# On Orthogonality Constraints for Transformers

**Aston Zhang**[‡,*], **Alvin Chan**[◇,*], **Yi Tay**[†], **Jie Fu**[◁], **Shuohang Wang**[○], **Shuai Zhang**[•], **Huajie Shao**[▷], **Shuochao Yao**[⋆], **Roy Ka-Wei Lee**[^]
[‡]AWS, [◇]NTU Singapore, [†]Google, [◁]Mila, Université de Montréal
[○]SMU, [•]ETH Zürich, [▷]UIUC, [⋆]George Mason University, [^]SUTD
az@astonzhang.com

## Abstract

Orthogonality constraints encourage matrices to be orthogonal for numerical stability. These plug-and-play constraints, which can be conveniently incorporated into model training, have been studied for popular architectures in natural language processing, such as convolutional neural networks and recurrent neural networks. However, a dedicated study on such constraints for transformers has been absent. To fill this gap, this paper studies orthogonality constraints for transformers, showing the effectiveness with empirical evidence from ten machine translation tasks and two dialogue generation tasks. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

## 1 Introduction

Transformers (Vaswani et al., 2017) are a class of neural architectures that have made a tremendous transformative impact on modern natural language processing research and applications. Transformers have not only served as a powerful inductive bias for general-purpose sequence transduction (Ott et al., 2018) but also lived as the core of large pre-trained language models (Devlin et al., 2018; Radford et al., 2018; Dai et al., 2019). That said, the study of more effective training for this class of models is still an open research question, bearing great potential to impact a myriad of applications and domains.

To improve numerical stability during training, the trick of enforcing orthogonality constraints has surfaced recently. In the analysis of numerical stability, enforcing orthogonality constraints can upper-bound the Lipschitz constant of linear transformations. The Lipschitz constant is a measure that approximates the rate of change (variation) of representations. Theoretically, controlling the Lipschitz constant, which may be achieved via orthogonality constraints, yields representations that are robust and less sensitive to perturbations.

In view of this, orthogonality constraints have been studied for convolutional neural networks (CNNs) (Bansal et al., 2018; Huang et al., 2018) and recurrent neural networks (RNNs) (Arjovsky et al., 2016; Vorontsov et al., 2017; Rodríguez et al., 2016). Such plug-and-play constraints can be incorporated into model training without additional hassle. For example, CNN-based models incorporating orthogonality constraints have demonstrated empirical effectiveness for tasks such as person re-identification (Han et al., 2019) and keyword spotting (Lee et al., 2019), while RNN-based models that enforce such constraints have shown promising empirical results for response generation (Tao et al., 2018) and text classification (Wei et al., 2020; Krishnan et al., 2020). However, a dedicated study on orthogonality constraints for transformers has been absent so far.

To fill this research gap, we study orthogonality constraints for transformers, which are imposed on (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention. Mathematically, orthogonality constraints on the weights of these linear transformations can be motivated by bounded Lipschitz constants. We also formally analyze the self-attention mechanism by bounding perturbations to the affinity matrix in the face of input changes.

Furthermore, we conduct extensive experiments on ten neural machine translation (both subword-

[*]Equal contribution.
[†]Work was done at NTU.



*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers)*, pages 375–382
August 1–6, 2021. 

level and character-level) tasks and two dialogue generation tasks. Our experimental results are promising, demonstrating that the performance of transformers can be consistently boosted with orthogonality constraints. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

**Notation** For any vector $\mathbf{x}$ and any matrix $\mathbf{X}$, $\|\mathbf{x}\|$ and $\|\mathbf{X}\|$ denote their $L_2$-norm and spectral norm, respectively.

## 2 Orthogonality Constraints for Transformers

Recall that in the transformer architecture, keys, queries, and values all come from the same place in the self-attention module. They are linearly transformed for computing multiple attention heads, where all the heads are aggregated by another linear transformation. The position-wise feed-forward network is also built on two linear transformations with activations. In the following, we will describe orthogonality constraints for (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention.

### 2.1 For Linear Transformations in Self-Attention and Position-wise Feed-Forward Networks

Note that linear transformations in self-attention and position-wise feed-forward networks are in the form:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b},$$

where $\mathbf{y}$ is the output, $\mathbf{x}$ is an input, $\mathbf{W}$ is a linear transformation weight matrix, and $\mathbf{b}$ is an optional bias term. This form provides us with convenient tools for motivating the application of orthogonality constraints to the weights of such linear transformations.

