DeepNet: Scaling Transformers to 1,000 Layers

Hongyu Wang*, Shuming Ma*, Li Dong, Shaohan Huang, Dongdong Zhang and Furu Wei

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. Extensive experiments demonstrate that DEEPNET has superior performance across various benchmarks, including machine translation, language modeling (i.e., BERT, GPT) and vision pre-training (i.e., BEiT). 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. Our code is available at https://aka.ms/torchscale.

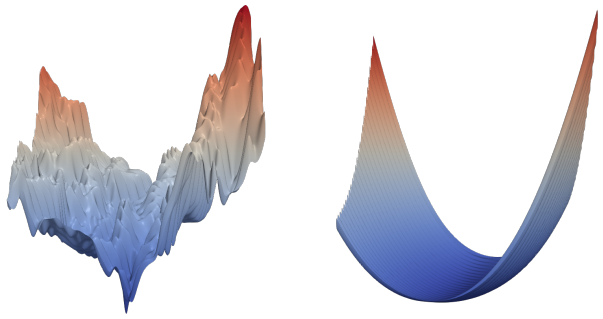
Index Terms—Transformers, Training Stability, Loss Landscape, Big Models, Optimization

1 INTRODUCTION

Recent years have witnessed a trend towards large-scale Transformer [1] models. The capacity has substantially increased from millions of parameters [2], [3] to billions [4], [5], [6], [7], [8], [9], [10], [11], and even trillions [12], [13]. 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 are limited by the training instability of Transformers.

Nguyen *et al*. [14] found 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 [15], 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 [16], [17], [18], or better architecture [15], [19], [20], [21]. 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 [22], which has theoretical justification of bounding the model update by a constant. We adopt the filter normalization [23] to visualize the loss surface of vanilla Post-LN and DEEPNET on the IWSLT-14 De-En data set at the early stage of training. Figure 1 shows that the loss surface of DEEPNET is much smoother compared with vanilla Post-LN. 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.



(a) Post-LN Transformers (b) DEEPNET

Fig. 1. The loss surface of 36-layer vanilla Post-LN and DEEPNET at the early stage of training.

Extensive experiments demonstrate that DEEPNET has superior performance across various benchmarks, including machine translation, language modeling (i.e., BERT, GPT) and vision pre-training (i.e., BEiT). 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 [24] with 48 layers and 12B model size.

- *Equal contribution. Work was done during Hongyu's internship at Microsoft Research.*
- *Hongyu Wang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, 100049. E-mail: wanghongyu22@mails.ucas.ac.cn*
- *Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang and Furu Wei are with Microsoft Research. E-mail: {shumma, lidong1, shaohanh, dozhang, fuwei}@microsoft.com*

Manuscript received February 20, 2023; revised mmm dd, 2023.

This paper includes the analysis for Pre-LN variants and the experiments for language modeling and vision pre-training of our ICML 2023 paper [25] which is an extension and application of our proposed framework for training stability of deep Transformers in this work.

2 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 [16], [18], [26]. Therefore, we study the 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 [16], 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 data set. Figure 2 visualizes their gradients and validation loss curves. As shown in Figure 2(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 2 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 3(a), we first visualize the norm of model update $||\Delta F||_2$ at the early stage of training:

$$||\Delta F||_2 = ||F(x, \theta_i) - F(x, \theta_0)||_2, \tag{1}$$

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 3(b) and Figure 3(c)). According to the theoretical analysis from Xiong *et al.* [27], the magnitude of gradient through LN is inversely proportional to the magnitude of its input:

$$||\frac{\partial LN(x)}{\partial x}||_2 = \mathcal{O}(\frac{\sqrt{d}}{||x||_2}).$$

Figure 3(b) and Figure 3(c) show that $||x||_2$ 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 3(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.

3 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.

3.1 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 3.3.

3.2 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:

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

We study the magnitude of the attention module. Lemma 1 proves that W^Q and W^K do not change the bound of attention output's magnitude.

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

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

where $\overset{\Theta}{=}$ *stands for equal upper bound of expected magnitude.*

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3386927

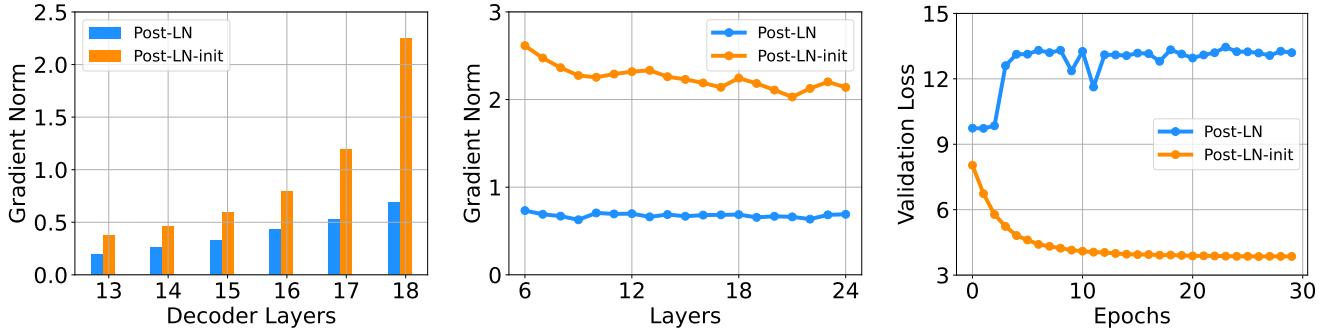
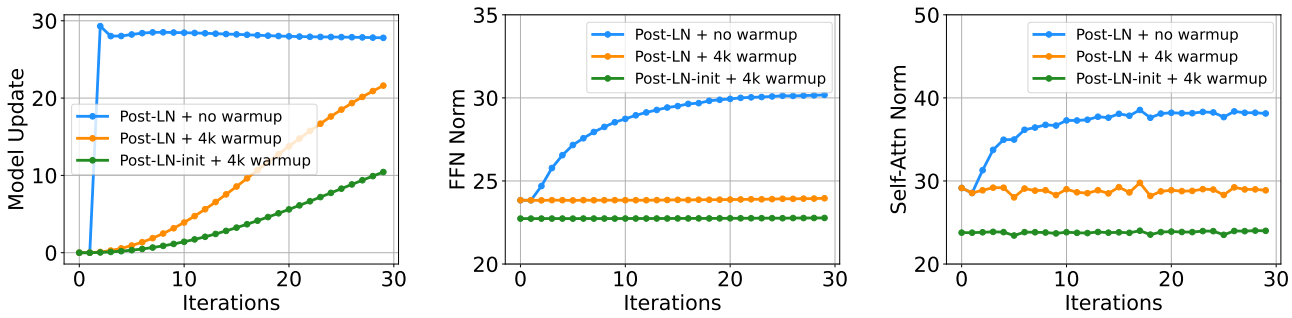
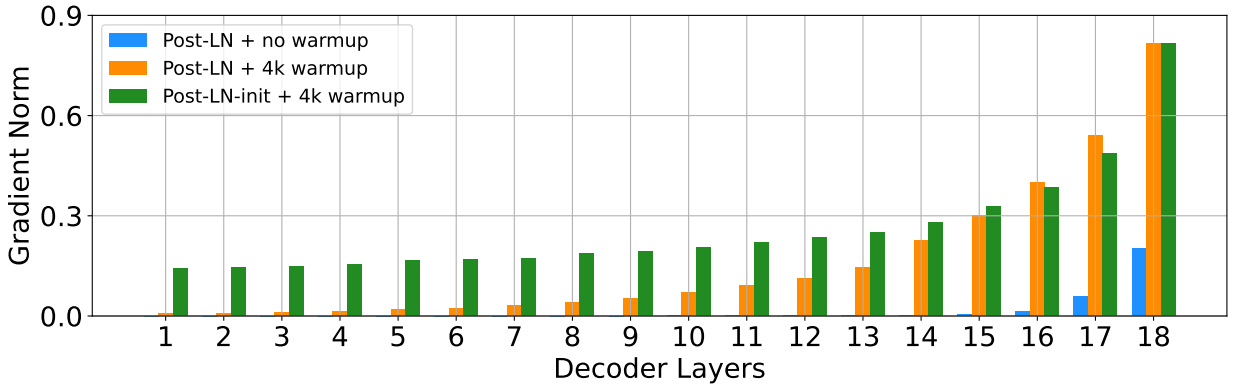


