \mathbf{R}^{d_k \times d}$ is the output projection matrix. Then, the attention module can be formulated as:

$$Attn(Q, K, V) = softmax(\frac{QW^Q(KW^K)^T}{\sqrt{d_k}})VW^VW^O$$

4

We study the magnitude of the attention module. Lemma 4.1 proves that W^Q and W^K do not change the bound of attention output's magnitude.

Lemma 4.1. *Given* $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n})^T \in \mathbf{R}^{n \times d}$, *where* $var(\mathbf{x_i}) = 1$, $mean(\mathbf{x_i}) = 0$ *and* $q_i \in \mathbf{R}$ *for all* $i \in [1, n]$, *it satisfies that*

$$softmax(q_1, q_2, ..., q_n)\mathbf{X} \stackrel{\Theta}{=} \mathbf{x_i},$$

where $\stackrel{\Theta}{=}$ *stands for equal bound of magnitude.*

In other words, the magnitude of attention output only depends on the value and output projection: $Attn(Q, K, V) \stackrel{\Theta}{=} V W^V W^O$. In this work, we only consider the magnitude of model update, so it is sufficiently instructive to study the case where the hidden dimension equals to 1. For simplicity, we reduce the matrices W^V, W^O to the scalars v, w, which means $Attn(Q, K, V) \stackrel{\Theta}{=} vwV$. Similarly, we have $FFN(X) \stackrel{\Theta}{=} vwX$, where v, w denotes the parameters of the feed-forward network.

We define the model update as $||\Delta F|| = ||F(x, \theta^*) - F(x, \theta)||$. Based on the analysis above, we have the following theorem to characterize $||\Delta F||$'s magnitude of an N-layer DEEPNET with N attentions and FFNs.

Theorem 4.2. *Given an N-layer DEEPNET $F(x, \theta)$ ($\theta = \{\theta_1, \theta_2, ..., \theta_{2N}\}$), where θ_{2l-1} and θ_{2l} denote the parameters of self-attention and FFN in l-th layer, and each sub-layer is normalized with DEEPNORM: $x_{l+1} = LN(\alpha x_l + G_l(x_l, \theta_l))$, $||\Delta F||$ satisfies:*

$$||\Delta F|| \leq \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha} ||\theta_i^* - \theta_i||$$

Vanilla Post-LN can be regarded as a special case of DEEPNET, where $\alpha = 1$ and $v_l = w_l = 1$ at Xavier initialization (Glorot and Bengio, 2010). Based on Theorem 4.2, we have $||\Delta F|| = \mathcal{O}(\sum_{i=1}^{2N} ||\theta_i^* - \theta_i||)$ for vanilla Post-LN. It shows that the model tends to accumulate the update of each sub-layer, which leads to exploding magnitude of model's update and destabilizes the optimization at the early stage. This explains our findings in Section 3.

Besides, Theorem 4.2 also explains why warm-ups and smaller initialization can stabilize the training of Post-LN. Warm-ups can reduce the magnitude of the model update by decreasing $||\theta_i^* - \theta_i||$, while smaller initialization lowers $\sqrt{v_i^2 + w_i^2}$.

Furthermore, we study the magnitude of DEEPNET with an N-layer encoder and an M-layer decoder. Let $F_{ed}(x, y, \theta_e, \theta_d)$ denotes the model, where x, y is the input of encoder and decoder. θ_e follows the same definition as θ in Theorem 4.2. $\theta_d = \{\theta_{d1}, \theta_{d2}, ..., \theta_{d,3M}\}$ stands for the parameters of self-attentions, cross-attentions, and FFNs. We use $\{\alpha_e, G_{el}\}$ and $\{\alpha_d, G_{dl}\}$ to distinguish the notations between the encoder and the decoder. The following theorem shows the expected magnitude of the encoder-decoder's model update $||\Delta F_{ed}|| = ||F_{ed}(x, y, \theta_e^*, \theta_d^*) - F_{ed}(x, y, \theta_e, \theta_d)||$.

Theorem 4.3. *Given an encoder-decoder DEEPNET $F_{ed}(x, y, \theta_e, \theta_d)$ with N encoder layers and M decoder layers, where each encoder sub-layer is normalized as $x_{l+1} = LN(\alpha_e x_l + G_{el}(x_l, \theta_{el}))$, and the decoder sub-layer is normalized as $x_{l+1} = LN(\alpha_d x_l + G_{dl}(x_l, \theta_{dl}))$, $||\Delta F_{ed}||$ satisfies:*

$$
\begin{aligned}
||\Delta F_{ed}|| \leq &\sum_{j=1}^{M} \frac{v_{d,3j-1} w_{d,3j-1}}{\alpha_d} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e} ||\theta_{ei}^* - \theta_{ei}|| \\
&+ \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d} ||\theta_{dj}^* - \theta_{dj}||
\end{aligned} \tag{1}
$$

The vanilla encoder-decoder model satisfies that all of $\{\alpha_e, \alpha_d, v_{ei}, w_{ei}, v_{di}, w_{di}\}$ equal to 1, so we have $||\Delta F_{ed}|| = \mathcal{O}(M \sum_{i=1}^{2N} ||\theta_{ei}^* - \theta_{ei}|| + \sum_{j=1}^{3M} ||\theta_{dj}^* - \theta_{dj}||)$. It indicates the similar accumulative effect which leads to fast growth of the magnitude regarding the model depth (see Figure 5). Furthermore, the cross-attention propagates the magnitude from the encoder to the decoder, which explains why the decoder is more unstable than the encoder (Liu et al., 2020).
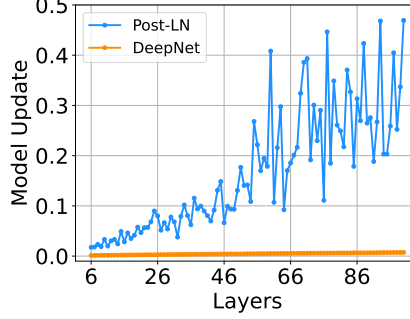
5

Figure 5: Model updates of vanilla Post-LN and DEEPNET at the early stage of training. The visualization is conducted on 64-128-2 tiny Transformers with depth varying from 6L-6L to 100L-100L. It shows that DEEPNET has much smaller and more stable updates than Post-LN.