Specifically, as described in Section 1, robustness of linear transformations to small perturbations can be measured by Lipschitz constants. Thus, we begin with motivating orthogonality constraints from the perspective of bounding Lipschitz constants of linear transformations.

Formally, the linear transformation (layer) of the aforementioned form $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$ has a Lipschitz constant equal to the largest singular value of the weight matrix $\mathbf{W}$. The linear layer is Lipschitz continuous with the constant $L$ if for all $\mathbf{x}$ and $\mathbf{x}'$, it holds that

$$\|(\mathbf{W}\mathbf{x} + \mathbf{b}) - (\mathbf{W}\mathbf{x}' + \mathbf{b})\| \leq L\|\mathbf{x} - \mathbf{x}'\|,$$

which can be re-written as

$$\frac{\|\mathbf{W}(\mathbf{x} - \mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|} \leq L,$$

where $\mathbf{x} \neq \mathbf{x}'$. Therefore, the smallest Lipschitz constant is

$$\sup_{\mathbf{x} \neq \mathbf{x}'} \frac{\|\mathbf{W}(\mathbf{x} - \mathbf{x}')\|}{\|\mathbf{x} - \mathbf{x}'\|}.$$

For numerical stability, our goal is to force the Lipschitz constant to be no greater than one at every linear transformation so that their multiplication throughout compositions of transformations is also upper bounded by one. Mathematically, we need to constrain the Lipschitz constant (the largest singular value) of $\mathbf{W}$ to be no greater than one, which requires the following orthogonality constraint:

$$\mathbf{W}^\top \mathbf{W} \approx \mathbf{I}.$$

Back to the context of multi-head self-attention of transformers, denote by $\mathbf{P}$ the concatenation of the linear transformation weights for the query, key, value, and the multi-head aggregation. To impose the orthogonality constraint for these linear transformations, we add the following loss to the transformer model for every layer:

$$L_{\text{LA}} = \lambda \|\mathbf{P}^\top \mathbf{P} - \mathbf{I}\|_F^2.$$

Likewise, for position-wise feed-forward network with two linear transformation weight matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, the orthogonality constraint can be imposed with another additional loss:

$$L_{\text{LF}} = \lambda \left[ \|\mathbf{M}_1^\top \mathbf{M}_1 - \mathbf{I}\|_F^2 + \|\mathbf{M}_2^\top \mathbf{M}_2 - \mathbf{I}\|_F^2 \right].$$

### 2.2 For the Affinity Matrix in Self-Attention

In transformers, given the query matrix $\mathbf{Q}$ and the key matrix $\mathbf{K}$ in the self-attention module, the affinity matrix

$$\mathbf{A} = \text{softmax}(\alpha \mathbf{Q}\mathbf{K}^\top), \tag{1}$$

where $\alpha$ is typically $\frac{1}{\sqrt{d}}$ ($d$ is the dimension of the key and the query). Given the value matrix $\mathbf{V}$, the self-attention computes representations via the matrix multiplication $\mathbf{AV}$.

Within the context of sequence transduction, when an input word token is aligned with another semantically similar token, we would expect a small change in the behavior of the self-attention mechanism, rather than a huge change in the output. In the affinity matrix $\mathbf{A}$ as defined in (1), let $\mathbf{A}_{i,*}$ be the row vector indexed by $i$. Essentially, each $\mathbf{A}_{i,*}$ is a probability distribution over the tokens in the sequence that directs the alignment-based pooling operation. Intuitively, for a robust self-attention mechanism, noisy perturbations should have a limited effect on the affinity scores of the tokens.

More formally, let us analyze the self-attention mechanism by bounding perturbations to the affinity matrix in the face of input changes. Mathematically, changes to the affinity scores are bounded such that $\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| \leq 2\alpha\|\mathbf{K}\|\epsilon$, where $\epsilon = \|\mathbf{Q}'_{i,*} - \mathbf{Q}_{i,*}\|$ is the noise from the query matrix. We can see this as the result of the following theorem.