Fig. 2. (a) Gradient norm in the top layers of 18L-18L models. (b) Gradient norm in the last layer of the models with depths varying from 6L-6L to 24L-24L. (c) Validation loss curves of 18L-18L models.



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



(d) Gradient norm in all decoder layers

Fig. 3. Visualization of the model update, the average input of LNs, and the gradients for the 18L-18L models at the early stage of training.

In other words, the magnitude of attention output only depends on the value and output projection: $Attn(Q,K,V) \overset{\Theta}{=} VW^VW^O$. 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 $Attn(Q,K,V) \overset{\Theta}{=} vwV$. Similarly, we have $FFN(X) \overset{\Theta}{=} vwX$, where v, w denotes the parameters of the feed-forward network.

We define the model update as $||\Delta F||_2 = ||F(x,\theta^*) - F(x,\theta)||_2$. Based on the analysis above, we have the following theorem to characterize $||\Delta F||_2$'s magnitude of an N-layer DEEPNET with N attentions and FFNs.

Theorem 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*: $x_{l+1} = LN(\alpha x_l + G_l(x_l, \theta_l))$, the expected model update $||\Delta F||_2$ satisfies:*

$$||\Delta F||_2 = \mathcal{O}(\sum_{i=1}^{2N} \frac{\sqrt{v_i^2 + w_i^2}}{\alpha} ||\theta_i^* - \theta_i||_2)$$

Vanilla Post-LN can be regarded as a special case of DEEPNET, where $\alpha = 1$ and $v_l = w_l = 1$ at Xavier initialization [28]. Based on Theorem 2, we have $||\Delta F||_2 = \mathcal{O}(\sum_{i=1}^{2N} ||\theta_i^* - \theta_i||_2)$ for vanilla Post-LN. It shows that the model tends to accumulate the update of each sub-layer,

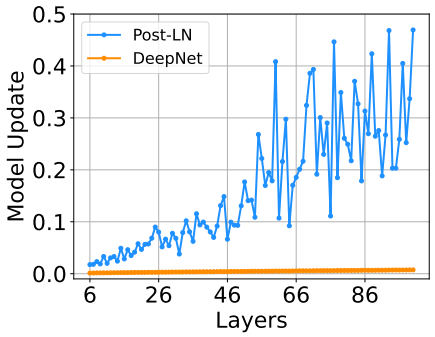


Fig. 4. Model updates of vanilla Post-LN and DEEPNET at the early stage of training. The visualization is conducted on 64-128-2 tiny Transformers with depth varying from 6L-6L to 100L-100L. It shows that DEEPNET has much smaller and more stable updates than Post-LN.

which leads to exploding magnitude of model's update and destabilizes the optimization at the early stage. This explains our findings in Section 2.

Besides, Theorem 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 $||\theta_i^* - \theta_i||_2$, 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 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 $||\Delta F_{ed}||_2 = ||F_{ed}(x, y, \theta_e^*, \theta_d^*) - F_{ed}(x, y, \theta_e, \theta_d)||_2$.

Theorem 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* $x_{l+1} = LN(\alpha_d x_l + G_{dl}(x_l, \theta_{dl}))$*, the expected model update* $||\Delta F_{ed}||_2$ *satisfies:*

$$
\begin{aligned}
&||\Delta F_{ed}||_2 \\
&= \mathcal{O}(\sum_{j=1}^{M} \frac{v_{d,3j-1} w_{d,3j-1}}{\alpha_d} \sum_{i=1}^{2N} \frac{\sqrt{v_{ei}^2 + w_{ei}^2}}{\alpha_e} ||\theta_{ei}^* - \theta_{ei}||_2 \\
&\quad + \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d} ||\theta_{dj}^* - \theta_{dj}||_2) \qquad (2)
\end{aligned}
$$

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 $||\Delta F_{ed}|| = \mathcal{O}(M \sum_{i=1}^{2N} ||\theta_{ei}^* - \theta_{ei}||_2 + \sum_{j=1}^{3M} ||\theta_{dj}^* - \theta_{dj}||_2)$. It indicates the similar accumulative effect which leads to fast growth of the magnitude regarding the model depth (see Figure 4). Furthermore, the cross-attention propagates the magnitude from the encoder to the decoder, which explains why the decoder is more unstable than the encoder [20].

3.3 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 [29]. 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.* [17], 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 $||\Delta F_{ed}||_2 = \Theta(\eta)$ where $\Delta F_{ed} \triangleq F_{ed}(x, y, \theta_e - \eta \frac{\partial \mathcal{L}}{\partial \theta_e}, \theta_d - \eta \frac{\partial \mathcal{L}}{\partial \theta_d}) - F_{ed}(x, y, \theta_e, \theta_d)$.

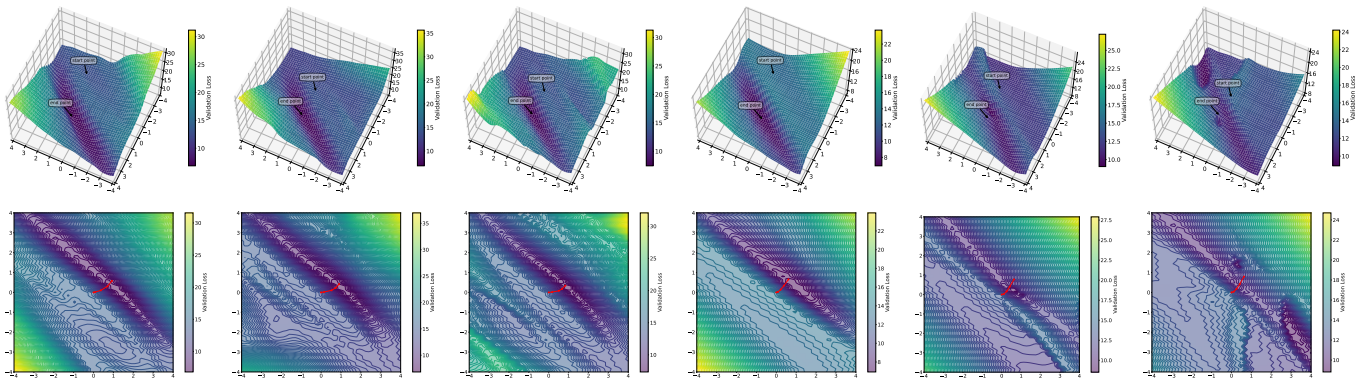
For SGD optimizer, the update of each decoder layer $||\theta_{di}^* - \theta_{di}||_2$ equals to $\eta ||\frac{\partial \mathcal{L}}{\partial \theta_{di}}||_2$. Xiong *et al.* [27] proved that Post-LN decreases the magnitude of backpropagating error signal, so we have $||\frac{\partial F}{\partial \theta_{dj}}||_2 \le ||\frac{\partial F}{\partial \theta_{d,3M}}||_2$. With $||\frac{\partial F}{\partial \theta_{d,3M}}||_2 \overset{\Theta}{=} \frac{||\theta_{d,3M}||_2}{\alpha_d}$ and the assumption $||\frac{\partial \mathcal{L}}{\partial F}||_2 = \mathcal{O}(1)$, the second term of Equation 2 can be bounded as:

