

The Case for Translation-Invariant Self-Attention in Transformer-Based Language Models

Ulme Wennberg Gustav Eje Henter

Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Sweden
{ulme, ghe}@kth.se

Abstract

Mechanisms for encoding positional information are central for transformer-based language models. In this paper, we analyze the position embeddings of existing language models, finding strong evidence of translation invariance, both for the embeddings themselves and for their effect on self-attention. The degree of translation invariance increases during training and correlates positively with model performance. Our findings lead us to propose translation-invariant self-attention (TISA), which accounts for the relative position between tokens in an interpretable fashion without needing conventional position embeddings. Our proposal has several theoretical advantages over existing position-representation approaches. Experiments show that it improves on regular ALBERT on GLUE tasks, while only adding orders of magnitude less positional parameters.

1 Introduction

The recent introduction of transformer-based language models by Vaswani et al. (2017) has set new benchmarks in language processing tasks such as machine translation (Lample et al., 2018; Gu et al., 2018; Edunov et al., 2018), question answering (Yamada et al., 2020), and information extraction (Wadden et al., 2019; Lin et al., 2020). However, because of the non-sequential and position-independent nature of the internal components of transformers, additional mechanisms are needed to enable models to take word order into account.

Liu et al. (2020) identified three important criteria for ideal position encoding: Approaches should be *inductive*, meaning that they can handle sequences and linguistic dependencies of arbitrary length, *data-driven*, meaning that positional dependencies are learned from data, and *efficient* in terms of the number of trainable parameters. Separately,

Shaw et al. (2018) argued for *translation-invariant* positional dependencies that depend on the relative distances between words rather than their absolute positions in the current text fragment. It is also important that approaches be *parallelizable*, and ideally also *interpretable*. Unfortunately, none of the existing approaches for modeling positional dependencies satisfy all these criteria, as shown in Table 1 and in Sec. 2. This is true even for recent years' state-of-the-art models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), and ELECTRA (Clark et al., 2020), which require many positional parameters but still cannot handle arbitrary-length sequences.

This paper makes two main contributions: First, in Sec. 3, we analyze the learned position embeddings in major transformer-based language models. Second, in Sec. 4, we leverage our findings to propose a new positional-dependence mechanism that satisfies all desiderata enumerated above. Experiments verify that this mechanism can be used alongside conventional position embeddings to improve downstream performance. Our [code is available](#).

2 Background

Transformer-based language models (Vaswani et al., 2017) have significantly improved modeling accuracy over previous state-of-the-art models like ELMo (Peters et al., 2018). However, the non-sequential nature of transformers created a need for other mechanisms to inject positional information into the architecture. This is now an area of active research, which the rest of this section will review.

The original paper by Vaswani et al. (2017) proposed summing each token embedding with a position embedding, and then used the resulting embedding as the input into the first layer of the model. BERT (Devlin et al., 2019) reached improved performance training data-driven d -dimensional em-

Method	Inductive?	Data-driven?	Parameter efficient?	Translation invariant?	Parallel-izable?	Interpretable?
Sinusoidal position embedding (Vaswani et al., 2017)	✓	✗	✓	✗	✓	✗
Absolute position embedding (Devlin et al., 2019)	✗	✓	✗	✗	✓	✗
Relative position embedding (Shaw et al., 2018)	✗	✓	✓	✓	✗	✗
T5 (Raffel et al., 2020)	✗	✓	✓	✓	✓	✓
Flow-based (Liu et al., 2020)	✓	✓	✓	✗	✗	✗
Synthesizer (Tay et al., 2020)	✗	✓	✓	✗	✓	✗
Untied positional scoring (Ke et al., 2021)	✗	✓	✗	✗	✓	✗
Rotary position embedding (Su et al., 2021)	✓	✗	✓	✓	✓	✗
Translation-invariant self-attention (proposed)	✓	✓	✓	✓	✓	✓

Table 1: Characteristics of position-representation approaches for different language-modeling architectures.

beddings for each position in text snippets of at most n tokens. A family of models have tweaked the BERT recipe to improve performance, including RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020), where the latter has layers share the same parameters to achieve a more compact model.

All these recent data-driven approaches are restricted to fixed max sequence lengths of n tokens or less (typically $n = 512$). Longformer (Beltagy et al., 2020) showed modeling improvements by increasing n to 4096, suggesting that the cap on sequence length limits performance. However, the Longformer approach also increased the number of positional parameters 8-fold, as the number of parameters scales linearly with n ; cf. Table 2.

Clark et al. (2019) and Htut et al. (2019) analyzed BERT attention, finding some attention heads to be strongly biased to local context, such as the previous or the next token. Wang and Chen (2020) found that even simple concepts such as word-order and relative distance can be hard to extract from absolute position embeddings. Shaw et al. (2018) independently proposed using relative position embeddings that depend on the signed distance between words instead of their absolute position, making local attention easier to learn. They reached improved BLEU scores in machine translation, but their approach (and refinements by Huang et al. (2019)) are hard to parallelize, which is unattractive in a world driven by parallel computing. Zeng et al. (2020) used relative attention in speech synthesis, letting each query interact with separate matrix transformations for each key vector, depending on their relative-distance offset. Raffel et al. (2020) directly model position-to-position interactions, by splitting relative-distance offsets into q bins. These relative-attention approaches all facilitate processing sequences of arbitrary length, but can only resolve linguistic dependencies up to a fixed predefined maximum distance.

