# Lipschitz Constrained Parameter Initialization for Deep Transformers

**Hongfei Xu**[1,2]   **Qiuhui Liu**[3*]   **Josef van Genabith**[1,2]   **Deyi Xiong**[4]   **Jingyi Zhang**[2]

[1]Saarland University / Saarland, Germany
[2]German Research Center for Artificial Intelligence / Saarland, Germany
[3]China Mobile Online Services / Henan, China
[4]Tianjin University / Tianjin, China
{hfxunlp, liuqhano}@foxmail.com, Josef.Van_Genabith@dfki.de,
dyxiong@tju.edu.cn, Jingyi.Zhang@dfki.de

## Abstract

The Transformer translation model employs residual connection and layer normalization to ease the optimization difficulties caused by its multi-layer encoder/decoder structure. Previous research shows that even with residual connection and layer normalization, deep Transformers still have difficulty in training, and particularly Transformer models with more than 12 encoder/decoder layers fail to converge. In this paper, we first empirically demonstrate that a simple modification made in the official implementation, which changes the computation order of residual connection and layer normalization, can significantly ease the optimization of deep Transformers. We then compare the subtle differences in computation order in considerable detail, and present a parameter initialization method that leverages the Lipschitz constraint on the initialization of Transformer parameters that effectively ensures training convergence. In contrast to findings in previous research we further demonstrate that with Lipschitz parameter initialization, deep Transformers with the *original* computation order can converge, and obtain significant BLEU improvements with up to 24 layers. In contrast to previous research which focuses on deep encoders, our approach additionally enables Transformers to also benefit from deep decoders.

## 1 Introduction

Neural machine translation has achieved great success in the last few years (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). The Transformer (Vaswani et al., 2017), which has outperformed previous RNN/CNN based translation models (Bahdanau et al., 2015; Gehring et al., 2017), is based on multi-layer self-attention networks and can be trained very efficiently. The multi-layer structure allows the Transformer to model complicated functions. Increasing the depth of models can increase their capacity but may also cause optimization difficulties (Mhaskar et al., 2017; Telgarsky, 2016; Eldan and Shamir, 2016; He et al., 2016; Bapna et al., 2018). In order to ease optimization, the Transformer employs residual connection and layer normalization techniques which have been proven useful in reducing optimization difficulties of deep neural networks for various tasks (He et al., 2016; Ba et al., 2016).

However, even with residual connections and layer normalization, deep Transformers are still hard to train: the original Transformer (Vaswani et al., 2017) only contains 6 encoder/decoder layers. Bapna et al. (2018) show that Transformer models with more than 12 encoder layers fail to converge, and propose the Transparent Attention (TA) mechanism which combines outputs of all encoder layers into a weighted encoded representation. Wang et al. (2019) find that deep Transformers with proper use of layer normalization are able to converge and propose to aggregate previous layers' outputs for each layer. Wu et al. (2019) explore incrementally increasing the depth of the Transformer Big by freezing pre-trained shallow layers. Concurrent work closest to ours is Zhang et al. (2019). They address the same issue, but propose a different layer-wise initialization approach to reduce the standard deviation.

Our contributions are as follows:

- We empirically demonstrate that a simple modification made in the Transformer's official implementation (Vaswani et al., 2018) which changes the computation order of residual connection and layer normalization can effectively ease its optimization;

- We deeply analyze how the subtle difference of computation order affects convergence in
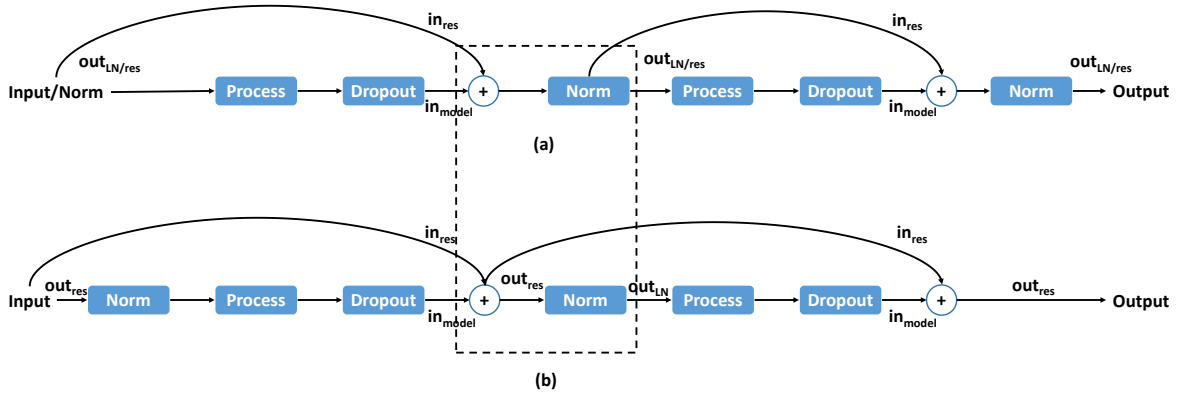
---

* Corresponding author.