$$
\begin{aligned}
&\sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d} ||\theta_{dj}^* - \theta_{dj}||_2 \qquad (3) \\
&\le \eta ||\frac{\partial \mathcal{L}}{\partial F}||_2 \cdot ||\frac{\partial \mathcal{F}}{\partial \theta_{d,3M}}||_2 \sum_{j=1}^{3M} \frac{\sqrt{v_{dj}^2 + w_{dj}^2}}{\alpha_d} \\
&\overset{\Theta}{=} 3\eta M \frac{v_d^2 + w_d^2}{\alpha_d^2} \qquad (4)
\end{aligned}
$$

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

In comparison with Post-LN, we visualize the model updates for DEEPNET on IWSLT-14 De-En translation data set at the early training stage. Figure 4 shows that the model update of DEEPNET is nearly constant, while the model update of Post-LN is exploding. Following Hao *et al.* [30], we further visualize the loss landscape and trajectory of the optimization of vanilla Post-LN and DEEPNET with varying model depth. Figure 5 presents that the loss landscape of vanilla Post-LN is less smooth with the increasing model depth, while our model tends to have consistent smoothness across different depth. Besides, vanilla Post-LN is easier to be stuck in a spurious local optima, which verifies our analysis in Section 2.

In summary, we apply our approach as follows:



(a) 10L-10L (b) 50L-50L (c) 100L-100L (d) 10L-10L (e) 50L-50L (f) 100L-100L

Fig. 5. Loss landscape and trajectory of DEEPNET (a, b, c) and vanilla Post-LN (d, e, f) at the early stage of training. The visualization is conducted on 64-128-2 tiny Transformers with varying depth.

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 architectures can be conducted in the same way (see Appendix B). 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 $(2M)^{\frac{1}{4}}$).

4 DEEPNET FOR PRE-LN TRANSFORMER

In this section, we further extend our analysis framework to Pre-LN variants, which are widely adopted as the architecture for vision transformers [31] and large language models [5], [32]. We first introduce the architecture of DEEPNET for Pre-LN. Then we estimate the expected magnitude of model update. Moreover, we show that the model update of Pre-LN-style DEEPNET grows logarithmically as the depth increases, which can also be bounded independent of depth with our proposed initialization.

4.1 Architecture

For DEEPNET, we introduce the extra normalization inside each sublayer to ease the explosion of activation during training. Specially, for the multihead attentions, the layer normalization modules are before the qkv projection and the output projection, which can be formulated as:

$$Q, K, V = W^Q\text{LN}(x), W^K\text{LN}(x), W^V\text{LN}(x) \quad (5)$$

$$\text{MSA}(x) = x + W^O\text{LN}(\text{Attention}(Q, K, V)) \quad (6)$$

where W^Q, W^K, W^V, and W^O are the parameters of the multihead self-attention. For the feed-forward network, we place the normalizations before the input projection and the output projection, which are written as:

$$\text{FC}_1(x) = W^1\text{LN}(x) \quad (7)$$

$$\text{FC}_2(x) = W^2\text{LN}(x) \quad (8)$$

$$\text{FFN}(x) = \text{FC}_2(\phi(\text{FC}_1(x))) \quad (9)$$

where W^1 and W^2 are parameters of the feed-forward layers, and ϕ is the non-linear activation function.

4.2 Expected Magnitude of Model Update

Based on the framework before, we study the magnitude of DEEPNET for Pre-LN with an N-layer encoder under SGD update. With Lemma 1, the query and key projection do not change the bound of expected magnitude of attention update. Similarly, we denote the parameters of the encoder θ_e as $\{v_l, w_l\}_{l=1}^L$, where w_l and v_l denote the scale of input and output projection of FFN, or value and output projection of attention module. We set the scale of shortcut α as 1 to prevent the exponential accumulation of model update along residual shortcuts. Above all, we have the following theorem to characterize the expected model update of an N-layer Pre-LN-style DEEPNET. The proof is detailed in the Appendix A.4.

Theorem 4. *Given an N-layer* DEEPNET $F(x, \theta)$*, the l-th sublayer is formulated as* $x^l = x^{l-1} + W^{l,2}LN(\phi(W^{l,1}LN(x^{l-1})))$*. Under SGD update,* ΔF *satisfies:*

$$\Delta F = \mathcal{O}(\eta(\frac{\sum_{l=1}^{L}(1+\frac{v_l^2}{w_l^2})}{\sum_{n=1}^{L} v_n^2} + \sum_{l=1}^{L}\sum_{k=2}^{L}\frac{1+\frac{v_l^2}{w_l^2}}{\sum_{n=1}^{L} v_n^2}\frac{v_k^2}{\sum_{n=1}^{k-1} v_n^2})) \tag{10}$$

where η *is learning rate,* L *equals to* $2N$*.*

If we apply standard initialization (e.g., Xavier initialization) for each sublayer, the output can preserve the variance of input. Therefore, v_l and w_l can be estimated as 1 at the beginning of training. With Theorem 4, we have $||\Delta F|| = \mathcal{O}(\log L)$. It shows that the expected magnitude of model update for DEEPNET grows logarithmically as the depth increases, which is much smaller than that of vanilla Post-LN. It indicates that DEEPNET is easier to be optimized and can be scaled up to extremely deep models.

4.3 Derivation

Furthermore, we demonstrate that the expected model update of DEEPNET can be further bounded with proper initialization. The detailed derivation can be found in Appendix A.4. We adopt $v = w = \beta = \sqrt{\log L}$ to bound the model update independent of the depth. In summary, we apply our initialization as follows:

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 $\sqrt{\log 2N}$ (or $\sqrt{\log 2M}$).