4.3 Derivation for DEEPNORM and the Initialization

We show that the expected model updates for DEEPNET can be bounded by a constant with proper parameters α and β. Our analysis is based on SGD update, and we empirically verify it works well for Adam optimizer (Kingma and Ba, 2015). We provide the analysis on the encoder-decoder architecture, which can be naturally extended to encoder-only and decoder-only models in the same way. Analogous to Zhang et al. (2019b), we set our goal for the model update as follows:

> **GOAL:** $F_{ed}(x, y, \theta_e, \theta_d)$ is updated by $\Theta(\eta)$ per SGD step after initialization as $\eta \to 0$. That is $||\Delta F_{ed}|| = \Theta(\eta)$ where $\Delta F_{ed} \triangleq F_{ed}(x, y, \theta_e - \eta\frac{\partial \mathcal{L}}{\partial \theta_e}, \theta_d - \eta\frac{\partial \mathcal{L}}{\partial \theta_d}) - F_{ed}(x, y, \theta_e, \theta_d)$.

For SGD optimizer, the update of each decoder layer $||\theta_{di}^* - \theta_{di}||$ equals to $\eta||\frac{\partial \mathcal{L}}{\partial \theta_{di}}||$. Xiong et al. (2020) proved that Post-LN decreases the magnitude of backpropagating error signal, so we have $||\frac{\partial F}{\partial \theta_{dj}}|| \leq ||\frac{\partial F}{\partial \theta_{d,3M}}||$. With $||\frac{\partial F}{\partial \theta_{d,3M}}|| \overset{\Theta}{=} \frac{||\theta_{d,3M}||}{\alpha_d}$ and the assumption $||\frac{\partial \mathcal{L}}{\partial F}|| = \mathcal{O}(1)$, the second term of Equation (1) can be bounded as:

$$\sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d}||\theta_{dj}^* - \theta_{dj}|| \leq \eta||\frac{\partial \mathcal{L}}{\partial F}|| \cdot ||\frac{\partial \mathcal{F}}{\partial \theta_{d,3M}}|| \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d}$$
$$\overset{\Theta}{=} 3\eta M \frac{v_d^2 + w_d^2}{\alpha_d^2} \tag{2}$$

There are multiple schemes to bound Equation (2) by $\Theta(\eta)$. In order to balance the effect of residual connections and the initialization, we set $\alpha_d^2 = (3M)^{\frac{1}{2}}$, $v_d^2 + w_2^d = (3M)^{\frac{1}{2}}$ and $v_d = w_d = \beta_d$ due to symmetry, that is $\alpha_d = (3M)^{\frac{1}{4}}$, $\beta_d = (12M)^{-\frac{1}{4}}$. Similarly, we use $v_e = w_e = \beta_e = 0.87(N^4M)^{-\frac{1}{16}}$, $\alpha_e = 0.81(N^4M)^{\frac{1}{16}}$ to bound the first term in Equation (1). Detailed derivation is shown in Appendix B.

In comparison with Post-LN, we visualize the model updates for DEEPNET on IWSLT-14 De-En translation dataset at the early training stage. Figure 5 shows that the model update of DEEPNET is nearly constant, while the model update of Post-LN is exploding.

In summary, we apply our approach as follows:

Models	LN	6L-6L	18L-18L	50L-50L	100L-100L
Vanilla Post-LN (Vaswani et al., 2017)	Post	**28.1**		diverged	
DS-Init (Zhang et al., 2019a)	Post	27.9		diverged	
Admin (Liu et al., 2020)	Post	27.9	**28.8**	diverged	
ReZero (Bachlechner et al., 2020)	No	26.9		diverged	
R-Fixup (Zhang et al., 2019b)	No	27.5	28.4	27.7	diverged
T-Fixup (Huang et al., 2020)	No	27.5	28.4	27.9	diverged
Vanilla Pre-LN (Vaswani et al., 2017)	Pre	27.0	28.1	28.0	27.4
DLCL (Wang et al., 2019)	Pre	27.4	28.2	diverged	27.5
NormFormer (Shleifer et al., 2021)	Pre	27.0	28.3	27.8	diverged
DEEPNET (ours)	Deep	27.8	**28.8**	**29.0**	**28.9**

Table 1: BLEU scores on the WMT-17 En-De test set for different models with varying depth. AL-BL refers to A-layer encoder and B-layer decoder.

Encoder-decoder architecture

1. Apply standard initialization (e.g., Xavier initialization) for each encoder and decoder layer.

2. For encoder layers, scale the weights of feed-forward networks as well as the value projection and the output projection of attention layers by $0.87(N^4M)^{-\frac{1}{16}}$, and set the weight of residual connections as $0.81(N^4M)^{\frac{1}{16}}$.

3. For decoder layers, scale the weights of feed-forward networks as well as the value projection and the output projection of attention layers by $(12M)^{-\frac{1}{4}}$, and set the weight of residual connections as $(3M)^{\frac{1}{4}}$.

The derivation of encoder-only (such as BERT) and decoder-only (such as GPT) architectures can be conducted in the same way (see Appendix C). We summarize the steps as follows:

Encoder-only (or decoder-only) architecture

1. Apply standard initialization (e.g., Xavier initialization) for each layer.

2. For each layer, scale the weights of feed-forward networks as well as the value projection and the output projection of attention layers by $(8N)^{-\frac{1}{4}}$ (or $(8M)^{-\frac{1}{4}}$), and set the weight of residual connections as $(2N)^{\frac{1}{4}}$ (or $(2M)^{\frac{1}{4}}$).

---

5 Neural Machine Translation

We verify the effectiveness of DEEPNET on the popular machine translation benchmarks, including IWSLT-14 German-English (De-En) dataset and WMT-17 English-German (En-De) dataset. We compare our method with multiple state-of-the-art deep Transformer models, including DLCL (Wang et al., 2019), NormFormer (Shleifer et al., 2021), ReZero (Bachlechner et al., 2020), R-Fixup (Zhang et al., 2019b), T-Fixup (Huang et al., 2020), DS-init (Zhang et al., 2019a), and Admin (Liu et al., 2020). We reproduce the baselines with their open-source code, and set the hyper-parameters the same for a fair comparison.