Tay et al. (2020) directly predicted both word and position contributions to the attention matrix without depending on token-to-token interactions. However, the approach is not inductive, as the size of the attention matrix is a fixed hyperparameter.

Liu et al. (2020) used sinusoidal functions with learnable parameters as position embeddings. They obtain compact yet flexible models, but use a neural ODE, which is computationally unappealing.

Ke et al. (2021) showed that self-attention works better if word and position embeddings are untied to reside in separate vector spaces, but their proposal is neither inductive nor parameter-efficient.

Su et al. (2021) propose rotating each embedding in the self-attention mechanism based on its absolute position, thereby inducing translational invariance, as the inner product of two vectors is conserved under rotations of the coordinate system. These rotations are, however, not learned.

The different position-representation approaches are summarized in Table 1. None of them satisfy all design criteria. In this article, we analyze the position embeddings in transformer models, leading us to propose a new positional-scoring mechanism that combines all desirable properties (final row).

3 Analysis of Existing Language Models

In this section, we introspect selected high-profile language models to gain insight into how they have learned to account for the effect of position.

3.1 Analysis of Learned Position Embeddings

First, we stack the position embeddings in the matrix $E_P \in \mathbb{R}^{n \times d}$, and inspect the symmetric matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$, where $P_{i,j}$ represents the inner product between the i th and j th embedding vectors. If inner products are translation invariant, $P_{i,j}$ will only depend on the difference between the indices, $j - i$, giving a *Toeplitz matrix*, a matrix where each diagonal is constant.

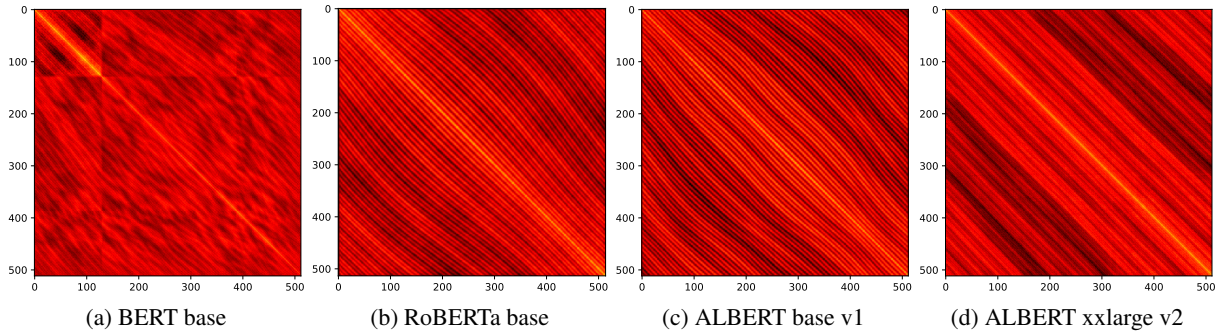


Figure 1: Heatmaps visualizing the matrix $P = E_P E_P^T$ of position-embedding inner products for different models. The greater the inner product between the embeddings, the brighter the color. See appendix Figs. 4, 5 for more.

Fig. 1 visualizes the P -matrices for the position embeddings in a number of prominent transformer models, listed from oldest to newest, which also is in order of increasing performance. We note that a clear Toeplitz structure emerges from left to right. Translation invariance is also seen when plotting position-embedding cosine similarities, as done by Wang and Chen (2020) for transformer-based language models and by Dosovitskiy et al. (2020) for 2D transformers modeling image data.

In Fig. 2 we further study how the degree of Toeplitzness (quantified by R^2 , the amount of the variance among matrix elements $P_{i,j}$ explained by the best-fitting Toeplitz matrix) changes for different ALBERT models. With longer training time (i.e., going from ALBERT v1 to v2), Toeplitzness increases, as the arrows show. This is associated with improved mean dev-set score. Such evolution is also observed in Wang and Chen (2020, Fig. 8).

3.2 Translation Invariance in Self-Attention

Next, we analyze how this translation invariance is reflected in self-attention. Recall that Vaswani et al. (2017) self-attention can be written as

$$\text{att}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1)$$

and define position embeddings E_P , word embeddings E_W , and query and key transformation weight matrices W_Q and W_K . By taking

$$QK^T = (E_W + E_P)W_QW_K^T(E_W + E_P)^T \quad (2)$$

and replacing each row of E_W by the average word embedding across the entire vocabulary, we obtain a matrix we call \hat{F}_P that quantifies the average effect of E_P on the softmax in Eq. (1). Plots of the resulting \hat{F}_P for all 12 ALBERT-base attention heads in the first layer are in appendix Fig. 8. Importantly, these matrices also exhibit Toeplitz structure. Fig. 3 graphs sections through the main diagonal for

selected heads, showing peaks at short relative distances, echoing Clark et al. (2019) and Htut et al. (2019). In summary, we conclude that position encodings, and their effect on softmax attention, have an approximately translation-invariant structure in successful transformer-based language models.