Figure 1: Two Computation Sequences of Transformer Translation Models: (a) the one used in the original paper, (b) the official implementation. We suggest to regard the output of layer normalization ($out_{LN/res}$) as the output of residual connection rather than the addition of $in_{res}$ and $in_{model}$ for (a), because it ($out_{LN/res}$) is the input ($in_{res}$) of the next residual connection computation.

deep Transformers, and propose to initialize deep Transformers under the Lipschitz constraint;

- In contrast to previous works, we empirically show that with proper parameter initialization, deep Transformers with the original computation order can converge;

- Our simple approach effectively ensures the convergence of deep Transformers with up to 24 layers, and achieves $+1.50$ and $+0.92$ BLEU improvements over the baseline on the WMT 14 English to German task and the WMT 15 Czech to English task;

- We further investigate deep decoders for the Transformer in addition to the deep encoders studied in previous works, and show that deep decoders can also benefit the Transformer.

## 2 Convergence of Different Computation Orders

In this paper we focus on the convergence of the training of deep transformers. To alleviate the training problem for the standard Transformer model, Layer Normalization (Ba et al., 2016) and Residual Connection (He et al., 2016) are adopted.

### 2.1 Empirical Study of the Convergence Issue

The official implementation (Vaswani et al., 2018) of the Transformer uses a different computation order (Figure 1 b) compared to the published version (Vaswani et al., 2017) (Figure 1 a), since it (Fig-

ure 1 b) seems better for harder-to-learn models.[1] Even though several studies (Chen et al., 2018; Domhan, 2018) have mentioned this change and although Wang et al. (2019) analyze the difference between the two computation orders during back-propagation, and Zhang et al. (2019) point out the effects of normalization in their work, how this modification impacts on the performance of the Transformer, especially for deep Transformers, has not been deeply studied before. Here we present both empirical convergence experiments (Table 1) and a theoretical analysis of the effect of the interaction between layer normalization and residual connection (Table 2).

In order to compare with Bapna et al. (2018), we used the same datasets from the WMT 14 English to German task and the WMT 15 Czech to English task for our experiments. We applied joint Byte-Pair Encoding (BPE) (Sennrich et al., 2016) with 32k merge operations. We used the same setting as the Transformer base (Vaswani et al., 2017) except the number of warm-up steps was set to $8k$.

Parameters were initialized with Glorot Initialization[2] (Glorot and Bengio, 2010) like in many other Transformer implementations (Klein et al., 2017; Hieber et al., 2017; Vaswani et al., 2018). We conducted experiments based on the Neutron implementation (Xu and Liu, 2019) of the Transformer translation model. Our experiments run on 2 GTX

---

| Models | Layers | | En-De | | Cs-En | |
|---|---|---|---|---|---|---|
| | Encoder | Decoder | v1 | v2 | v1 | v2 |
| Bapna et al. (2018)* | 16 | 6 | 28.39 | None | 29.36 | None |
| Wang et al. (2019) | 30 | 6 | 29.3 | | | |
| Wu et al. (2019) | 8 | | 29.92 | None | | |
| Zhang et al. (2019) | 20 | | 28.67 | | | |
| Transformer* | 6 | | 27.77‡ | 27.31 | 28.62 | 28.40 |
| | 12 | | ¬ | 28.12 | ¬ | 29.38 |
| | 18 | | ¬ | 28.60 | ¬ | 29.61 |
| | 24 | | ¬ | **29.02** | ¬ | **29.73** |

Table 1: Results of Different Computation Orders. "¬" means fail to converge, "None" means not reported in original works, "*" indicates our implementation of their approach. † and ‡ mean $p < 0.01$ and $p < 0.05$ while comparing between v1 (the official publication) and v2 (the official implementation) with the same number of layers in the significance test. Wu et al. (2019) use the Transformer Big setting, while the others are based on the Transformer Base Setting. Zhang et al. (2019) use merged attention decoder layers with a $50k$ batchsize.