**Theorem 2.1** (Bounded Perturbations to the Affinity Matrix)**.** *Expressing $\mathbf{A}_{i,*}$ to be the $i^{\text{th}}$ row of the affinity matrix $\mathbf{A}$ as defined in (1) and $\mathbf{Q}_{i,*}$ to be the $i^{\text{th}}$ row of the query matrix $\mathbf{Q}$, the perturbation to the affinity matrix is bounded as such:*

$$\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| \leq 2\alpha\|\mathbf{K}\|\epsilon,$$

*where $\mathbf{A}' = softmax(\alpha\mathbf{Q}'\mathbf{K}^\top)$ and $\epsilon = \|\mathbf{Q}'_{i,*} - \mathbf{Q}_{i,*}\|$ is the $L_2$ perturbation value in $\mathbf{Q}_{i,*}$.*

The detailed proof of Theorem 2.1 is provided in the appendix. In standard training, the spectral norm of the key matrix $\|\mathbf{K}\|$ or the noise $\epsilon$ from the query matrix may be large, and as a result the changes to affinity scores may become "unbounded". We speculate that this may hurt the generalization of the self-attention mechanism.

We impose orthogonality constraints on the affinity matrix $\mathbf{A}$. More concretely, we obtain an additional loss term for every layer of the transformer model using the Frobenius norm $\|\cdot\|_F$:

$$L_{\text{AM}} = \lambda\|\mathbf{A}^\top\mathbf{A} - \mathbf{I}\|_F^2,$$

where $\mathbf{I}$ is the identity matrix and $\lambda$ is a scaling factor to control the ratio to the original task loss.

With orthogonally constrained affinity scores, each row vector of $\mathbf{A}$ is now orthonormal to all the other row vectors. Given that each row vector is a probability distribution over the tokens in the sequence that directs the alignment-based pooling operation, a diverse form of the self-attention mechanism would be more encouraged. This could be viewed as an additional quality of orthogonality constrained transformers.

## 3 Experiments

We evaluate the effectiveness of orthogonality constrained transformers (OC-transformers for brevity) on ten neural machine translation tasks and two dialogue generation tasks. Specifically, we assess three variants, largely pertaining to where orthogonality constraints are applied, i.e., (i) AM (for the affinity matrix in self-attention), (ii) LA (for the linear transformations in self-attention), and (iii) LF (for the linear transformations in position-wise feed-forward networks). We evaluate them in an incremental fashion with three main model labels: VAR-I (AM only), VAR-II (AM + LA), and VAR-III (AM + LA + LF). The scaling factor $\lambda$ is tuned amongst $\{10^{-6}, 10^{-8}, 10^{-10}\}$.

### 3.1 Neural Machine Translation

For neural machine translation (NMT), we evaluate on both the subword-level and character-level tasks.

**Experimental Setup** For subword-level NMT, we evaluate our models on seven NMT datasets using the Tensor2Tensor[1] framework (Vaswani et al., 2018), namely IWSLT'14 De→En, IWSLT'14 Ro→En, IWSLT'15 En→Vi, IWSLT'17 En→Id, WMT'17 En→Et, SETIMES En→Mk, and the well-established large-scale WMT'16 En→De.

All the models are trained with the transformer-base setting. Owing to the smaller size, we use the transformer-small setting for IWSLT'14 datasets. For the WMT'16 En→De dataset, we train both the transformer-base and transformer-big settings on $4\times$ GPUs with gradient accumulation of $2\times$ to emulate $8\times$ GPU training. By determining improvement on approximate BLEU scores on the validation set, we train models for $2M$ steps for the transformer-base setting and $800K$ steps for the transformer-big setting. Note that between the standard transformer and OC-transformer, we maintain all the other hyperparameters to keep the comparisons as fair as possible. For character-level NMT, we evaluate on three language pairs, namely

[1] https://github.com/tensorflow/tensor2tensor

Table 1: Experimental results on subword-level neural machine translation.