4 Proposed Self-Attention Mechanism

We now introduce our proposal for parameterizing the positional contribution to self-attention in an efficient and translation-invariant manner, optionally removing the position embeddings entirely.

4.1 Leveraging Translation Invariance for Improved Inductive Bias

Our starting point is the derivation of Ke et al. (2021). They expand QK^T while ignoring cross terms, yielding

$$QK^T \approx E_WW_QW_K^TE_W^T + E_PW_QW_K^TE_P^T, \quad (3)$$

an approximation they support by theory and empirical evidence. They then “untie” the effects of words and positions by using different W -matrices for the two terms in Eq. (3). We agree with sepa-

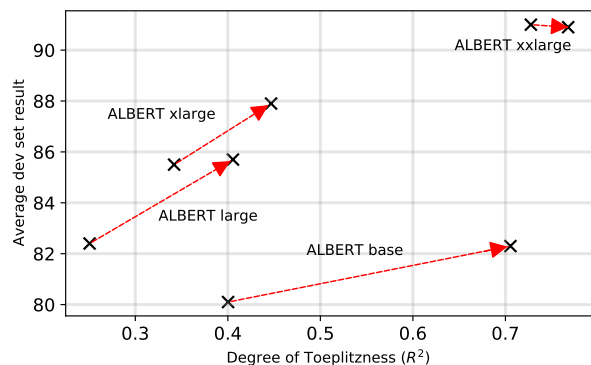


Figure 2: Scatterplot of the degree of Toeplitzness of P for different ALBERT models (v1→v2) against average performance numbers (from Lan et al.’s GitHub) over SST-2, MNLI, RACE, and SQuAD 1.1 and 2.0.

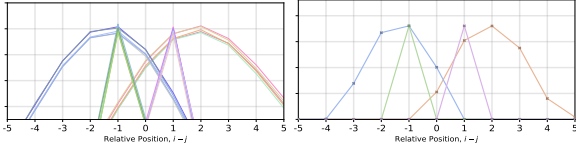


Figure 3: Positional responses of select attention heads. Left: Sections $(\widehat{F}_P)_{i,j}$ through \widehat{F}_P of ALBERT base v2, varying j for 5 different i , keeping $j = i$ centered. The sections are similar regardless of i since \widehat{F}_P is close to Toeplitz. Colors distinguish different heads. Right: TISA scoring functions, attending to similar positions as heads on the left. Larger plots in Figs. 6, 7.

rating these effects, but also see a chance to reduce the number of parameters.

Concretely, we propose to add a second term $F_P \in \mathbb{R}^{n \times n}$, a Toeplitz matrix, inside the parentheses of Eq. (1). F_P can either a) supplement or b) replace the effect of position embeddings on attention in our proposed model. For case a), we simply add F_P to the existing expression inside the softmax, while for case b) a term $\sqrt{d_k} F_P$ is inserted in place of the term $E_P W_Q W_K^T E_P^T$ in Eq. (3). This produces two new self-attention equations:

$$\text{att} = \begin{cases} \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} + F_P \right) V & \text{a)} \\ \text{softmax} \left(\frac{Q_W K_W^T}{\sqrt{d_k}} + F_P \right) V_W & \text{b)} \end{cases} \quad (4)$$

where the inputs Q_W , K_W , and V_W (defined by $Q_W = E_W W_Q$, and similarly for K_W and V_W) do not depend on the position embeddings E_P . Case a) is not as interpretable as TISA alone (case b), since the resulting models have two terms, E_P and F_P , that share the task of modeling positional information. Our two proposals apply to any sequence model with a self-attention that follows Eq. (1), where the criteria in Table 1 are desirable.

4.2 Positional Scoring Function

Next, we propose to parameterize the Toeplitz matrix F_P using a *positional scoring function* $f_\theta(\cdot)$ on the integers \mathbb{Z} , such that $(F_P)_{i,j} = f_\theta(j - i)$. f_θ defines F_P -matrices of any size n . The value of $f_\theta(j - i)$ directly models the positional contribution for how the token at position i attends to position j . We call this *translation-invariant self-attention*, or TISA. TISA is inductive and can be simplified down to arbitrarily few trainable parameters.

Let $k = j - i$. Based on our findings for \widehat{F}_P in Sec. 3, we seek a parametric family $\{f_\theta\}$ that allows both localized and global attention, without diverging as $|k| \rightarrow \infty$. We here study one family

	Standard	Ke et al. (2021)	TISA
General formula	nd	$nd + 2d^2$	$3SHL$
Longformer	3,145,728	4,325,376	2,160
BERT/roBERTa	393,216	1,572,864	2,160
ALBERT	65,536	98,304	2,160

Table 2: Number of positional parameters for base models of different language-model architectures and different positional information processing methods, with max sequence length $n \in (512, 4096)$, position embeddings of dimension $d \in (128, 768)$, $S = 5$ kernels, $H = 12$ attention heads, and $L = 12$ layers with distinct TISA positional scoring functions. Parameter sharing gives ALBERT lower numbers. TISA can be used alone or added to the counts in other columns.

that satisfies the criteria: the radial-basis functions

$$f_\theta(k) = \sum_{s=1}^S a_s \exp\left(-|b_s|(k - c_s)^2\right). \quad (5)$$

Their trainable parameters are $\theta = \{a_s, b_s, c_s\}_{s=1}^S$, i.e., 3 trainable parameters per kernel s . Since these kernels are continuous functions (in contrast to the discrete bins of Raffel et al. (2020)), predictions change smoothly with distance, which seems intuitively meaningful for good generalization.