We use BLEU as the evaluation metric for all experiments. Table 1 reports the results of the baselines and DEEPNET on WMT-17 En-De translation dataset. According to their LNs, the baselines are grouped into three categories: Pre-LN, Post-LN, and No-LN. All the compared models are base-size with different depths.

Compared with the models with Post-LN, DEEPNET is more stable, and can successfully scale to 100L-100L, reaching the 28.9 BLEU on the test set. In contrast, the baselines with Post-LN lead to
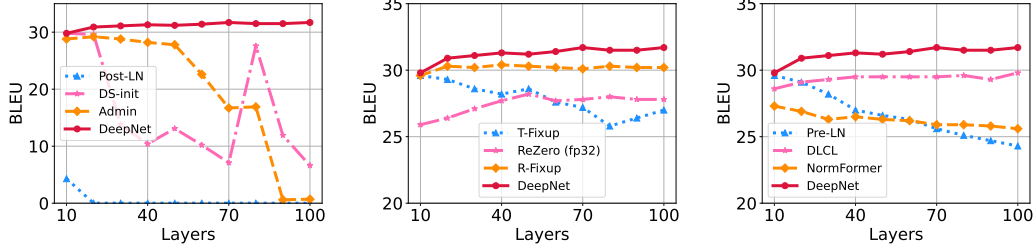
Figure 6: BLEU scores on the IWSLT-14 De-En test set for different deep models with varing depth from 10L-10L to 100L-100L.
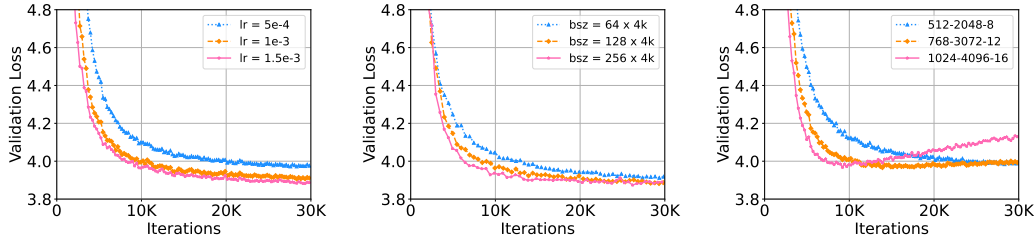

Figure 7: WMT-17 En-De validation loss curves for 18L-18L DEEPNET with varing learning rate, batch size and hidden dimension.

unstable optimization when the depth goes to 50L-50L. Besides, DEEPNET achieves comparable performance with these baselines when the models are shallow.

In addition, we compare DEEPNET with the methods without LN. Both R-Fixup and T-Fixup introduce better initialization methods, which stabilize the training of No-LN Transformer with up to 50-50 layers. Yet, their performance is not as good as those with Post-LN. Besides, half-precision could destabilize the training of ReZero, leading to its divergence with 18-18 layers. This observation is also reported by Liu et al. (2020). Moreover, deeper models (50L-50L) do not outperform the shallow models (18L-18L). In comparison, DEEPNET achieves better translation accuracy than these methods, and scaling to deeper models brings no harm to the performance.

Compared with the Post-LN baselines, the models with Pre-LN are more stable. Both vanilla Pre-LN and DLCL can be scaled to 100L-100L, and 50L-50L NormFormer is also trained successfully. Nevertheless, Pre-LN leads to a 0.5-1.0 BLEU drop compared with the converged Post-LN models. We presume this should be caused by the problem that gradients of Pre-LN at earlier layers tend to be larger than gradients at later layers (Shleifer et al., 2021). We leave it as the future work. In contrast, DEEPNET alleviates the problem by using Post-LN, and outperforms all the Pre-LN baselines.

Convergence with varying depth. We vary the depths of the models from 10L-10L to 100L-100L with an interval of 10 layers.All experiments are conducted with mixed precision training, except ReZero[3]. Figure 6 shows the results on the IWSLT-14 dataset. We train the models for 8,000 steps because we find most divergence occurs at the beginning of optimization. Overall, DEEPNET is stable from shallow to deep. It converges fast, achieving over 30 BLEU in only 8,000 steps while most of the baselines do not. Moreover, the performance keeps improving as the model goes deeper.

Large learning rate, batch size, and hidden dimension. We further scale DEEPNET to larger learning rate, batch size, and hidden dimension, respectively. For each experiment, we only change one hyperparameter with the others fixed. Figure 7 reports the loss curves on the WMT-17 validation set. It shows that DEEPNET can be trained without difficulty in all the largest settings. The loss of DEEPNET with 1024 hidden size increases after 10K steps because of overfitting. Besides, it indicates that DEEPNET can benefit from the larger settings, resulting in faster convergence and lower validation loss.

---

[3]According to our experiments, ReZero is unstable with half precision, even when the model is shallow.
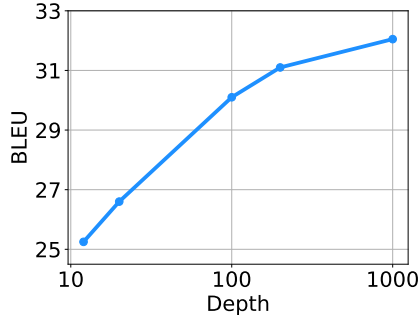
Figure 8: Average BLEU scores for DEEPNET with varying depth on the OPUS-100 En-X and X-En test sets.

Models	# Layers	# Params	X→En	En→X	Avg
Baseline (Zhang et al., 2020)	12	133M	27.5	21.4	24.5
	24	173M	29.5	22.9	26.2
	48	254M	31.4	24.0	27.7
DEEPNET (ours)	200	863M	33.2	29.0	31.1
	1000	3.8B	**33.9**	**30.2**	**32.1**

Table 2: Average BLEU for DEEPNET and the baseline on the OPUS-100 test sets.