| Model | BLEU | | | | | |
|---|---|---|---|---|---|---|
| | De→En | Ro→En | En→Vi | En→Id | En→Et | En→Mk |
| Transformer | 34.68 | 32.36 | 28.43 | 47.40 | 14.17 | 13.96 |
| OC-transformer (VAR-I) | 34.87 | **32.68** | 30.16 | 48.09 | 14.83 | 14.74 |
| OC-transformer (VAR-II) | 34.92 | 32.63 | **30.51** | 48.05 | **15.06** | **14.70** |
| OC-transformer (VAR-III) | **35.20** | 32.44 | 30.42 | **48.33** | 14.87 | 14.62 |
| Relative Gain (%) | +1.5% | +1.0% | +7.3% | +2.0% | +6.3% | +5.3% |

Table 2: Experimental results on neural machine translation with the WMT'16 En→De *newstest2014* test set.

| Model | BLEU |
|---|---|
| MoE (Shazeer et al., 2017) | 26.0 |
| Transformer-base (Vaswani et al., 2017) | 27.3 |
| Transformer-big (Vaswani et al., 2017) | 28.4 |
| Transformer-ott-big (Ott et al., 2018) | 29.3 |
| Dynamic convolution (Wu et al., 2019) | **29.7** |
| OC-transformer-base based on (Vaswani et al., 2017) (VAR-III) | 28.5 |
| OC-transformer-big based on (Vaswani et al., 2017) (VAR-III) | 29.6 |

WMT En→Fr, IWSLT'14 Ro→En, and IWSLT'15 De→En. The transformer-small setting is used for all the three language pairs and trained for $200K$ steps.

**Experimental Results** Table 1 reports experimental results on subword-level NMT datasets. Overall, we note that performance of transformers is consistently boosted by orthogonality constraints, ascertaining the effectiveness of adopting such plug-and-play tricks. More specifically, they are able to achieve $+1.0\%$ to $+7.3\%$ relative gain over the standard transformer. Notably, all the variants (VAR-I, VAR-II, and VAR-III) boost the performance of transformers: it demonstrates that orthogonality constraints are indeed useful. Moreover, orthogonal constraints on the self-attention affinity matrix are beneficial in general even if the rest of the model is not fully enforced with orthogonality constraints.

Table 2 reports the results on the large-scale WMT'16 En→De dataset. Orthogonality constraints boost the performance of the transformer-big setting based on (Vaswani et al., 2017), increasing the BLEU from 28.4 to 29.6. This result outperforms the more advanced transformer-ott-big proposed in (Ott et al., 2018) and comes close to 29.7 that was achieved by the very competitive dynamic convolution model (Wu et al., 2019). Likewise, orthogonality constraints also boost the performance of the transformer-base setting with the BLEU increased from 27.3 to 28.5.

Table 3 reports the results on character-level NMT. We observe that orthogonality constraints consistently boost the performance of standard transformers on all the three language pairs: En→Fr (+3.5%), Ro→En (+2.6%), and De→En (+1.6%).

### 3.2 Sequence-to-Sequence Dialogue Generation

We conduct experiments on the sequence-to-sequence dialog generation task whereby the goal is to generate the reply in a two-way conversation.

**Experimental Setup** We use two datasets: PersonaChat (Zhang et al., 2018) and DailyDialog (Li et al., 2017). We implement our task in Tensor2Tensor using the transformer-small setting in a sequence-to-sequence fashion (Sutskever et al., 2014). We train all the models for $20K$ steps, which we find sufficient for model convergence. Beam search of beam size $4$ and length penalty $0.6$ is adopted for decoding the output sequence. We evaluate all the models with the language generation evaluation suite in (Sharma et al., 2017).

**Experimental Results** Table 4 reports our results on the PersonaChat and DailyDialog datasets. The key observation is that all the variants of enforcing orthogonality constraints boost performance of standard transformers. The best results of OC-transformers make a substantial improvement in all

Table 3: Experimental results on character-level neural machine translation.

| Model | BLEU | | |
|---|---|---|---|
| | En→Fr | Ro→En | De→En |
| Transformer (Vaswani et al., 2017) | 18.74 | 22.04 | 27.59 |
| OC-transformer based on (Vaswani et al., 2017) (VAR-III) | **19.40** | **22.61** | **28.02** |
| Relative Gain (%) | +3.5% | +2.6% | +1.6% |

Table 4: Experimental results on the PersonaChat dataset (Zhang et al., 2018) and the DailyDialog dataset (Li et al., 2017) on nine evaluation measures (Sharma et al., 2017). SkipT stands for SkipThought cosine similarity, EmbA stands for embedding average, VecE stands for vector extrema, and GreedyM stands for greedy matching.