5 NEURAL MACHINE TRANSLATION

We verify the effectiveness of DEEPNET on the popular machine translation benchmarks, including IWSLT-14 German-English (De-En), WMT-17 English-German (En-De), WMT-14 English-German (En-De) and WMT-14 English-French (En-Fr) data set. We compare our method with multiple state-of-the-art deep Transformer models, including DLCL [19], NormFormer [15], ReZero [21], R-Fixup [17], T-Fixup [18], DS-init [16], and Admin [20]. 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. Besides, we adopt the in-built BLEU scripts of Fairseq to evaluate all models. Table 1 and Table 2 reports the results of the baselines and DEEPNET on WMT-17 En-De, WMT-14 En-De and WMT-14 En-Fr translation data set, respectively. 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 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.* [20]. 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 [15]. 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[1]. Figure 6 shows the results on the IWSLT-14 data set. 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.

We conduct experiments on the large-scale multilingual machine translation, which is a good testbed for large models. We first use OPUS-100 corpus [33] 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 3 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

1. According to our experiments, ReZero is unstable with half precision, even when the model is shallow.

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.

Models	LN	6L-6L	18L-18L	50L-50L	100L-100L
Vanilla Post-LN [1]	Post-LN	**28.1**		diverged	
DS-Init [16]	Post-LN	27.9		diverged	
Admin [20]	Post-LN	27.9	**28.8**	diverged	
ReZero [21]	No-LN	26.9		diverged	
R-Fixup [17]	No-LN	27.5	28.4	27.7	diverged
T-Fixup [18]	No-LN	27.5	28.4	27.9	diverged
Vanilla Pre-LN [1]	Pre-LN	27.0	28.1	28.0	27.4
DLCL [19]	Pre-LN	27.4	28.2	diverged	27.5
NormFormer [15]	Pre-LN	27.0	28.3	27.8	diverged
DEEPNET (ours)	DEEPNORM	27.8	**28.8**	**29.0**	**28.9**

TABLE 2
BLEU scores on the WMT14 En-De and WMT14 En-Fr test set for different models with varying depth. AL-BL refers to A-layer encoder and B-layer decoder.

Models	LN	WMT14 En-De		WMT14 En-Fr	
		6L-6L	18L-18L	6L-6L	18L-18L
T-Fixup [18]	No-LN	26.4	28.0	39.8	41.9
Vanilla Post-LN [1]	Post-LN	**27.4**	diverged	39.6	diverged
Admin [20]	Post-LN	**27.4**	28.4	39.4	**42.4**
Vanilla Pre-LN [1]	Pre-LN	26.6	27.8	39.6	41.8
NormFormer [15]	Pre-LN	27.2	27.9	39.7	42.1
DEEPNET (ours)	DEEPNORM	27.3	**28.7**	**39.9**	**42.4**

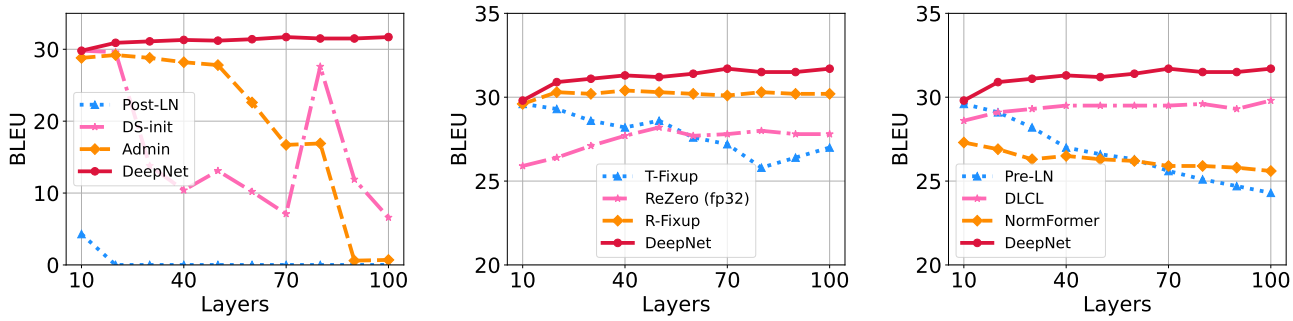


Fig. 6. BLEU scores on the IWSLT-14 De-En test set for different deep models with varying depth from 10L-10L to 100L-100L.

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.

6 MASSIVELY MULTILINGUAL NEURAL MACHINE TRANSLATION

Scaling law in terms of depth. We train DEEPNET of {12, 20, 100, 200, 1000} layers on the OPUS-100 data set. For the evaluation, we report the case-sensitive detokenized BLEU using sacreBLEU [34] for the results of OPUS-100.[2]. 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.

Comparsion given similar training FLOPs. Following [35], [36], the training FLOPs can be estimated as $6ND$, where N and D denote the parameters of the model and

2. BLEU+case.mixed+lang.{src}-{tgt}+numrefs.1+smooth.exp+tok.13a +version.1.4.14

TABLE 3
Average BLEU for DEEPNET and the baseline on the OPUS-100 test sets.

Models	# Layers	# Params	X→En	En→X	Avg
Baseline [33]	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 4
Comparison for DEEPNET and the baseline given the similar training FLOPs on the OPUS-100 test sets. All models are trained with the same batch size and 50K steps. AL-BL refers to A-layer encoder and B-layer decoder.

Models	# Layers	# Params	X→En	En→X	Avg
Post-LN [1]	12L-12L	610M	33.1	28.9	31.0
DEEPNET (ours)	90L-6L	480M	33.6	28.6	31.1
	48L-48L		**33.7**	**29.5**	**31.6**

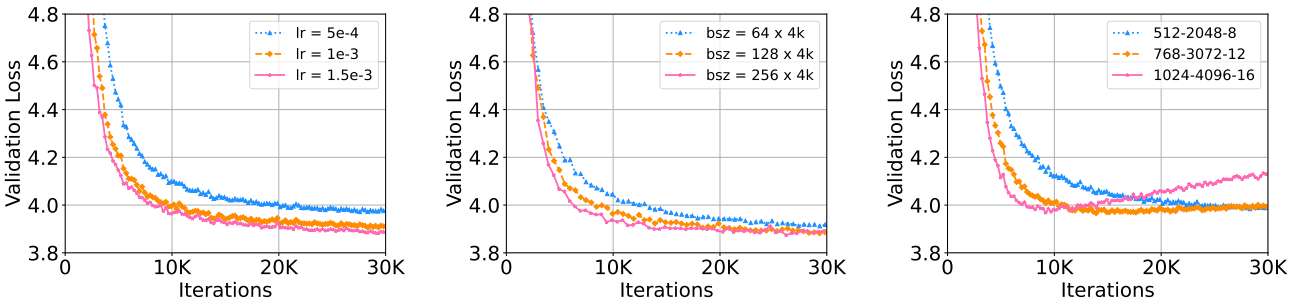


Fig. 7. WMT-17 En-De validation loss curves for 18L-18L DEEPNET with varying learning rate, batch size and hidden dimension.