Lin et al. (2019) found that word-order information in BERT’s position embeddings gets increasingly washed out from layer 4 onward. As suggested by Dehghani et al. (2019) and Lan et al. (2020), we inject positional information into each of the H heads at all L layers, resulting in one learned function $f_{\theta(h,l)}$ for each head and layer. The total number of positional parameters of TISA is then $3SHL$. As seen in Table 2, this is several orders of magnitude less than the embeddings in prominent language models.

The inductivity and localized nature of TISA suggests the possibility to rapidly pre-train models on shorter text excerpts (small n), scaling up to longer n later in training and/or at application time, similar to the two-stage training scheme used by Devlin et al. (2019), but without risking the under-training artifacts visible for BERT at $n > 128$ in Figs. 1 and 4. However, we have not conducted any experiments on the performance of this option.

5 Experiments

The main goal of our experiments is to illustrate that TISA can be added to models to improve their performance (Table 3a), while adding a minuscule amount of extra parameters. We also investigate the performance of models without position em-

Task	Baseline	$S=1$	3	5	Δ	$\Delta\%$
SST-2	92.9	93.3	93.1	93.1	0.4	6.5%
MNLI	83.8	84.1	84.4	84.8	1.0	5.9%
QQP	88.2	88.0	88.3	88.3	0.1	1.2%
STS-B	90.3	90.4	90.0	90.4	0.1	1.5%
CoLA	57.2	57.0	56.5	58.5	1.3	2.9%
MRPC	89.6	90.1	89.0	90.1	0.5	5.3%
QNLI	91.6	91.7	91.4	91.6	0.1	0.4%
RTE	72.9	71.1	73.6	73.6	0.7	2.7%

(a) ALBERT base v2 models with position embeddings

Task	Baseline	$S=1$	3	5	Δ	$\Delta\%$
SST-2	85.1	85.9	85.8	86.0	0.9	6.2%
MNLI	78.8	80.9	81.4	81.6	2.8	13.4%
QQP	86.3	86.2	86.5	86.8	0.5	3.4%
STS-B	89.0	89.0	89.1	89.1	0.1	0.3%
MRPC	82.8	83.1	83.3	83.1	0.5	3.3%
QNLI	86.6	87.2	87.4	87.7	1.1	7.8%
RTE	62.1	61.7	62.5	62.8	0.7	1.9%

(b) ALBERT base v2 models without position embeddings

Table 3: GLUE task dev-set performance (median over 5 runs) with TISA (S kernels) and without (baseline). Δ is the maximum performance increase in a row and $\Delta\%$ is the corresponding relative error reduction rate.

beddings (Table 3b), comparing TISA to a bag-of-words baseline ($S = 0$). All experiments use pretrained ALBERT base v2 implemented in Huggingface (Wolf et al., 2020). Kernel parameters $\theta^{(h)}$ for the functions in Eq. (5) were initialized by regression to the \hat{F}_P profiles of the pretrained model, (see Appendix C for details); example plots of resulting scoring functions are provided in Fig. 3. We then benchmark each configuration with and without TISA for 5 runs on GLUE tasks (Wang et al., 2018), using jiant (Phang et al., 2020) and standard dataset splits to evaluate performance.

Our results in Table 3a show relative error reductions between 0.4 and 6.5% when combining TISA and conventional position embeddings. These gains are relatively stable regardless of S . We also note that Lan et al. (2020) report 92.9 on SST-2 and 84.6 on MNLI, meaning that our contribution leads to between 1.3 and 2.8% relative error reductions over their scores. The best performing architecture ($S=5$), gives improvements over the baseline on 7 of the 8 tasks considered and on average increases the median F1 score by 0.4 points. All these gains have been realized using a very small number of added parameters, and without pre-training on any data after adding TISA to the architecture. The only joint training happens on the training data of each particular GLUE task.

Results for TISA alone, in Table 3b, are not as

strong. This could be because these models are derived from an ALBERT model pretrained using conventional position embeddings, since we did not have the computational resources to tune from-scratch pretraining of TISA-only language models.

Figs. 3 and 6 plot scoring functions of different attention heads from the initialization described in Appendix C. Similar patterns arose consistently and rapidly in preliminary experiments on pretraining TISA-only models from scratch. The plots show heads specializing in different linguistic aspects, such as the previous or next token, or multiple tokens to either side, with other heads showing little or no positional dependence. This mirrors the visualizations of ALBERT base attention heads in Figs. 3, 6, 7, 8 and the findings of Htut et al. (2019) and Clark et al. (2019) on BERT, but TISA makes this directly visible in an interpretable model, without having to probe correlations in a black box.