| | Transformer | OC-transformer (VAR-I) | OC-transformer (VAR-II) | OC-transformer (VAR-III) | Relative Gain |
|---|---|---|---|---|---|
| | **PersonaChat** | | | | |
| BLEU-1 | 13.2 | 15.1 | 15.4 | **16.3** | +23.5% |
| BLEU-4 | 2.04 | 2.28 | 2.38 | **2.50** | +22.5% |
| Meteor | 6.10 | 6.55 | 6.60 | **6.70** | +9.8% |
| Rouge | 14.2 | 14.7 | **15.1** | **15.1** | +6.3% |
| CIDEr | 18.2 | 18.7 | **19.3** | 18.3 | +6.0% |
| SkipT | 41.9 | 42.8 | **43.9** | 43.3 | +4.8% |
| EmbA | 84.3 | 84.6 | **84.9** | 84.6 | +0.7% |
| VecE | **49.0** | 48.2 | **49.0** | 48.6 | +0.0% |
| GreedyM | 65.8 | 66.2 | **66.5** | 66.4 | +1.1% |
| | **DailyDialog** | | | | |
| BLEU-1 | 12.1 | 13.5 | 13.3 | **14.0** | +15.7% |
| BLEU-4 | 6.22 | 6.70 | 6.52 | **7.11** | +14.3% |
| Meteor | 8.23 | 8.43 | 8.39 | **8.72** | +6.0% |
| Rouge | 21.1 | 21.4 | **21.7** | **21.7** | +2.8% |
| CIDEr | 79.3 | 79.2 | 79.6 | **82.1** | +3.5% |
| SkipT | 66.9 | 67.1 | 67.1 | **67.2** | +0.4% |
| EmbA | 84.9 | **85.7** | 85.6 | 85.5 | +0.9% |
| VecE | 53.1 | 53.3 | **53.4** | 53.2 | +0.5% |
| GreedyM | 72.1 | 72.3 | **72.6** | 72.2 | +0.7% |

the nine evaluation measures. Notably, on both datasets, the best variants are either VAR-II or VAR-III. VAR-I performs decently and boosts performance of standard transformers on both tasks, signifying that the orthogonality constrained affinity matrix in self-attention is sufficiently effective. This mirrors the results on neural machine translation and is consistent across the findings. The relative gain of applying orthogonality constraints is also promising, peaking at $+23.5\%$ on BLEU-1 scores and $+2.8\%$ to $+6.3\%$ on Rouge.

## 4 Conclusion

We studied orthogonality constraints for transformers, which are imposed on (i) linear transformations in self-attention and position-wise feed-forward networks and (ii) the affinity matrix in self-attention. We showed that such plug-and-play constraints, which can be conveniently incorporated, consistently boost performance of transformers on ten different machine translation tasks and two dialogue generation tasks. For example, on the large-scale WMT'16 En→De benchmark, simply plugging-and-playing orthogonality constraints on the original transformer model (Vaswani et al., 2017) increases the BLEU from 28.4 to 29.6, coming close to the 29.7 BLEU achieved by the very competitive dynamic convolution (Wu et al., 2019).

**Broader Impact** Given widespread adoptions of transformer models, the proposed plug-and-play orthogonal constraints could also be useful to computer vision, automatic speech recognition, time series analysis, and biological sequence analysis.

## References


Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. 2018. Can we gain more from orthogonality regularizations in training deep cnns? *arXiv preprint arXiv:1810.09102*.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chuchu Han, Ruochen Zheng, Changxin Gao, and Nong Sang. 2019. Complementation-reinforced attention network for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3433–3445.

Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. 2018. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jitin Krishnan, Hemant Purohit, and Huzefa Rangwala. 2020. Diversity-based generalization for neural unsupervised text classification under domain shift. *arXiv preprint arXiv:2002.10937*.

Mingu Lee, Jinkyu Lee, Hye Jin Jang, Byeonggeun Kim, Wonil Chang, and Kyuwoong Hwang. 2019. Orthogonality constrained multi-head attention for keyword spotting. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 86–92. IEEE.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

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

Pau Rodríguez, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. 2016. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*, pages 4418–4424.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. 2017. On orthogonality and learning recurrent networks with long term dependencies. In *International Conference on Machine Learning*, pages 3570–3578. PMLR.