6 Massively Multilingual Neural Machine Translation

We conduct experiments on the large-scale multilingual machine translation, which is a good testbed for large models. We first use OPUS-100 corpus (Zhang et al., 2020) to evaluate our model. OPUS-100 is an English-centric multilingual corpus covering 100 languages, which is randomly sampled from the OPUS collection. We scale DEEPNET up to 1,000 layers. The model has a 500-layer encoder, a 500-layer decoder, 512 hidden size, 8 attention head, and 2,048 dimensions of feed-forward layers. More details can be found in the Appendix.

Table 2 summarizes the results of DEEPNET and the baselines. It shows that increasing the depth can significantly improve the translation quality of NMT: the baseline of 48 layers achieves a gain of 3.2 points on average over the 12-layer model. DEEPNET can successfully scale up the depth to 1,000 layers, outperforming the baseline by an improvement of 4.4 BLEU. It is noted that DEEPNET is only trained for 4 epochs, and the performance can be further improved given more computation budgets.

Scaling law in terms of depth We train DEEPNET of {12, 20, 100, 200, 1000} layers on the OPUS-100 dataset. Figure 8 illustrates the scaling curve. Compared with bilingual NMT, multilingual NMT benefits more from scaling the depth of the model because of its hunger in model capacity. We observe logarithmic growth of the BLEU score for multilingual NMT, and the scaling law can be written as:

$$L(d) = A \log(d) + B$$

where d is the depth, and A, B are the constants regarding the other hyper-parameters.

More data and language directions. To explore the limits of DEEPNET on multilingual NMT, we then scale up the training data by using CCMatrix (Schwenk et al., 2021). We also expand the data from CCAligned (El-Kishky et al., 2020), OPUS (Zhang et al., 2020), and Tatoeba[4] to cover all languages of Flores101 evaluation sets. The final data consists of 102 languages, 1932 directions, and 12B sentence pairs. With the data, we train DEEPNET with a 100-layer encoder, 100-layer decoder, 1,024 hidden dimension, 16 heads, and 4,096 intermediate dimension of feed-forward layers. More details can be found in the Appendix.

---

[4] https://tatoeba.org/en/

Models	# Layers	# Params	WMT	OPUS	TED	Flores
M2M-100 (Fan et al., 2021)	48	12B	31.9	18.4	18.7	13.6
DEEPNET (ours)	200	3.2B	**33.9**	**23.0**	**20.1**	**18.6**

Table 3: BLEU scores for DEEPNET and M2M-100 on various evaluation sets.

We compare DEEPNET with the state-of-the-art multilingual NMT model M2M-100 (Fan et al., 2021). M2M-100 has a 24-layer encoder, a 24-layer decoder, and 4,096 hidden size, resulting in up to 12B parameters. Compared with M2M-100, DEEPNET is deep and narrow with only 3.2B parameters. For a fair comparison, we generate the model with beam size 5 and length penalty 1.

Following M2M-100 (Fan et al., 2021), we evaluate the models on several multilingual translation evaluation datasets, including WMT (Bojar et al., 2014; 2017; 2018; Barrault et al., 2019), OPUS (Zhang et al., 2020), TED (Qi et al., 2018), and Flores (Goyal et al., 2021). The language pairs from the WMT dataset are English-centric. There are 10 languages including English, and most of them are high-resource. For the OPUS dataset, we select the non-English directions from the test set, which has 30 evaluation pairs. The TED evaluation set has 28 languages and 756 directions, and the data is from the spoken language domain. The Flores dataset has all translation pairs between 102 languages. We use a subset covering the languages supported by both M2M-100 and DEEPNET, resulting in 87 languages and 7,482 translation directions.

We report the results in Table 3. For a fair comparison, we use the same evaluation methods as the baseline. The details can be found in the Appendix. It shows that DEEPNET has significantly better performance than M2M-100 on all evaluation datasets, indicating that deepening the model is a very promising direction to improve the quality of NMT models.

7 Conclusion and Future Work

We improve the stability of Transformer and successfully scale it to 1,000 layers. This is achieved by our DEEPNET with a novel normalization function called DEEPNORM. It has theoretical justification to stabilize the optimization with a constant upper bound for model updates. Experimental results verify the effectiveness of our methods across various benchmarks. We focus on machine translation as a test bed in the current experiments. In the future, we will extend DEEPNET to support more diverse tasks, e.g., language model pre-training (Dong et al., 2019; Bao et al., 2020; Chi et al., 2021a; Ma et al., 2021; Chi et al., 2021b), protein structure prediction (Jumper et al., 2021), and BEiT vision pre-training (Bao et al., 2022; Wang et al., 2021).

References

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian J. McAuley. Rezero is all you need: Fast convergence at large depth. *CoRR*, abs/2003.04887, 2020.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. PMLR, 2020.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

Loïc Barrault, Ondrej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André

Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 1–61. Association for Computational Linguistics, 2019.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 12–58. The Association for Computer Linguistics, 2014.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 conference on machine translation (WMT17). In Ondrej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, and Julia Kreutzer, editors, *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 169–214. Association for Computational Linguistics, 2017.

Ondrej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 272–303. Association for Computational Linguistics, 2018.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS 2020*, 2020.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *NAACL-HLT 2021*, pages 3576–3588, 2021a.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. XLM-E: cross-lingual language model pre-training via ELECTRA. *CoRR*, abs/2106.16138, 2021b.

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 Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *ACL 2020*, pages 8440–8451, 2020.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *NeurIPS 2019*, pages 13042–13054, 2019.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen,

and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. *CoRR*, abs/2112.06905, 2021.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. CCAligned: A massive collection of cross-lingual web-document pairs. In Bonnie Webber, Trevor Cohn, Yu-lan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5960–5969. Association for Computational Linguistics, 2020.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48, 2021.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS 2010*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Xiao Shi Huang, Felipe Pérez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483, 2020.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Xu Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *NeurIPS 2019*, pages 103–112, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596 (7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR 2015*, 2015.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR 2021*, 2021.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668, 2021.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, 2020.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. DeltaLM: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736, 2021.

Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. *CoRR*, abs/1910.05895, 2019.

Matt Post. A call for clarity in reporting BLEU scores. In Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics, 2018.

Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 529–535. Association for Computational Linguistics, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. CCMatrix: Mining billions of high-quality parallel sentences on the web. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6490–6500. Association for Computational Linguistics, 2021.

Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization. *CoRR*, abs/2110.09456, 2021.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990, 2022.

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

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

Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *ArXiv*, abs/2111.02358, 2021.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533, 2020.

Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J. D. Prince, and Yanshuai Cao. Optimizing deeper transformers on small datasets. In *ACL/IJCNLP 2021*, pages 2089–2102, 2021.

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

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL 2020*, pages 1628–1639. Association for Computational Linguistics, 2020.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *ICLR 2019*, 2019b.

A Main Theorem Proof

A.1 Proof of Theorem 4.1

Lemma A.1. *Given* $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n})^T \in \mathbf{R}^{n \times d}$, *where* $var(\mathbf{x_i}) = 1$, $mean(\mathbf{x_i}) = 0$ *and* $q_i \in \mathbf{R}$ *for all* $i \in [1, n]$, *it satisfies that*

$$softmax(q_1, q_2, ..., q_n)\mathbf{X} \stackrel{\Theta}{=} \mathbf{x_i},$$

where $\stackrel{\Theta}{=}$ *stands for equal bound of magnitude.*

Proof. The weight s_i of $\mathbf{x_i}$ to output is $s_i = \dfrac{e^{q_i}}{e^{\sum_{j=1}^{n} q_j}}$, $\sum_{i=1}^{n} s_i = 1$.

$$||\text{softmax}(q_1, q_2, ..., q_n)\mathbf{X}|| = ||\sum_{i=1}^{n} s_i \mathbf{x_i}|| \leq \sum_{i=1}^{n} s_i ||\mathbf{x_i}|| \tag{3}$$

With $var(\mathbf{x_i}) = 1$, $mean(\mathbf{x_i}) = 0$, for all $i \in [1, n]$, we have $||\mathbf{x_i}|| = d$. Therefore, $||\text{softmax}(q_1, q_2, ..., q_n)\mathbf{X}|| \leq ||\mathbf{x_i}|| = d$, which is equivalent to $\text{softmax}(q_1, q_2, ..., q_n)\mathbf{X} \stackrel{\Theta}{=} \mathbf{x_i}$. $\square$

A.2 Proof of Theorem 4.2

Theorem A.2. *Given an* N-*layer* DEEPNET $F(x, \theta)$ *(*$\theta = \{\theta_1, \theta_2, ..., \theta_{2N}\}$*), where* θ_{2l-1} *and* θ_{2l} *denote the parameters of self-attention and FFN in* l-*th layer, and each sub-layer is normalized with* DEEPNORM: $x_{l+1} = LN(\alpha x_l + G_l(x_l, \theta_l))$, $||\Delta F||$ *satisfies:*

$$||\Delta F|| \leq \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha} ||\theta_i^* - \theta_i||$$

Proof. Our aim is to study the magnitude of model updates. Following Zhang et al. (2019b), we make the following assumptions to simplify the derivations:

1. Hidden dimension d equals to 1.

2. $var(x + G_l(x)) \stackrel{\Theta}{=} var(x) + var(G_l(x))$

3. All relevant weights v, w are positive with magnitude less than 1 and α, β for DEEPNORM are positive with magnitude greater than 1.

Given Assumption 1, if $G(x)$ is feed-forward network with $\theta = \{v, w\}$, then $G(x) \stackrel{\Theta}{=} vwx$. According to Theorem 4.1, the query and key projections do not change the bound of the attention output's magnitude. Therefore, if $G(x)$ is self-attention with $\theta = \{q, k, v, w\}$, then $G(x) \stackrel{\Theta}{=} vwx$. Especially, if Xavier initialization is used for the projection, then the output can preserve the input variance, which is equivalent to $v = w = 1$. With Assumption 2, we have:

$$x_{l+1} = f_l(x_l, \theta_l) = \frac{\alpha x + G_l(x)}{\sqrt{Var(\alpha x + G_l(x))}} \stackrel{\Theta}{=} \frac{\alpha + v_l w_l}{\sqrt{\alpha^2 + v_l^2 w_l^2}} x \tag{4}$$

With Equation (4), the magnitude of $\frac{\partial f_l}{\partial x}$ and $\frac{\partial f_l}{\partial \theta_l}$ is bounded by:

$$\frac{\partial f_l}{\partial x} \stackrel{\Theta}{=} \frac{\alpha + v_l w_l}{\sqrt{\alpha^2 + v_l^2 w_l^2}}$$
$$\frac{\partial f_l}{\partial \theta_l} \stackrel{\Theta}{=} \left(\frac{\partial f}{\partial v_l}, \frac{\partial f}{\partial w_l}\right) \stackrel{\Theta}{=} \frac{\alpha x_l(\alpha - v_l w_l)}{(\alpha^2 + v_l^2 w_l^2)^{\frac{3}{2}}} (w_l, v_l) \tag{5}$$

Besides, the model update $||\Delta F||$ satisfies:

$$||\Delta F|| = ||F(x, \theta^*) - F(x, \theta)|| = ||x_{2N+1}^* - x_{2N+1}|| = ||f(x_{2N}^*, \theta_{2N}^*) - f(x_{2N}, \theta_{2N})|| \quad (6)$$

Using Taylor expansion for Equation (6), we get:

$$
\begin{aligned}
||\Delta F|| &= ||x_{2N+1}^* - x_{2N+1}|| && (7)\\
&\approx ||\frac{\partial f}{\partial x}(x_{2N}, \theta_{2N})(x_{2N}^* - x_{2N}) + \frac{\partial f}{\partial \theta}(x_{2N}, \theta_{2N})(\theta_{2N}^* - \theta_{2N})^T||\\
&\leq ||\frac{\partial f}{\partial x}(x_{2N}, \theta_{2N})|| \cdot ||x_{2N}^* - x_{2N}|| + ||\frac{\partial f}{\partial \theta}(x_{2N}, \theta_{2N})|| \cdot ||\theta_{2N}^* - \theta_{2N}||\\
&= \frac{\alpha + v_{2N}w_{2N}}{\sqrt{\alpha^2 + v_{2N}^2 w_{2N}^2}}||x_{2N}^* - x_{2N}|| + \frac{\alpha(\alpha - v_{2N}w_{2N})}{(\alpha^2 + v_{2N}^2 w_{2N}^2)^{\frac{3}{2}}}\sqrt{v_{2N}^2 + w_{2N}^2}||\theta_{2N}^* - \theta_{2N}||\\
&\approx ||x_{2N}^* - x_{2N}|| + \frac{\sqrt{v_{2N}^2 + w_{2N}^2}}{\alpha}||\theta_{2N}^* - \theta_{2N}|| && (8)
\end{aligned}
$$