Interestingly, the ALBERT baseline on STS-B in Table 3a is only 1.3 points ahead of the bag-of-words baseline in Table 3b. This agrees with experiments shuffling the order of words (Pham et al., 2020; Sinha et al., 2021) finding that modern language models tend to focus mainly on higher-order word co-occurrences, rather than word order, and suggests that word-order information is underutilized in state-of-the-art language models.

6 Conclusion

We have analyzed state-of-the-art transformer-based language models, finding that translation-invariant behavior emerges during training. Based on this we proposed TISA, the first positional information processing method to simultaneously satisfy the six key design criteria in Table 1. Experiments demonstrate competitive downstream performance. The method is applicable also to transformer models outside language modeling, such as modeling time series in speech or motion synthesis, or to describe dependencies between pixels in computer vision.

Acknowledgments

We would like to thank Gabriel Skantze, Dmytro Kalpakchi, Viktor Karlsson, Filip Cornell, Oliver Åstrand, and the anonymous reviewers for their constructive feedback. This research was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#).
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? An analysis of BERT’s attention](#). In *Proc. BlackboxNLP@ACL*, pages 276–286.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Pre-training transformers as energy-based cloze models](#). In *Proc. EMNLP*, pages 285–294.
- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2019. [Universal transformers](#). In *Proc. ICLR*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proc. NAACL-HLT*, pages 4171–4186.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. [An image is worth 16x16 words: Transformers for image recognition at scale](#). *CoRR*, abs/2010.11929.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proc. EMNLP*, pages 489–500.
- Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor O. K. Li. 2018. [Meta-learning for low-resource neural machine translation](#). In *Proc. EMNLP*, pages 3622–3631.
- Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. [Do attention heads in BERT track syntactic dependencies?](#)
- Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinulescu, and Douglas Eck. 2019. [Music transformer](#). In *Proc. ICLR*.
- Guolin Ke, Di He, and Tie-Yan Liu. 2021. [Rethinking positional encoding in language pre-training](#). In *Proc. ICLR*.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. [Phrase-based & neural unsupervised machine translation](#). In *Proc. EMNLP*, pages 5039–5049.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *Proc. ICLR*.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. [A joint neural model for information extraction with global features](#). In *Proc. ACL*, pages 7999–8009.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. [Open sesame: Getting inside BERT’s linguistic knowledge](#). In *Proc. BlackboxNLP@ACL*, pages 241–253.
- Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. [Learning to encode position for transformer with continuous dynamical model](#). In *Proc. ICML*, pages 6327–6335.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#).
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proc. NAACL*, pages 2227–2237.
- Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. [Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?](#)
- Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. [jiant 2.0: A software toolkit for research on general-purpose text understanding models](#). <http://jiant.info/>.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *JMLR*, 21(140):1–67.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. [Self-attention with relative position representations](#). In *Proc. NAACL-HLT*, pages 464–468.
- Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. [Masked language modeling and the distributional hypothesis: Order word matters pre-training for little](#).
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. [Roformer: Enhanced transformer with rotary position embedding](#). *CoRR*, abs/2104.09864.
- Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2020. [Synthesizer: Rethinking self-attention in transformer models](#).
- 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 *Proc. NIPS*, pages 5998–6008.

- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proc. EMNLP-IJCNLP*, pages 5784–5789.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proc. BlackboxNLP@EMNLP*, pages 353–355.
- Yu-An Wang and Yun-Nung Chen. 2020. [What do position embeddings learn? An empirical study of pre-trained language model positional encoding](#). In *Proc. EMNLP*, pages 6840–6849.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proc. EMNLP System Demonstrations*, pages 38–45.
- Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. [LUKE: Deep contextualized entity representations with entity-aware self-attention](#). In *Proc. EMNLP*, pages 6442–6454.
- Zhen Zeng, Jianzong Wang, Ning Cheng, and Jing Xiao. 2020. [Prosody learning mechanism for speech synthesis system without text length limit](#). In *Proc. Interspeech*, pages 4422–4426.

A Visualizing $E_P E_P^T$ for Additional Language Models

Fig. 1 shows the inner product between different position embeddings for the models BERT base uncased, RoBERTa base, ALBERT base v1 as well as ALBERT xxlarge v2. Leveraging our analysis findings of translation invariance in the matrix of $E_P E_P^T$ in these pretrained networks, we investigate the generality of this phenomenon by visualizing the same matrix for additional existing large language models. We find that similar Toeplitz patterns emerge for all investigated networks.

B Coefficient of Determination R^2

The coefficient of determination, R^2 , is a widely used concept in statistics that measures what fraction of the variance in a dependent variable that can be explained by an independent variable. Denoting the Residual Sum of Squares, RSS , and Total Sum of Squares, TSS , we have that

$$R^2 = 1 - \frac{RSS}{TSS}, \quad (6)$$

where $R^2 = 0$ means that the dependent variable is not at all explained, and $R^2 = 1$ means that the variance is fully explained by the independent variable.