Jiyao Wei, Jian Liao, Zhenfei Yang, Suge Wang, and Qiang Zhao. 2020. Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing*, 383:165–173.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

## A Proof of Theorem 2.1

*Proof.* Let $\mathbf{x} = \mathbf{Q}_{i,*}$, $g(\mathbf{x}) = \mathbf{x}\mathbf{K}^\top$, and $f(\mathbf{y}) = \text{softmax}(\mathbf{y})$. Expressing each row in $\mathbf{A}$ as $\mathbf{A}_{i,*} = \text{softmax}(\alpha\mathbf{Q}_{i,*}\mathbf{K}^\top)$, we have

$$\mathbf{A}_{i,*} = f(\alpha g(\mathbf{x})). \tag{2}$$

We first consider bounding $g(\mathbf{x})$ with respect to $\|\mathbf{x}' - \mathbf{x}\|$:

$$\|g(\mathbf{x}') - g(\mathbf{x})\| = \|(\mathbf{x}' - \mathbf{x})\mathbf{K}^\top\|.$$

Recalling the definition of the spectral norm, $\|\mathbf{A}\| = \max_{\mathbf{x}\in\mathbb{R}^l\setminus\{\mathbf{0}\}} \frac{\|\mathbf{x}\mathbf{A}\|}{\|\mathbf{x}\|}$:

$$\|g(\mathbf{x}') - g(\mathbf{x})\| \leq \|\mathbf{K}\|\|\mathbf{x}' - \mathbf{x}\|. \tag{3}$$

Here, we can observe that the Lipschitz constant for $g$ is $\|\mathbf{K}\|$.

Next, we bound $f(\mathbf{y}) = \text{softmax}(\mathbf{y})$ with respect to $\|\mathbf{y}' - \mathbf{y}\|$. Since $f$ is a differentiable function, it holds that

$$\|f(\mathbf{y}') - f(\mathbf{y})\| \leq \|\mathbf{J}\|^*\|(\mathbf{y}' - \mathbf{y})\|, \tag{4}$$

where $\mathbf{J}$ is the Jacobian matrix of $f(\mathbf{y})$ with respect to $\mathbf{y}$, i.e., $\mathbf{J}_{i,j} = \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_j}$, and $\|\mathbf{J}\|^* = \max_{\mathbf{y}} \|\mathbf{J}\|$. Since $f(\mathbf{y})_i = \frac{e^{\mathbf{y}_i}}{\sum e^{\mathbf{y}_j}}$, for diagonal entries of $\mathbf{J}$ we have

$$\begin{aligned}\mathbf{J}_{i,i} &= \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_i} \\ &= \frac{e^{\mathbf{y}_i}}{\sum e^{\mathbf{y}_j}} - \frac{e^{2\mathbf{y}_i}}{(\sum e^{\mathbf{y}_j})^2} \\ &= f_i - {f_i}^2,\end{aligned}$$

where $f_i = f(\mathbf{y})_i$ for brevity. For non-diagonal entries of $\mathbf{J}$ where $i \neq j$, we have

$$\begin{aligned}\mathbf{J}_{i,j} &= \frac{\partial f(\mathbf{y})_i}{\partial \mathbf{y}_j} \\ &= -\frac{e^{\mathbf{y}_i}e^{\mathbf{y}_j}}{(\sum e^{\mathbf{y}_j})^2} \\ &= -f_i f_j.\end{aligned}$$

With this, we can express the Jacobian $\mathbf{J}$ as

$$\begin{aligned}\mathbf{J} &= \begin{bmatrix} f_1 - {f_1}^2 & \cdots & -f_1 f_n \\ \vdots & \ddots & \vdots \\ -f_n f_1 & \cdots & f_n - {f_n}^2 \end{bmatrix} \\ &= \text{diag}(f_i) - \mathbf{f}^\top\mathbf{f},\end{aligned}$$

where $\mathbf{f} = [f_1, .., f_n]$ and $\mathbf{f}^\top\mathbf{f}$ is the outer product of $\mathbf{f}$. We can then express the spectral norm of $\mathbf{J}$ as

$$\begin{aligned}\|\mathbf{J}\| &= \|\text{diag}(f_i) - \mathbf{f}^\top\mathbf{f}\| \\ &\leq \|\text{diag}(f_i)\| + \|\mathbf{f}^\top\mathbf{f}\|.\end{aligned} \tag{5}$$