Then, we have:

$$||x_{2N+1}^* - x_{2N+1}|| \leq \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha}||\theta_i^* - \theta_i|| \quad (9)$$

$\square$

For vanilla Post-LN with standard initialization, $\alpha = v_i = w_i = 1$, so $||\Delta F|| = \mathcal{O}(\sum_{i=1}^{2N} ||\theta_i^* - \theta_i||)$.

Proof of Theorem 4.3

Theorem A.3. *Given an encoder-decoder* DEEPNET *$F_{ed}(x, y, \theta_e, \theta_d)$ with N encoder layers and M decoder layers, where each encoder sub-layer is normalized as $x_{l+1} = LN(\alpha_e x_l + G_{el}(x_l, \theta_{el}))$, and the decoder sub-layer is normalized as $x_{l+1} = LN(\alpha_d x_l + G_{dl}(x_l, \theta_{dl}))$, $||\Delta F_{ed}||$ satisfies:*

$$
\begin{aligned}
||\Delta F_{ed}|| &\leq \sum_{j=1}^{M} \frac{v_{d,3j-1}w_{d,3j-1}}{\alpha_d} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e}||\theta_{ei}^* - \theta_{ei}||\\
&+ \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d}||\theta_{dj}^* - \theta_{dj}|| && (10)
\end{aligned}
$$

Proof. The derivation of self-attention and FFN layers is given in Appendix A.2. For the cross-attention layers, we have:

$$y_{l+1} = f_{dl}(y_l, x_e, \theta_{dl}) = \frac{\alpha_d y_l + G_l(x_e, y_l)}{\sqrt{Var(\alpha_d y_l + G_{dl}(x_e, y_l))}} \stackrel{\Theta}{=} \frac{\alpha_d y_l + v_l w_l x_e}{\sqrt{\alpha_d^2 + v_l^2 w_l^2}} \quad (11)$$

With Equation (11), we have the bound of the derivative of f_{dl}:

$$\frac{\partial f_{dl}}{\partial y} \stackrel{\Theta}{=} \frac{\alpha_d}{\sqrt{\alpha_d^2 + v_l^2 w_l^2}}, \quad \frac{\partial f_{dl}}{\partial x_e} \stackrel{\Theta}{=} \frac{v_l w_l}{\sqrt{\alpha_d^2 + v_l^2 w_l^2}}$$

$$\frac{\partial f_{dl}}{\partial \theta_{dl}} \stackrel{\Theta}{=} (\frac{\partial f_{dl}}{\partial v_{dl}}, \frac{\partial f_{dl}}{\partial w_{dl}}) \stackrel{\Theta}{=} \frac{\alpha_d x_e(\alpha_d - v_{dl}w_{dl})}{(\alpha_d^2 + v_{dl}^2 w_{dl}^2)^{\frac{3}{2}}}(w_{dl}, v_{dl})$$

By means of Taylor expansion, we estimate the update of l-th cross-attention layer $||y_{l+1}^* - y_{l+1}||$ as:

$$||y^*_{l+1} - y_{l+1}|| = ||f^*_{dl}(y^*_l, x^*_{2N+1}, \theta^*_{dl}) - f_{dl}(y_l, x_{2N+1}, \theta_{dl})||$$

$$\approx \frac{\alpha_d}{\sqrt{\alpha_d^2 + v_{dl}^2 w_{dl}^2}}||y^*_l - y_l|| + \frac{v_{dl}w_{dl}}{\sqrt{\alpha_d^2 + v_{dl}^2 w_{dl}^2}}||x^*_{2N+1} - x_{2N+1}||$$

$$+ \frac{\alpha_d(\alpha_d - v_{dl}w_{dl})}{(\alpha_d^2 + v_{dl}^2 w_{dl}^2)^{\frac{3}{2}}}\sqrt{v_{dl}^2 + w_{dl}^2}||\theta^*_{dl} - \theta_{dl}||$$

$$\leq ||y^*_l - y_l|| + \frac{v_{dl}w_{dl}}{\alpha_d}||x^*_{2N+1} - x_{2N+1}|| + \frac{\sqrt{v_{dl}^2 + w_{dl}^2}}{\alpha_d}||\theta^*_{dl} - \theta_{dl}|| \quad (12)$$

According to Theorem 4.2, we have $||x^*_{2N+1} - x_{2N+1}|| = \mathcal{O}(\sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e}||\theta^*_{ei} - \theta_{ei}||)$. Therefore, the magnitude of $||\Delta F_{ed}||$ satisfies:

$$||\Delta F_{ed}|| \leq \sum_{j=1}^{M} \frac{v_{d,3j-1}w_{d,3j-1}}{\alpha_d} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e}||\theta^*_{ei} - \theta_{ei}|| + \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d}||\theta^*_{dj} - \theta_{dj}|| \quad (13)$$

$\square$

As a special case, the corresponding parameters in Equation (13) for vanilla Post-LN with standard initialization are 1, so its model update $||\Delta F_{ed}|| = \mathcal{O}(M \sum_{i=1}^{2N} ||\theta^*_{ei} - \theta_{ei}|| + \sum_{j=1}^{3M} ||\theta^*_{dj} - \theta_{dj}||)$.