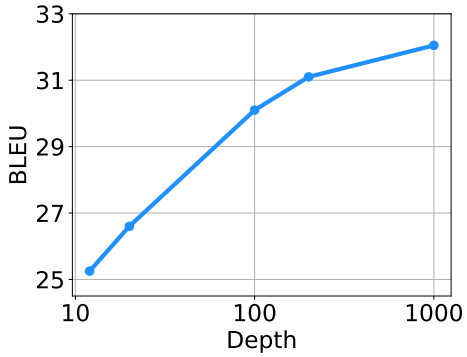


Fig. 8. Average BLEU scores for DEEPNET with varying depth on the OPUS-100 En-X and X-En test sets.

the size of training data, respectively. Therefore, we train DEEPNET of a 48-layer encoder layers, a 48-layer decoder and 512 hidden dimension on the OPUS-100 data set, while the baseline [1] has a 12-layer encoder, a 12-layer decoder and 1024 hidden dimension. Given the similar training FLOPs, all models are trained with 50K steps and the same batch size. The other hyperparameters are detailed in Appendix.

Table 4 shows that the deep and narrow DEEPNET outperforms the shallow and wide baseline by a gain of 0.6 BLEU on the test set of OPUS-100 data set, indicating that deepening the model is a more promising direction given the similar training FLOPs.

Comparison with the asymmetric encoder-decoder. We present the comparison of the asymmetric and symmetric encoder-decoder architecture in Table 4. We train DEEPNET with a 90-layer encoder and a 6-layer decoder on the OPUS-100 data set. As shown in Table 4, the symmetric architecture (48L-48L) outperforms the asymmetric architecture (90L-6L) by a gain of 0.5 BLEU on the test set. It shows that a shallow decoder leads to the degradation of performance on multilingual machine translation, especially for En → X translation directions.

More data and language directions. To explore the limits of DEEPNET on multilingual NMT, we then scale up the training data by using CCMatrix [37]. We also expand the data from CCAligned [38], OPUS [33], and Tatoeba[3] 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

3. https://tatoeba.org/en/

TABLE 5
BLEU scores for DEEPNET and M2M-100 on various evaluation sets.

Models	# Layers	# Params	WMT	OPUS	TED	Flores
M2M-100 [24]	48	12B	31.9	18.4	18.7	13.6
DEEPNET (ours)	200	3.2B	**33.9**	**23.0**	**20.1**	**18.6**

TABLE 6
Results on the GLUE development set.

Models	LR	MNLI	QNLI	QQP	SST	CoLA	MRPC	STS	Avg.
Transformer [1]	5e-4	**86.7/86.7**	92.2	**91.0**	93.4	59.8	86.4	89.4	85.7
	1e-3	diverged							
DEEPNET (ours)	1e-3	86.6/86.6	**92.8**	90.6	**93.7**	**60.4**	**90.2**	**90.1**	**86.4**

layers. Following [35], [36], the training FLOPs can be estimated as 5.2 ZFLOPs, resulting in up to 18 days on 128 TESLA V100-32GB GPUs. More details can be found in the Appendix D.4.

We compare DEEPNET with the state-of-the-art multilingual NMT model M2M-100 [24]. 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 [24], we evaluate the models on several multilingual translation evaluation data sets, including WMT [39], [40], [41], [42], OPUS [33], TED [43], and Flores [44]. The language pairs from the WMT data set are English-centric. There are 10 languages including English, and most of them are high-resource. For the OPUS data set, 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 data set 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 5. For a fair comparison, we use the same evaluation methods as the baseline. For WMT, OPUS, and TED, we adopt the same test sets and evaluation scripts as in M2M-100 [24], and the results of M2M-100 are directly from the paper [24]. For the Flores-101 evaluation set, we report the spBLEU[4] of M2M-12B with the public checkpoint and script.[5] Table 5 shows that DEEPNET has significantly better performance than M2M-100 on all evaluation data sets, indicating that deepening the model is a very promising direction to improve the quality of NMT models.

7 MASKED LANGUAGE MODELING

We compare DEEPNET with Transformer [1] on masked language modeling [2], [45]. For a fair comparison, we pre-train DEEPNET and the baselines on the English Wikipedia and the Bookcorpus [45] with 12 layers, 768 hidden dimensions, and 3072 FFN dimensions. More details regarding hyperparameters can be found in the Appendix.

We search the pre-training learning rate among {5e-4, 1e-3}, and choose the largest one that can converge. We fine-tune the models on the GLUE [46] benchmarks. Table 6 demonstrates the results of DEEPNET and the baselines. It shows that our model has better performance and training stability than the strong baselines with a gain of average 0.7 points.

8 CAUSAL LANGUAGE MODELING

We implement DEEPNET on causal language modeling, which is the pre-training task for recent large language models (e.g., GPT [5], [47], LLaMA [32], [48], etc). We start with a model that has the same configuration as GPT-3 Medium (350M), and further scale its depth from 24L to 48L and 72L. All models are trained on an English-language corpus, which is a subset of the data from [45] and the English portion of CC100 corpus. We adopt the GPT-2 tokenizer [4] to preprocess the data. The other training hyperparameters are detailed in the Table 22 of Appendix.

We compare DEEPNET with GPT-2 [4] and Normformer [15]. Normformer is a state-of-the-art architecture for causal language modeling. For a fair comparison, we reproduce the results of our model and the baselines under the same setting. We evaluate their performance of in-context learning. Following the previous work [5], we choose Winogrande [49], Winograd [50], Storycloze [51], and Hellaswag [52] as the benchmark.

Table 7 summarizes the results in the zero-shot setting. It shows that DEEPNET achieves significant improvements over both GPT-2 and Normformer across different scales. Besides, DEEPNET tolerates a larger learning rate than the baselines, indicating that our model is more stable in optimization. This allows DEEPNET to further scale up without pain. Table 8 and Table 9 report the results in the few-shot setting. DEEPNET is also better at few-shot learning than the baselines across four data sets, proving the effectiveness of DEEPNET on causal language modeling.

4. https://github.com/facebookresearch/flores
5. https://github.com/pytorch/fairseq/tree/main/examples/m2m_100

TABLE 7
Zero-shot results for DEEPNET and the baselines (WGe: Winogrande, WG: Winograd, SC: Storycloze, and HS: Hellaswag data set).

Models	# Layers	LR	WGe	WG	SC	HS	Avg.
GPT-2 [4]	24L	5e-4	**55.2**	65.3	70.8	44.8	59.0
GPT-2 [4]		1e-3			diverged		
Normformer [15]		5e-4	54.3	68.1	72.0	45.9	60.1
Normformer [15]		1e-3			diverged		
DEEPNET (ours)		1e-3	54.3	**71.9**	**72.4**	**46.9**	**61.4**
GPT-2 [4]	48L	5e-4	**57.3**	67.0	74.0	48.0	61.6
Normformer [15]		5e-4	56.5	70.5	74.0	49.8	62.7
DEEPNET (ours)		1.2e-3	57.0	**73.3**	**74.7**	**51.2**	**64.1**
GPT-2 [4]	72L	5e-4	**58.0**	70.9	75.7	51.7	64.1
Normformer [15]		5e-4	57.4	**75.4**	75.2	53.6	65.4
DEEPNET (ours)		1.2e-3	57.9	73.7	**76.6**	**55.1**	**65.8**

TABLE 8
One-shot results for DEEPNET and the baselines (WGe: Winogrande, WG: Winograd, SC: Storycloze, and HS: Hellaswag data set).