Applied to a matrix, $A \in \mathbb{R}^{n \times n}$, to determine its degree of Toeplitzness, we get RSS by finding the Toeplitz matrix, $A_T \in \mathbb{R}^{n \times n}$, that minimizes the following expression:

$$RSS = \min_{A_T} \sum_{i=1}^n \sum_{j=1}^n (A - A_T)_{i,j}^2 \quad (7)$$

Furthermore, we can compute TSS as:

$$TSS = \sum_{i=1}^n \sum_{j=1}^n \left(A_{i,j} - \left(\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n A_{i,j} \right) \right)^2 \quad (8)$$

C Extracting ALBERT positional scores

In order to extract out the positional contributions to the attention scores from ALBERT, we disentangle the positional and word-content contributions from equation (3), and remove any dependencies on the text sequence through E_W . We exchange $E_W \approx E_{\overline{W}}$, with the average word embedding

over the *entire vocabulary*, which we call $E_{\overline{W}}$.

$$F_P \approx \frac{1}{\sqrt{d_k}} (E_W W_Q W_K^T E_P^T + \quad (9)$$

$$+ E_P W_Q W_K^T E_W^T + E_P W_Q W_K^T E_P^T) \quad (10)$$

$$\approx \frac{1}{\sqrt{d_k}} (E_{\overline{W}} W_Q W_K^T E_P^T + \quad (11)$$

$$+ E_P W_Q W_K^T E_{\overline{W}}^T + E_P W_Q W_K^T E_P^T) \quad (12)$$

This way, we can disentangle and extract the positional contributions from the ALBERT model.

Initialization of Position-Aware Self-Attention

Using this trick, we initialize F_P with formula (12). Since F_P is only generating the positional scores, which are independent of context, it allows for training a separate positional scorer neural network to predict the positional contributions in the ALBERT model. Updating only 2,160 parameters (see Table 2) significantly reduces the computational load. This pretraining initialization scheme converges in less than 20 seconds on a CPU.

Removing Position Embeddings When removing the effect of position embeddings, we calculate the average position embedding and exchange all position embeddings for it. This reduces the variation between position embeddings, while conserving the average value of the original input vectors $E_W + E_P$.

Extracted Attention Score Contributions

Leveraging our analysis findings of translation invariance in large language models, we visualize the scoring functions as a function of relative distance offset between tokens. Fig. 3 shows the implied scoring functions for 4 attention heads for 5 different absolute positions. Figs. 6, 7 show all 12 attention heads of ALBERT base v2 with TISA.

D Number of Positional Parameters of Language Models

In the paper, define positional parameters as those modeling only positional dependencies. In most BERT-like models, these are the position embeddings only (typically $n \times d$ parameters). Ke et al. (2021) propose to separate position and content embeddings, yielding more expressive models with separate parts of the network for processing separate information sources. In doing so, they introduce two weight matrices specific to positional information processing, $U_Q \in \mathbb{R}^{d \times d}$ and $U_K \in \mathbb{R}^{d \times d}$, totaling $nd + 2d^2$ positional parameters.