B Derivation for Encoder-Decoder Architecture

Here, we give the derivation of DEEPNET for the encoder-decoder architecture with an N-layer encoder and an M-layer decoder. As in Section 4.3, we have $v_d = w_d = (12M)^{-\frac{1}{4}}$, $\alpha_d = (3M)^{\frac{1}{4}}$ to bound the second term of Equation (13) to $\Theta(\eta)$. For the first term, we set $v_{ei} = v_e, w_{ei} = w_e$, so that it goes to:

$$\sum_{j=1}^{M} \frac{v_{d,3j-1}w_{d,3j-1}}{\alpha_d} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e}||\theta^*_{ei} - \theta_{ei}|| = M \frac{(12M)^{-\frac{1}{2}}}{(3M)^{\frac{1}{4}}} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e}||\theta^*_{ei} - \theta_{ei}||$$

$$(14)$$

$$\overset{\Theta}{=} \eta \left(\frac{N^4 M}{27}\right)^{\frac{1}{4}} \frac{v_e^2 + w_e^2}{\alpha_e^2} \quad (15)$$

In this work, we use $\alpha_e^2 = (N^4 M/27)^{\frac{1}{8}}$, $v_e^2 + w_e^2 = (N^4 M/27)^{-\frac{1}{8}}$ and $v_e = w_e = \beta_e$ that is $\alpha_e = 0.81(N^4 M)^{\frac{1}{16}}$, $\beta_e = 0.87(N^4 M)^{-\frac{1}{16}}$ to satisfy the condition.

C Derivation for Encoder-only (Decoder-only) Architecture

For an N-layer DEEPNET, starting from Theorem 4.2 we have,

$$||x^*_{2N+1} - x_{2N+1}|| \leq \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha}||\theta^*_i - \theta_i||$$

$$\leq \eta \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha}||\frac{\partial \mathcal{L}}{\partial F}|| \cdot ||\frac{\partial F}{\partial \theta_i}|| \quad (16)$$

By assumption $||\frac{\partial \mathcal{L}}{\partial F}|| = \mathcal{O}(1)$, and $||\frac{\partial F}{\partial \theta_i}|| \leq ||\frac{\partial F}{\partial \theta_{2N}}|| \overset{\Theta}{=} \frac{||\theta_{2N}||}{\alpha}$, we achieve:

$$\sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha}||\frac{\partial \mathcal{L}}{\partial F}|| \cdot ||\frac{\partial F}{\partial \theta_i}|| \leq \mathcal{O}(\frac{\sqrt{v_{2N}^2 + w_{2N}^2}}{\alpha} \sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha}) = \mathcal{O}(1) \quad (17)$$

Due to symmetry, we set $v_i = v, w_j = w$, so it goes to $2N \frac{v^2 + w^2}{\alpha^2} = 1$. In this work, we use $v = w = (8N)^{-\frac{1}{4}}$ and $\alpha = (2N)^{\frac{1}{4}}$ to satisfy the condition.

D Experimental Details

D.1 Hyperparameters for IWSLT-14 De-En

Hyperparameters	Value
Learning rate	5e-4
Learning rate scheduler	inverse sqrt
Warm-up updates	4000
Warm-up init learning rate	1e-7
Max tokens	4000
Adam ϵ	1e-8
Adam β	(0.9, 0.98)
Label smoothing	0.1
Training updates	8K
Gradient clipping	0.0
Dropout	0.4
Weight decay	0.0001
Hidden size	512
FFN inner hidden size	2048
Attention heads	8

Table 4: Hyperparameters for the machine translation experiments on the IWSLT-14 De-En dataset.

D.2 Hyperparameters for WMT-17 En-De

Hyperparameters	No-LN	Pre-LN	Post-LN	DEEPNORM
Learning rate	5e-4	1.5e-3	1.5e-3	1.5e-3
Learning rate scheduler			inverse sqrt	
Warm-up updates			4000	
Warm-up init learning rate			1e-7	
Max tokens			128×4096	
Adam ϵ			1e-8	
Adam β			(0.9, 0.98)	
Label smoothing			0.1	
Training updates			100K	
Gradient clipping			0.0	
Dropout			0.4	
Weight decay			0.0001	
Hidden size			512	
FFN inner hidden size			2048	
Attention heads			8	

Table 5: Hyperparameters for the base-setting experiments on the WMT-17 En-De dataset.

Hyperparameters	Base size	Medium size	Large size
Hidden size	512	768	1,024
FFN inner hidden size	2048	3072	4096
Attention heads	8	12	16
Layers		18-18	
Learning rate		5e-4	
Learning rate scheduler		inverse sqrt	
Warm-up updates		4000	
Warm-up init learning rate		1e-7	
Max tokens		128×4096	
Adam ϵ		1e-6	
Adam β		(0.9, 0.98)	
Label smoothing		0.1	
Training updates		30K	
Gradient clipping		1.0	
Dropout		0.4	
Weight decay		0.0	

Table 6: Hyperparameters for the large-setting experiments on the WMT-17 En-De dataset.

D.3 Hyperparameters for OPUS-100

Hyperparameters	Value
Learning rate	5e-4
Learning rate scheduler	inverse sqrt
Warm-up updates	4000
Warm-up init learning rate	1e-7
Max tokens	128×4096
Adam ϵ	1e-8
Adam β	(0.9, 0.98)
Label smoothing	0.1
Training epochs	4
Gradient clipping	0.0
Dropout	0.1
Weight decay	0.0
Hidden size	512
FFN inner hidden size	2048
Attention heads	8

Table 7: Hyperparameters for the machine translation experiments on the OPUS-100 dataset.

D.4 Hyperparameters for 102-Language Machine Translation

Hyperparameters	Value
Learning rate	5e-4
Learning rate scheduler	inverse sqrt
Warm-up updates	6000
Warm-up init learning rate	1e-7
Max tokens	256×4096
Adam ϵ	1e-6
Adam β	(0.9, 0.98)
Label smoothing	0.1
Training updates	260K
Gradient clipping	1.0
Dropout	0.1
Weight decay	0.0
Hidden size	1024
FFN inner hidden size	4096
Attention heads	16
Layers	100-100

Table 8: Hyperparameters for the machine translation experiments on the 102-language dataset.