Models	# Layers	LR	WGe	WG	SC	HS	Avg.
GPT-2 [4]	24L	5e-4	**54.4**	66.7	71.0	44.8	59.2
GPT-2 [4]		1e-3			diverged		
Normformer [15]		5e-4	54.0	67.4	72.1	45.6	59.8
Normformer [15]		1e-3			diverged		
DEEPNET (ours)		1e-3	54.1	**70.2**	**72.8**	**47.3**	**61.1**
GPT-2 [4]	48L	5e-4	56.0	69.5	74.2	48.5	62.1
Normformer [15]		5e-4	54.7	71.2	74.8	50.6	62.8
DEEPNET (ours)		1.2e-3	**56.8**	**71.6**	**74.9**	**51.5**	**63.7**
GPT-2 [4]	72L	5e-4	56.9	71.2	76.0	52.2	64.1
Normformer [15]		5e-4	57.8	69.8	76.8	54.0	64.6
DEEPNET (ours)		1.2e-3	**59.8**	**74.0**	**77.9**	**55.5**	**66.8**

TABLE 9
Four-shot results for DEEPNET and the baselines (WGe: Winogrande, WG: Winograd, SC: Storycloze, and HS: Hellaswag data set).

Models	# Layers	LR	WGe	WG	SC	HS	Avg.
GPT-2 [4]	24L	5e-4	54.0	67.7	69.8	44.6	59.0
GPT-2 [4]		1e-3			diverged		
Normformer [15]		5e-4	54.3	70.2	71.4	45.9	60.5
Normformer [15]		1e-3			diverged		
DEEPNET (ours)		1e-3	**57.6**	**74.7**	**72.8**	**47.5**	**63.2**
GPT-2 [4]	48L	5e-4	57.7	71.2	73.8	48.7	62.9
Normformer [15]		5e-4	56.8	**75.4**	75.9	50.7	**64.7**
DEEPNET (ours)		1.2e-3	**57.9**	71.9	**76.4**	**51.9**	64.5
GPT-2 [4]	72L	5e-4	57.5	73.3	76.1	52.4	64.8
Normformer [15]		5e-4	57.7	**74.0**	77.0	54.9	65.9
DEEPNET (ours)		1.2e-3	**58.3**	**74.0**	**79.0**	**55.7**	**66.8**

9 MASKED IMAGE MODELING

We pretrain DEEPNET under masked image modeling framework (BEiT; [53], [54]), and then fine-tune the model on various downstream vision tasks by appending lightweight task layers. Specifically, we encourage DEEPNET to reconstruct corresponding discrete visual tokens [54], based on the corrupt input images.

We compare DEEPNET with the vanilla ViT [31]. All models are pretrained on the ImageNet-1k [55] with 300 epochs schedule under the same settings for a fair comparison. After that, we fine-tune them on ImageNet-1k for the image classification and on ADE20k [56] for the semantic segmentation. Further, we evaluate the robustness of all fine-tuned models on various ImageNet variants, namely ImageNet-Adversarial [57], ImageNet-Rendition [58] and ImageNet-Sketch [59]. We summarize the results of those vision tasks in Table 10. The hyperparameters are detailed in Appendix.

Table 10 shows that DEEPNET surpasses the vanilla ViT by 0.4% and 0.6% for ViT-Base and ViT-Large on the validation set of ImageNet, respectively. Moreover, DEEPNET outperforms the baseline by a significant margin across three

TABLE 10
Results on vision tasks. We report top-1 accuracy on ImageNet and its variants, and mIoU metric on ADE20k for semantic segmentation. We compare both ViT-Base (12L) and ViT-Large (24L).

Models	# Layers	ADE20k	ImageNet	ImageNet Adversarial	ImageNet Rendition	ImageNet Sketch	Avg.
ViT [31]	12L	51.4	84.5	45.9	55.6	42.2	57.1
DEEPNET (ours)		**52.2**	**84.9**	**48.9**	**57.7**	**43.9**	**58.9**
ViT [31]	24L	54.2	86.2	60.1	63.2	48.5	64.5
DEEPNET (ours)		**54.6**	**86.8**	**65.4**	**67.5**	**52.0**	**67.9**

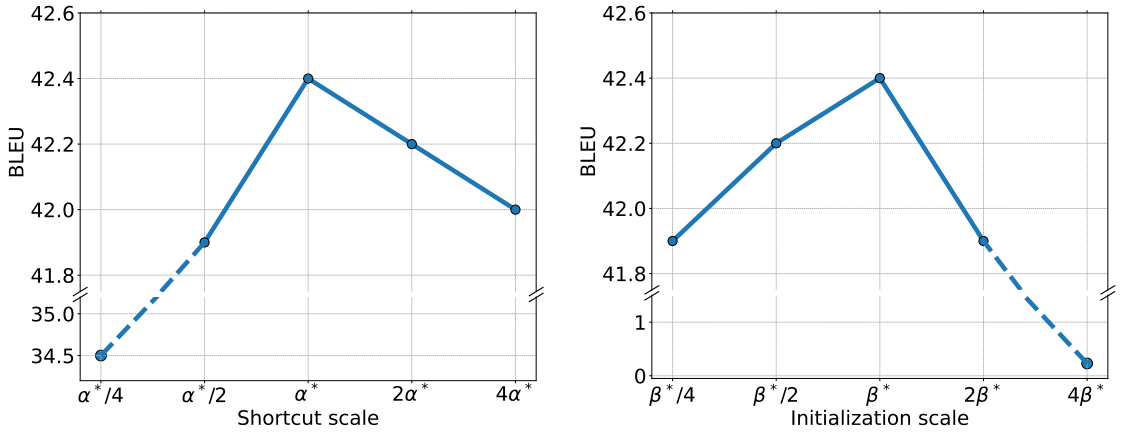


Fig. 9. BLEU scores of DEEPNET on the test set of WMT-14 En-Fr data set for different scales of shortcut (Left) and initialization (Right). Note that α^* and β^* denote the parameters of DEEPNORM. All models are trained with an 18-layer encoder and 18-layer decoder.

TABLE 11
Ablations for DeepNorm and its initialization on WMT-14 En-Fr test set.

Models	# Layers	BLEU
DeepNet	36L	**42.4**
- DeepNorm		diverged
- Initializaton		42.2

ImageNet variants. By appending the UperNet [60] task layer, we conduct semantic segmentation experiments on ADE20k. For ViT-Base models, DEEPNET achieves a gain of 0.8% mIoU compared with the vanilla ViT. For ViT-Large models, DEEPNET can boost the performance to 54.6%.

10 ABLATION STUDY

In this section, we present the ablation study of DEEPNET on the WMT-14 English-French (En-Fr) data set. All models are trained with an 18-layer encoder, an 18-layer decoder and 512 hidden dimensions for 100K steps. The hyperparameters are detailed in the Appendix. We report the BLEU scores of all models on the test set.

First we ablate the effect of DEEPNORM and its initialization. Table 11 shows that removing the initialization leads to the degradation of performance, 0.2 BLEU dropped compared with DEEPNET. Besides, removing DEEPNORM results in the divergence.