| v1 | v2 |
|---|---|
| $\mu = mean(in_{model} + in_{res})$ | |
| $\sigma = std(in_{model} + in_{res})$ | |
| $out_{LN} = \frac{(in_{model}+in_{res}-\mu)}{\sigma} * w + b$ | |
| $out_{res}^{v1} = out_{LN} = \frac{w}{\sigma} * out_{res}^{v2} - \frac{w}{\sigma}*\mu + b$ | $out_{res}^{v2} = in_{res} + in_{model}$ |

Table 2: Computation with Layer Normalization and Residual Connection. v1 and v2 stand for the computation order of the original Transformer paper and that of the official implementation respectively. "mean" and "std" are the computation of mean value and standard deviation. $in_{model}$ and $in_{res}$ stand for output of current layer and accumulated outputs from previous layers respectively. $w$ and $b$ are weight and bias of layer normalization which are initialized with a vector full of 1s and another vector full of 0s. $out_{LN}$ is the computation result of the layer normalization. $out_{res}^{v1}$ and $out_{res}^{v2}$ are results of residual connections of v1 and v2.

1080 Ti GPUs, and a batch size of $25k$ target tokens is achieved through gradient accumulation of small batches.

We used a beam size of 4 for decoding, and evaluated tokenized case-sensitive BLEU with the averaged model of the last 5 checkpoints saved with an interval of 1,500 training steps.

Results of the two different computation orders are shown in Table 1, which shows that deep Transformers with the computation order of the official implementation (v2) have no convergence issue.

## 2.2 Theoretical Analysis

Since the subtle change of computation order results in large differences in convergence, we further analyze the differences between the computation orders to investigate how they affect convergence.

We conjecture that the convergence issue of deep Transformers is perhaps due to the fact that layer normalization over residual connections in Figure 1 (a) effectively reduces the impact of residual connections due to subsequent layer normalization, in order to avoid a potential explosion of combined layer outputs (Chen et al., 2018), which is also studied by Wang et al. (2019); Zhang et al. (2019). We therefore investigate how the layer normalization and the residual connection are computed in the two computation orders, shown in Table 2.

Table 2 shows that the computation of residual connection in v1 is weighted by $\frac{w}{\sigma}$ compared to v2, and the residual connection of previous layers will be shrunk if $\frac{w}{\sigma} < 1.0$, which makes it difficult for deep Transformers to converge.

## 3 Lipschitz Constrained Parameter Initialization

Since the diminished residual connections (Table 2) may cause the convergence issue of deep v1 Transformers, is it possible to constrain $\frac{w}{\sigma} \geq 1.0$? Given that $w$ is initialized with 1, we suggest that the standard deviation of $in_{model} + in_{res}$ should be constrained as follows:

$$0.0 < \sigma = std(in_{model} + in_{res}) \leq 1.0 \quad (1)$$

in which case $\frac{w}{\sigma}$ will be greater than or at least equal to 1.0, and the residual connection of v1 will not be shrunk anymore. To achieve this goal, we can constrain elements of $in_{model} + in_{res}$ to be in $[a, b]$ and ensure that their standard deviation is smaller than 1.0.

Let's define $P(x)$ as any probability distribution of $x$ between $[a, b]$:

$$\int_a^b P(x)dx = 1.0 \qquad (2)$$

then the standard deviation of $x$ is:

$$\sigma(P(x), x) = \sqrt{\int_a^b P(x)\left(x - \int_a^b P(x)xdx\right)^2 dx} \qquad (3)$$

Given that $(x - \int_a^b P(x)xdx) < (b - a)$ for x $\in$ $[a, b]$ as $P(x)$ is constrained by Equation 2, we reformulate Equation 3 as follows:

$$\sigma(P(x), x) < \sqrt{\int_a^b P(x)(b - a)^2 dx} \qquad (4)$$

From Equation 4 we obtain:

$$\sigma(P(x), x) < (b - a)\sqrt{\int_a^b P(x)dx} \qquad (5)$$

After applying Equation 2 in Equation 5, we find that:

$$\sigma(P(x), x) < b - a \qquad (6)$$

Thus, as long as $b - a \le 1$ (the range of elements of the representation $x$), the requirements for the corresponding $\sigma$ described in Equation 1 can be satisfied.