D.5 Evaluation Details

For IWSLT-14 and WMT-17, we use the in-built BLEU scripts of Fairseq to report the scores. Besides, we report the case-sensitive detokenized BLEU using sacreBLEU (Post, 2018) for the results of OPUS-100.[5]

For WMT, OPUS, and TED, we use the same test sets and evaluation scripts as in M2M (Fan et al., 2021), and the results of M2M are directly from the paper (Fan et al., 2021). For the Flores-101 evaluation set, we report the spBLEU[6] of M2M-12B with the public checkpoint and script.[7]

---

[5]BLEU+case.mixed+lang.{src}-{tgt}+numrefs.1+smooth.exp+tok.13a+version.1.4.14
[6]https://github.com/facebookresearch/flores
[7]https://github.com/pytorch/fairseq/tree/main/examples/m2m_100

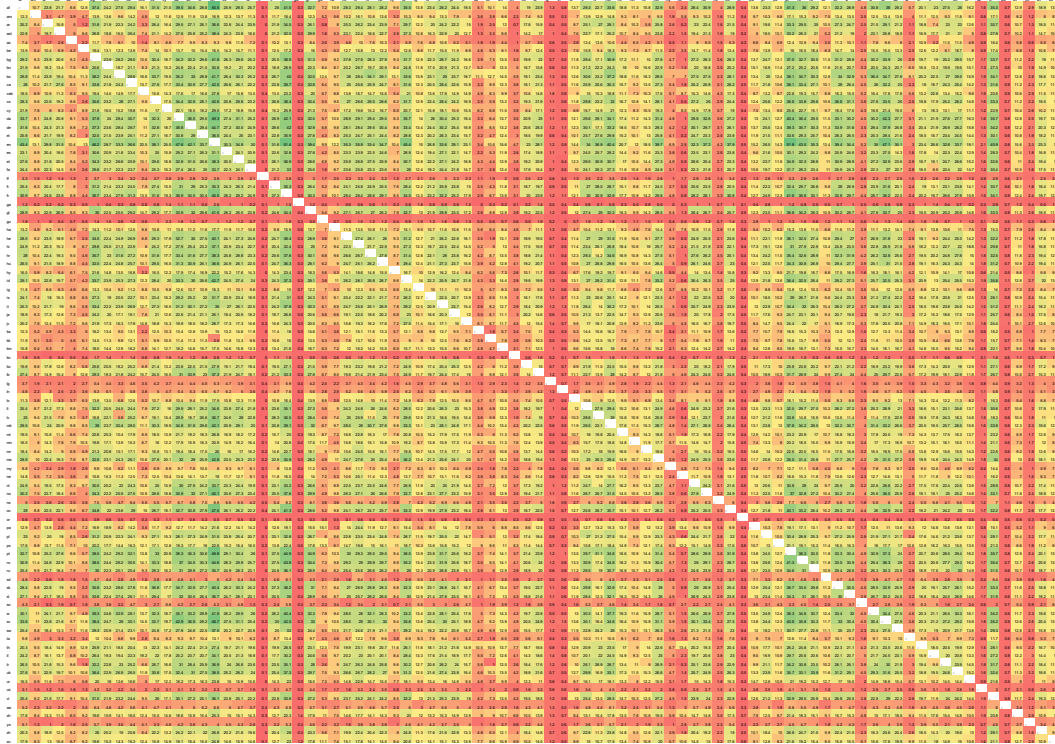# E Experimental Results in Section 6

Figure 9: Evaluation results of 12B M2M-100 on a subset of FLORES-101 devtest set. The i-th row is the source language, while j-th column is the target language. There are 87 languages and 7,482 directions.
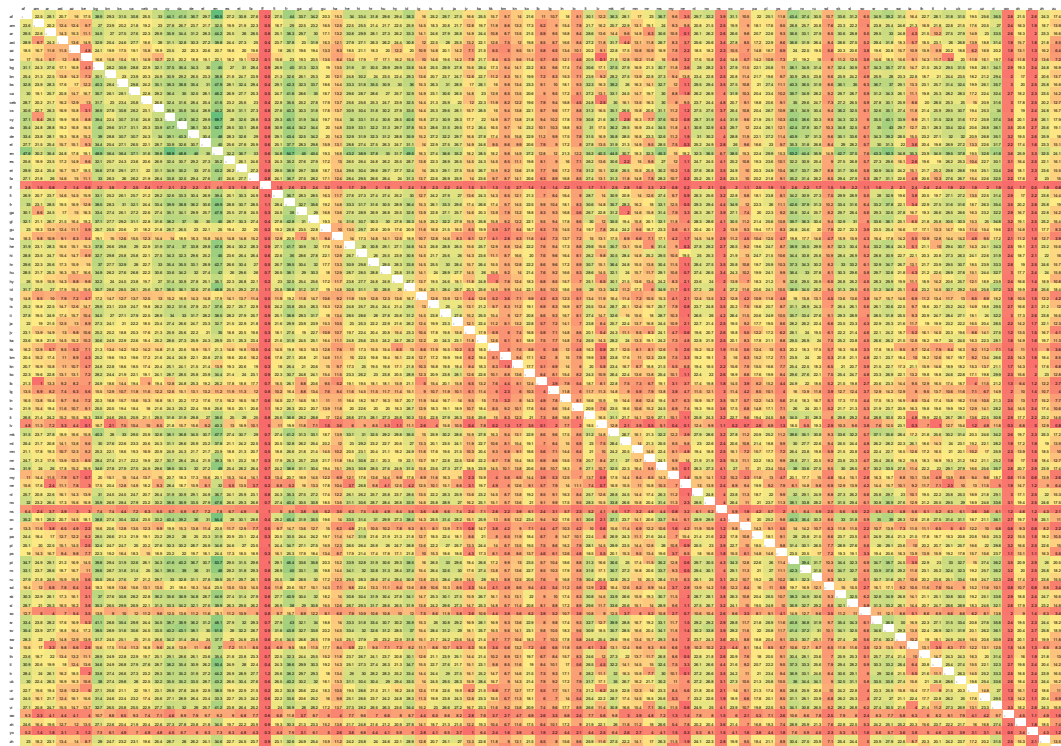
Figure 10: Evaluation results of 3.2B DEEPNET on a subset of FLORES-101 devtest set. The i-th row is the source language, while j-th column is the target language. There are 87 languages and 7,482 directions.