Moreover, we ablate different values of shortcut scale α and initialization scale β of DEEPNORM. Let α^* and β^* denote the parameters for DEEPNORM. We set the scale of shortcut as α^* and vary β from $\{0.25, 0.5, 1, 2, 4\}\beta^*$. Then we set the scale of initialization as β^* and vary α from $\{0.25, 0.5, 1, 2, 4\}\alpha^*$. Figure 9 shows that small shortcut scale and large initialization scale results in the instability of the training, while large shortcut scale and small initialization scale tends to undermine the performance. Therefore, in this work, we use α and β of similar magnitude to achieve the balance between good performance and stable training.

11 RELATED WORK

Transformers have achieved success across many fields, including machine translation [1], [61], language modelling [2], [4], [5], [45], speech recognition [62], vision pre-training [31], [53], [54] and vision-language pre-training [63]. Despite their great success, the Transformers suffer a lot from the instability of their optimization, which increases the cost of the training for large-scale models. Most successful implementations involve warmup stage, Adam optimizer and layer normalization.

There are a lot of efforts to understand the effect of these components and improve the stability of Transformers. For Post-LN Transformers, Liu *et al.* [64] claimed that the necessity of warmup stage comes from reducing large variance of Adam optimizer at the early stage. They further proposed RAdam to rectify the variance of the adaptive learning rate. Zhang *et al.* [16] showed that a depth-scaled initialization can reduce the output variance of residual connections to ease gradient vanishing through layer normalization. Liu *et al.* [20] argued that the gradient vanishing of decoder is eased by Adam, and heavy dependency on Post-LN's residual

branches amplifies small parameter perturbations, leads to significant disturbances in the model output. Wang *et al.* [19] adopted densely connected layers to train deep Transformer for machine translation.

Except for Post-LN Transformers, Xiong *et al.* [27], Wang *et al.* [19] and Nguyen *et al.* [14] empirically validated that Pre-LN Transformers are easier to be optimized, and the warmup stage can be safely removed. Xiong *et al.* [27] found that warmup stage also helps quiet a lot for other optimizer (e.g. Stochastic Gradient Descent). They further proved that for Post-LN Transformers, the gradients' scale in deep layers is larger. Thus they argued that explosive gradients in deep layers of Post-LN require warmup stage to stabilize. Ding *et al.* [65] proposed the precision bottleneck relaxation and sandwich-LN to stabilize the training. Normformer [15] introduced head-scaled attention mechanism and extra normalization to improve the performance and speed up training for language modeling.

Another line of research aims to train LayerNorm-free Transformers. Bachlechner *et al.* [21] introduced ReZero, which removed layer normalization and set the weights of residual branches as zero. ReZero successfully trained very deep Transformer and achieved faster training and better performance for language modeling. Zhang *et al.* [17] first showed that a deep residual network with CNN, MLP blocks can be successfully trained without normalization. They proposed a weight initialization named Fixup to constraint explosive variance of model's update, and added extra layers to preserve model's capability. Inspired by this work, Huang *et al.* [18] further analysed the magnitude of attention module and proposed a weight initialization named T-Fixup for deep LayerNorm-free Transformer. With analysis framework of T-Fixup [18], Xu *et al.* [26] proposed a data-dependent initialization strategy for vanilla and relation-aware Transformer on pre-trained encodings.

12 Conclusion

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. Extensive experimental results verify the effectiveness of our methods across various tasks, including machine translation, language modeling (i.e., BERT, GPT) and vision pre-training (i.e., BEiT), which makes DeepNet a promising option for scaling up any Transformer models. In the future, we will extend DeepNet to support more diverse tasks, e.g., protein structure prediction [66].

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[3] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *ACL 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., 2020, pp. 8440–8451.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *NeurIPS*, 2020.

[6] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. X. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *NeurIPS*, 2019, pp. 103–112.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[8] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," in *ICLR*, 2021.

[9] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, H. F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, E. Rutherford, T. Hennigan, J. Menick, A. Cassirer, R. Powell, G. van den Driessche, L. A. Hendricks, M. Rauh, P. Huang, A. Glaese, J. Welbl, S. Dathathri, S. Huang, J. Uesato, J. Mellor, I. Higgins, A. Creswell, N. McAleese, A. Wu, E. Elsen, S. M. Jayakumar, E. Buchatskaya, D. Budden, E. Sutherland, K. Simonyan, M. Paganini, L. Sifre, L. Martens, X. L. Li, A. Kuncoro, A. Nematzadeh, E. Gribovskaya, D. Donato, A. Lazaridou, A. Mensch, J. Lespiau, M. Tsimpoukelli, N. Grigorev, D. Fritz, T. Sottiaux, M. Pajarskas, T. Pohlen, Z. Gong, D. Toyama, C. de Masson d'Autume, Y. Li, T. Terzi, V. Mikulik, I. Babuschkin, A. Clark, D. de Las Casas, A. Guy, C. Jones, J. Bradbury, M. Johnson, B. A. Hechtman, L. Weidinger, I. Gabriel, W. S. Isaac, E. Lockhart, S. Osindero, L. Rimell, C. Dyer, O. Vinyals, K. Ayoub, J. Stanway, L. Bennett, D. Hassabis, K. Kavukcuoglu, and G. Irving, "Scaling language models: Methods, analysis & insights from training gopher," *CoRR*, vol. abs/2112.11446, 2021.

[10] X. V. Lin, T. Mihaylov, M. Artetxe, T. Wang, S. Chen, D. Simig, M. Ott, N. Goyal, S. Bhosale, J. Du, R. Pasunuru, S. Shleifer, P. S. Koura, V. Chaudhary, B. O'Horo, J. Wang, L. Zettlemoyer, Z. Kozareva, M. T. Diab, V. Stoyanov, and X. Li, "Few-shot learning with multilingual language models," *CoRR*, vol. abs/2112.10668, 2021.

[11] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zheng, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, and B. Catanzaro, "Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model," *CoRR*, vol. abs/2201.11990, 2022.

[12] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *CoRR*, vol. abs/2101.03961, 2021.

[13] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. Bosma, Z. Zhou, T. Wang, Y. E. Wang, K. Webster, M. Pellat, K. Robinson, K. Meier-Hellstern, T. Duke, L. Dixon, K. Zhang, Q. V. Le, Y. Wu, Z. Chen, and C. Cui, "Glam: Efficient scaling of language models with mixture-of-experts," *CoRR*, vol. abs/2112.06905, 2021.

[14] T. Q. Nguyen and J. Salazar, "Transformers without tears: Improving the normalization of self-attention," *CoRR*, vol. abs/1910.05895, 2019.

[15] S. Shleifer, J. Weston, and M. Ott, "Normformer: Improved transformer pretraining with extra normalization," *CoRR*, vol. abs/2110.09456, 2021.

[16] B. Zhang, I. Titov, and R. Sennrich, "Improving deep transformer with depth-scaled initialization and merged attention," in *EMNLP-IJCNLP*, 2019, pp. 898–909.

[17] H. Zhang, Y. N. Dauphin, and T. Ma, "Fixup initialization: Residual learning without normalization," in *ICLR*, 2019.

[18] X. S. Huang, F. Pérez, J. Ba, and M. Volkovs, "Improving transformer optimization through better initialization," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 4475–4483.

[19] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," in *ACL*, 2019, pp. 1810–1822.

[20] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han, "Understanding the difficulty of training transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5747–5763.