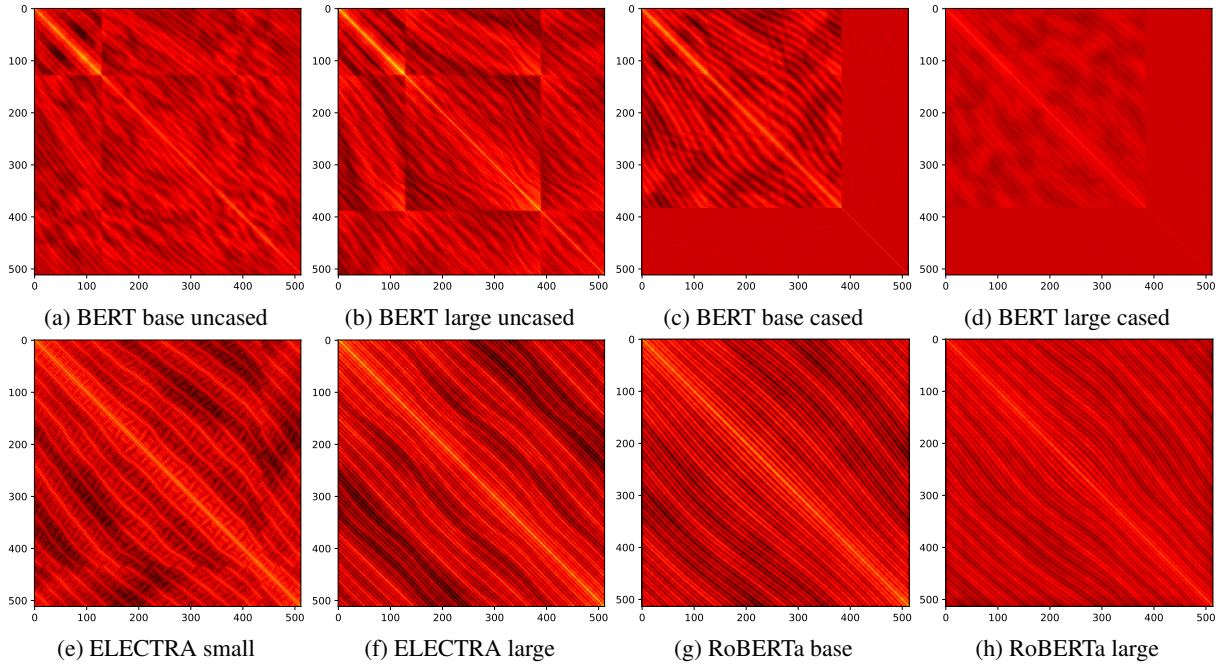


Figure 4: Visualizations of the inner-product matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$ for different BERT, ELECTRA, and RoBERTa models. We see that ELECTRA and RoBERTa models show much stronger signs of translational invariance than their BERT counterparts. Most BERT models follow the pattern noted by Wang and Chen (2020), where the Toeplitz structure is much more pronounced for the first 128×128 submatrix, reflecting how these models mostly were trained on 128-token sequences, and only scaled up to $n = 512$ for the last 10% of training (Devlin et al., 2019). Position embeddings 385 through 512 of the BERT cased models show a uniform color, suggesting that these embeddings are almost completely untrained.

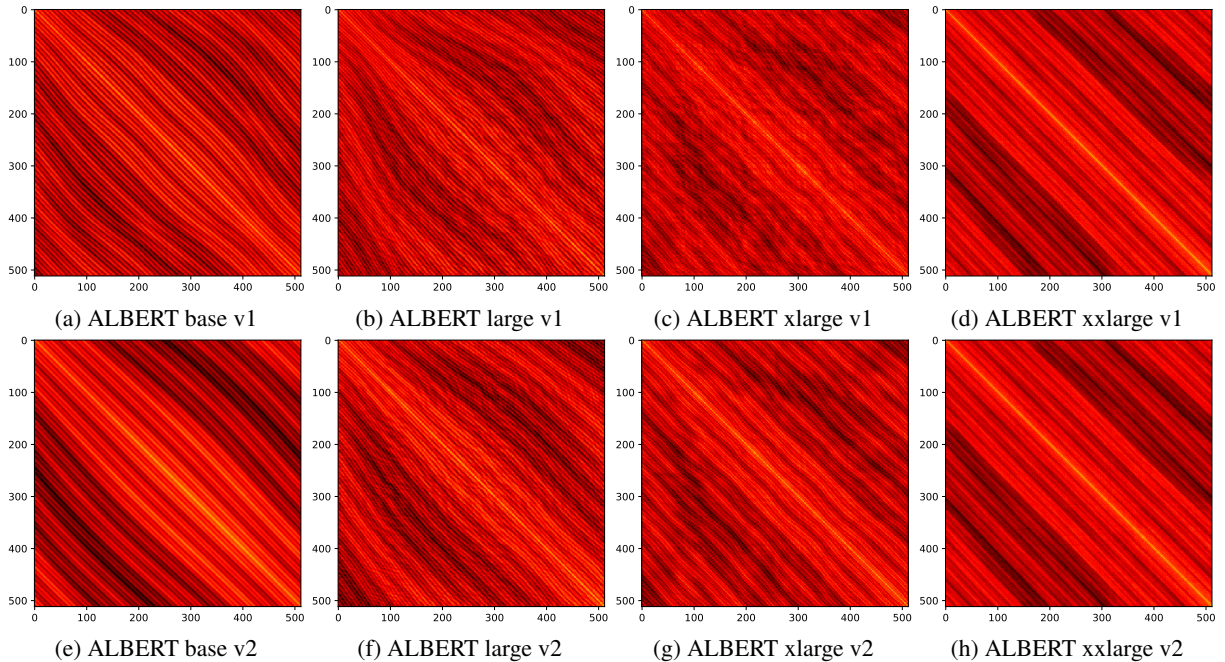


Figure 5: Visualizations of the inner-product matrix $P = E_P E_P^T \in \mathbb{R}^{n \times n}$ for different ALBERT models (Lan et al., 2020). We plot both v1 and v2 to show the progression towards increased Toeplitzness during training.

Hyperparameter Selection We performed a manual hyperparameter search starting from the hyperparameters that the Lan et al. (2020) re-

port in https://github.com/google-research/albert/blob/master/run_glue.sh. Our hyperparameter config files can be found with our code.