Note that $\text{diag}(f_i)$ and $\mathbf{f}^\top\mathbf{f}$ are both symmetric matrices. The spectral norm of a symmetric matrix $\mathbf{M}$ is the largest absolute value of its eigenvalues $\lambda$:

$$\|\mathbf{M}\| = \max_i |\lambda_i(\mathbf{M})|. \tag{6}$$

For a diagonal matrix like $\text{diag}(f_i)$, its eigenvectors are the standard basis vector while its eigenvalues are the non-zero diagonal entries, i.e., $\lambda_i(\text{diag}(f_i)) = f_i$. Thus, we can get

$$\|\text{diag}(f_i)\| = \max_i f_i. \tag{7}$$

Next, we find $\|\mathbf{f}^\top\mathbf{f}\|$ through the eigenvalues of $\mathbf{f}^\top\mathbf{f}$. When we take the product of $\mathbf{f}^\top\mathbf{f}$ and $\mathbf{f}^\top$,

$$\begin{aligned}\mathbf{f}^\top\mathbf{f}\cdot\mathbf{f}^\top &= \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \\ &= \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \cdot \sum_i {f_i}^2 \\ &= \left(\sum_i {f_i}^2\right)\mathbf{f}^\top.\end{aligned}$$

From this, we know $\lambda_1(\mathbf{f}^\top\mathbf{f}) = \sum_i {f_i}^2$, with the corresponding eigenvector $\mathbf{v}_1 = \mathbf{f}^\top$. Since the remaining $n-1$ eigenvectors are orthogonal to $\mathbf{v}_1$, i.e., $\mathbf{v}_1^\top\mathbf{v}_i = \mathbf{f}\mathbf{v}_i = 0, \forall i \neq 1$, we have

$$\begin{aligned}\mathbf{f}^\top\mathbf{f}\cdot\mathbf{v}_i &= \mathbf{f}^\top(\mathbf{f}\cdot\mathbf{v}_i) \\ &= \mathbf{0}.\end{aligned}$$

This implies that $\sum_i {f_i}^2$ is the only non-zero eigenvalue of $\mathbf{f}^\top\mathbf{f}$. Thus, with (6), this gives

$$\|\mathbf{f}^\top\mathbf{f}\| = \sum_i {f_i}^2.$$

Combining this with (5) and (7), we get

$$\|\mathbf{J}\| \leq \max_i f_i + \sum_i {f_i}^2. \tag{8}$$

Recall that $\|\mathbf{J}\|$ is the largest possible spectral norm of $\mathbf{J}$, i.e., $\|\mathbf{J}\|^* = \max_{\mathbf{y}} \|\mathbf{J}\|$. Moreover, by

definition of probability, it holds that $f_i \leq 1$ and sum of probabilities $\sum f_i \leq 1$. Therefore,

$$\begin{aligned} \|\mathbf{J}\|^* &\leq \max_{i,\mathbf{y}} f_i + \max_{\mathbf{y}} \sum_i {f_i}^2 \\ &\leq 1 + 1 = 2. \end{aligned} \tag{9}$$

With (4) and (9), we get

$$\|f(\mathbf{y}') - f(\mathbf{y})\| \leq 2\|(\mathbf{y}' - \mathbf{y})\|. \tag{10}$$

Bounding $\|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\|$ with (2), (10), and (3), this gives

$$\begin{aligned} \|\mathbf{A}'_{i,*} - \mathbf{A}_{i,*}\| &= \|f(\alpha g(\mathbf{x}')) - f(\alpha g(\mathbf{x}))\| \\ &\leq \|2\alpha g(\mathbf{x}') - 2\alpha g(\mathbf{x})\| \\ &= 2\alpha\|g(\mathbf{x}') - g(\mathbf{x})\| \\ &\leq 2\alpha\|\mathbf{K}\|\|\mathbf{x}' - \mathbf{x}\| \\ &= 2\alpha\|\mathbf{K}\|\epsilon. \end{aligned}$$

□