[21] T. Bachlechner, B. P. Majumder, H. H. Mao, G. W. Cottrell, and J. J. McAuley, "Rezero is all you need: Fast convergence at large depth," *CoRR*, vol. abs/2003.04887, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *NeurIPS*, 2018, pp. 6391–6401.

[24] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, M. Auli, and A. Joulin, "Beyond english-centric multilingual machine translation," *J. Mach. Learn. Res.*, vol. 22, pp. 107:1–107:48, 2021.

[25] H. Wang, S. Ma, S. Huang, L. Dong, W. Wang, Z. Peng, Y. Wu, P. Bajaj, S. Singhal, A. Benhaim, B. Patra, Z. Liu, V. Chaudhary, X. Song, and F. Wei, "Magneto: A foundation transformer," in *International Conference on Machine Learning, ICML 2023*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds.

[26] P. Xu, D. Kumar, W. Yang, W. Zi, K. Tang, C. Huang, J. C. K. Cheung, S. J. D. Prince, and Y. Cao, "Optimizing deeper transformers on small datasets," in *ACL/IJCNLP*, 2021, pp. 2089–2102.

[27] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 10 524–10 533.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS 2010*, Y. W. Teh and D. M. Titterington, Eds.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*, 2015.

[30] Y. Hao, L. Dong, F. Wei, and K. Xu, "Visualizing and understanding the effectiveness of BERT," in *EMNLP-IJCNLP 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 4141–4150.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[32] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: open and efficient foundation language models," *CoRR*, vol. abs/2302.13971, 2023.

[33] B. Zhang, P. Williams, I. Titov, and R. Sennrich, "Improving massively multilingual neural machine translation and zero-shot translation," in *ACL 2020*. Association for Computational Linguistics, 2020, pp. 1628–1639.

[34] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, October 31 - November 1, 2018*, O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, C. Monz, M. Negri, A. Névéol, M. L. Neves, M. Post, L. Specia, M. Turchi, and K. Verspoor, Eds. Association for Computational Linguistics, 2018, pp. 186–191.

[35] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," *CoRR*, vol. abs/2203.15556, 2022.

[36] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020.

[37] H. Schwenk, G. Wenzek, S. Edunov, E. Grave, A. Joulin, and A. Fan, "CCMatrix: Mining billions of high-quality parallel sentences on the web," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., 2021, pp. 6490–6500.

[38] A. El-Kishky, V. Chaudhary, F. Guzmán, and P. Koehn, "CCAligned: A massive collection of cross-lingual web-document pairs," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., 2020, pp. 5960–5969.

[39] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014*. The Association for Computer Linguistics, 2014, pp. 12–58.

[40] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2017 conference on machine translation (WMT17)," in *Proceedings of the Second Conference on Machine Translation, WMT*, O. Bojar, C. Buck, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, and J. Kreutzer, Eds., 2017, pp. 169–214.

[41] O. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow, P. Koehn, and C. Monz, "Findings of the 2018 conference on machine translation (WMT18)," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018*, O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, C. Monz, M. Negri, A. Névéol, M. L. Neves, M. Post, L. Specia, M. Turchi, and K. Verspoor, Eds.

[42] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri, "Findings of the 2019 conference on machine translation (WMT19)," in *Proceedings of the Fourth Conference on Machine Translation, WMT*, O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, A. Martins, C. Monz, M. Negri, A. Névéol, M. L. Neves, M. Post, M. Turchi, and K. Verspoor, Eds. Association for Computational Linguistics, 2019, pp. 1–61.

[43] Y. Qi, D. S. Sachan, M. Felix, S. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 529–535.

[44] N. Goyal, C. Gao, V. Chaudhary, P. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan, "The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation," *CoRR*, vol. abs/2106.03193, 2021.

[45] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[46] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *BlackboxNLP*, 2018, pp. 353–355.

[47] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023.

[48] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, and et al., "Llama 2: open foundation and fine-tuned chat models," *CoRR*, vol. abs/2307.09288, 2023.

[49] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "WinoGrande: An adversarial winograd schema challenge at scale," in *AAAI*, 2020, pp. 8732–8740.

[50] H. J. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," in *Principles of Knowledge Representation and Reasoning*, 2012.

[51] N. Mostafazadeh, M. Roth, A. Louis, N. Chambers, and J. Allen, "Lsdsem 2017 shared task: The story cloze test," in *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 2017, pp. 46–51.

[52] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in *ACL*, 2019, pp. 4791–4800.

[53] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022.

[54] Z. Peng, L. Dong, H. Bao, Q. Ye, and F. Wei, "BEiT v2: Masked image modeling with vector-quantized visual tokenizers," *ArXiv*, vol. abs/2208.06366, 2022.

[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and

L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.
[56] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019.
[57] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *IEEE CVPR*, 2021.
[58] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *IEEE ICCV*, 2021.
[59] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 506–10 518.
[60] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *ECCV*, 2018.
[61] S. Ma, L. Dong, S. Huang, D. Zhang, A. Muzio, S. Singhal, H. H. Awadalla, X. Song, and F. Wei, "DeltaLM: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders," *CoRR*, vol. abs/2106.13736, 2021.
[62] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.
[63] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. Mohammed, S. Singhal, S. Som, and F. Wei, "Image as a foreign language: BEiT pretraining for all vision and vision-language tasks," *ArXiv*, vol. abs/2208.10442, 2022.
[64] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *ICLR*, 2020.
[65] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang, "Cogview: Mastering text-to-image generation via transformers," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 19 822–19 835.
[66] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.
[67] R. Karakida, S. Akaho, and S. Amari, "Universal statistics of fisher information in deep neural networks: Mean field approach," in *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, K. Chaudhuri and M. Sugiyama, Eds.

Hongyu Wang received the B.E. degree from the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, in 2022. He is currently working toward the Ph. D. degree in School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include deep learning, natural language processing and computer vision.

Shuming Ma is a senior researcher at Microsoft Research Asia. Before joining MSRA in 2019, he received his Master's and Bachelor's degrees from Peking University, with a focus on natural language processing. His research interests are large-scale language model pre-training and multilingual NLP. He has published 30+ papers at top-tier conferences (e.g. ICML, ACL, EMNLP, NAACL).

Li Dong is a Principal Researcher at Microsoft Research Asia, working on multimodal learning, and human language technology. He received his PhD in School of Informatics at University of Edinburgh in 2019.

Shaohan Huang received the B.S. and M.S. degrees from Beihang University, Beijing, China in 2014 and 2017, respectively. He is currently a senior researcher at Microsoft Research Asia, Beijing, China. His current research interests include deep learning, and natural language processing.

Dongdong Zhang is a principal researcher at Microsoft Research Asia. His research interests are neural machine translation, large-scale language model pre-training, multilingual generation, etc.

Furu Wei received the B.E. and Ph.D. degree from the Department of Computer Science, Wuhan University, Wuhan, China, in 2004 and 2009 respectively. He is currently a Partner Research Manager at Microsoft Research Asia, Beijing, China, where he is leading the Natural Language Processing group and overseeing the team's research on Foundation Models (across tasks, languages and modalities) and AGI, NLP, MT, Speech and Multimodal AI.