To achieve this goal, we can simply constrain the range of elements of $x$ to be smaller than 1 and initialize the sub-model before layer normalization to be a k-Lipschitz function, where $k \le 1$. Because if the function $F$ of the sub-layer is a k-Lipschitz function, for inputs x, y $\in$ [a, b], $|F(x) - F(y)| <$ $k|x - y|$ holds. Given that $|x - y| \le b - a$, we can get $|F(x) - F(y)| < k(b - a)$, the range of the output of that sub-layer is constrained by making it a k-Lipschitz function with constrained input.

| Layers | En-De | | Cs-En | |
|---|---|---|---|---|
| | v1-L | v2-L | v1-L | v2-L |
| 6 | 27.96$^\dagger$ | 27.38 | 28.78$^\ddagger$ | 28.39 |
| 12 | 28.67$^\dagger$ | 28.13 | 29.17 | 29.45 |
| 18 | 29.05$^\ddagger$ | 28.67 | 29.55 | 29.63 |
| 24 | **29.46** | 29.20 | 29.70 | **29.88** |

Table 3: Results with Lipschitz Constrained Parameter Initialization.

The k-Lipschitz constraint can be satisfied effectively through weight clipping,[3] and we empirically find that deep Transformers are only hard to train at the beginning and only applying a constraint to parameter initialization is sufficient, which is more efficient and can avoid a potential risk of weight clipping on performance. Zhang et al. (2019) also show that decreasing parameter variance at the initialization stage is sufficient for ensuring the convergence of deep Transformers, which is consistent with our observation.

## 4 Experiments

We use the training data described in Section 2 to examine the effectiveness of the proposed Lipschitz constrained parameter initialization approach.

In practice, we initialize embedding matrices and weights of linear transformations with uniform distributions of $[-e, +e]$ and $[-l, +l]$ respectively. We use $\sqrt{\frac{2}{esize + vsize}}$ as $e$ and $\sqrt{\frac{1}{isize}}$ as $l$ where $esize$, $vsize$ and $isize$ stand for the size of embedding, vocabulary size and the input dimension of the linear transformation respectively.[4]

Results for two computation orders with the new parameter initialization method are shown in Table 3. v1-L indicates v1 with Lipschitz constrained parameter initialization, the same for v2-L.

Table 3 shows that deep v1-L models do not suffer from convergence problems anymore with our new parameter initialization approach. It is also worth noting that unlike Zhang et al. (2019), our parameter initialization approach does not degrade the translation quality of the 6-layer Transformer, and the 12-layer Transformer with our approach already achieves performance comparable to the 20-layer Transformer in Zhang et al. (2019) (shown in Table 1).

---

[3]Note that the weight of the layer normalization cannot be clipped, otherwise residual connections will be more heavily shrunk.

[4]To preserve the magnitude of the variance of the weights in the forward pass.

| Encoder | Decoder | En-De | Cs-En |
|---|---|---|---|
| 6 | | 27.96 | 28.78 |
| 24 | 6 | 28.76 | 29.20 |
| 6 | 24 | 28.63 | 29.36 |
| 24 | | **29.46** | **29.70** |

Table 4: Effects of Encoder and Decoder Depth with Lipschitz Constrained Parameter Initialization.

While previous approaches (Bapna et al., 2018; Wang et al., 2019) only increase the depth of the encoder, we suggest that deep decoders should also be helpful. We analyzed the influence of deep encoders and decoders separately and results are shown in Table 4.

Table 4 shows that the deep decoder can indeed benefit performance in addition to the deep encoder, especially on the Czech to English task.

## 5 Conclusion

In contrast to previous works (Bapna et al., 2018; Wang et al., 2019; Wu et al., 2019) which show that deep Transformers with the computation order as in Vaswani et al. (2017) have difficulty in convergence, we show that deep Transformers with the original computation order can converge as long as proper parameter initialization is performed.

We first investigate convergence differences between the published Transformer (Vaswani et al., 2017) and its official implementation (Vaswani et al., 2018), and compare the differences of computation orders between them. We conjecture that the convergence issue of deep Transformers is because layer normalization sometimes shrinks residual connections, we support our conjecture with a theoretical analysis (Table 2), and propose a Lipschitz constrained parameter initialization approach for solving this problem.

Our experiments show the effectiveness of our simple approach on the convergence of deep Transformers, which achieves significant improvements on the WMT 14 English to German and the WMT 15 Czech to English news translation tasks. We also study the effects of deep decoders in addition to deep encoders extending previous works.

## Acknowledgments

## References

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808. Association for Computational Linguistics.

Ronen Eldan and Ohad Shamir. 2016. The power of depth for feedforward neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940, Columbia University, New York, New York, USA. PMLR.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine*

*Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

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

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. 2017. When and why are deep networks better than shallow ones? In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2343–2348.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Matus Telgarsky. 2016. benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539, Columbia University, New York, New York, USA. PMLR.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy. Association for Computational Linguistics.

Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint arXiv:1903.07402*.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 897–908, Hong Kong, China. Association for Computational Linguistics.