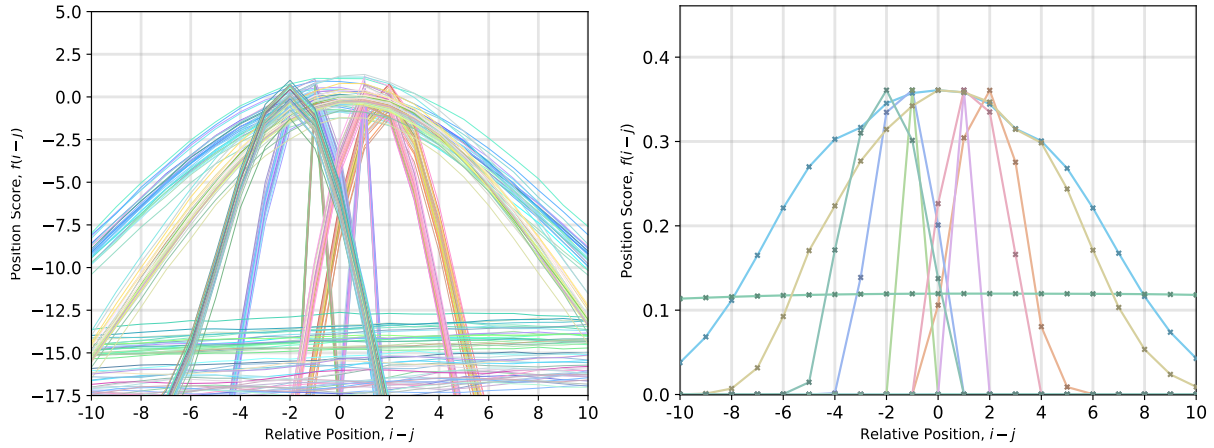


Figure 6: Positional responses of all attention heads. Sections through \hat{F}_P of ALBERT base v2, aligned to the main diagonal, (left) show similar profiles as the corresponding TISA scoring functions (right). Vertical axes differ due to 1) the scaling factor $\sqrt{d_k}$ and 2) softmax being invariant to vertical offset.

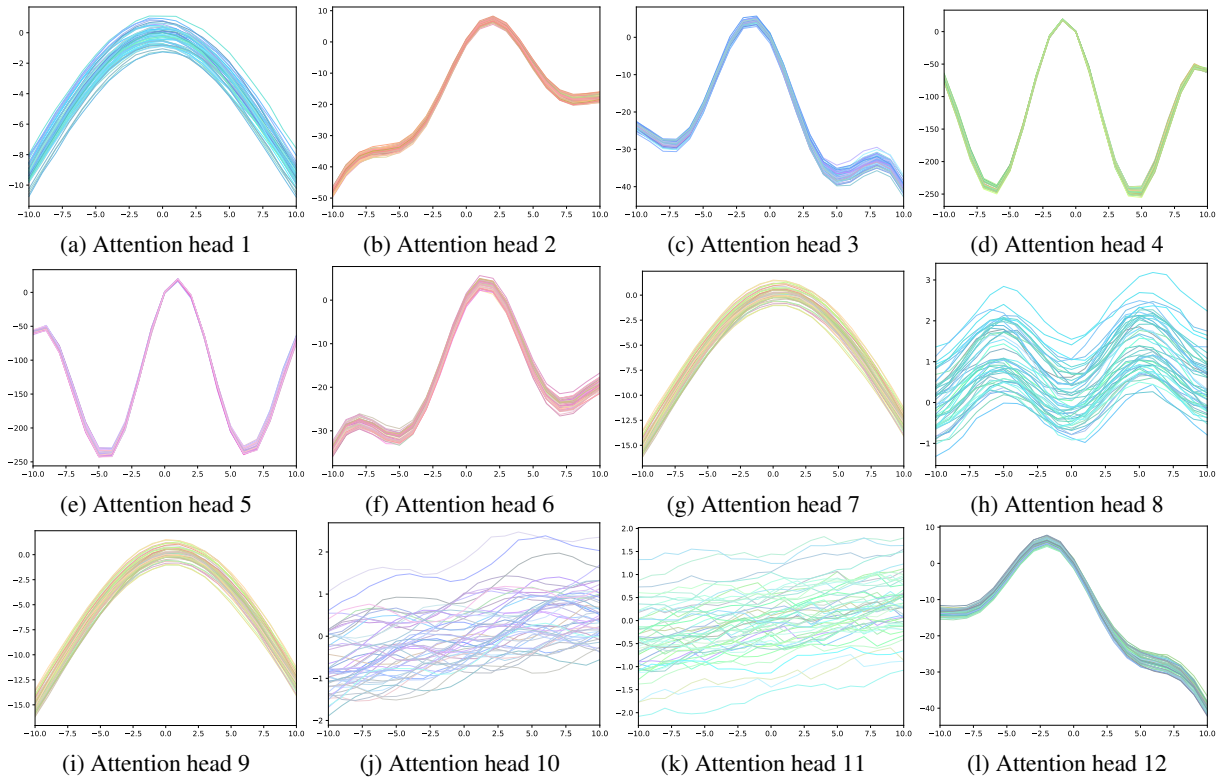


Figure 7: Rows from the positional attention matrices \hat{F}_P for all ALBERT base v2 attention heads, centered on the main diagonal. Note that the vertical scale generally differs between plots. The plots are essentially aligned sections through the matrices in Fig. 8, but zoomed in to show details over short relative distances since this is where the main peak(s) are located, and the highest values are by far the most influential on softmax attention.

E Reproducibility

Experiments were run on a GeForce RTX 2080 machine with 8 GPU-cores. Each downstream experiment took about 2 hours to run.

Datasets and code can be downloaded from https://github.com/nyu-ml1/jiant/blob/master/guides/tasks/supported_tasks.md and <https://github.com/ulmewennberg/tisa>.

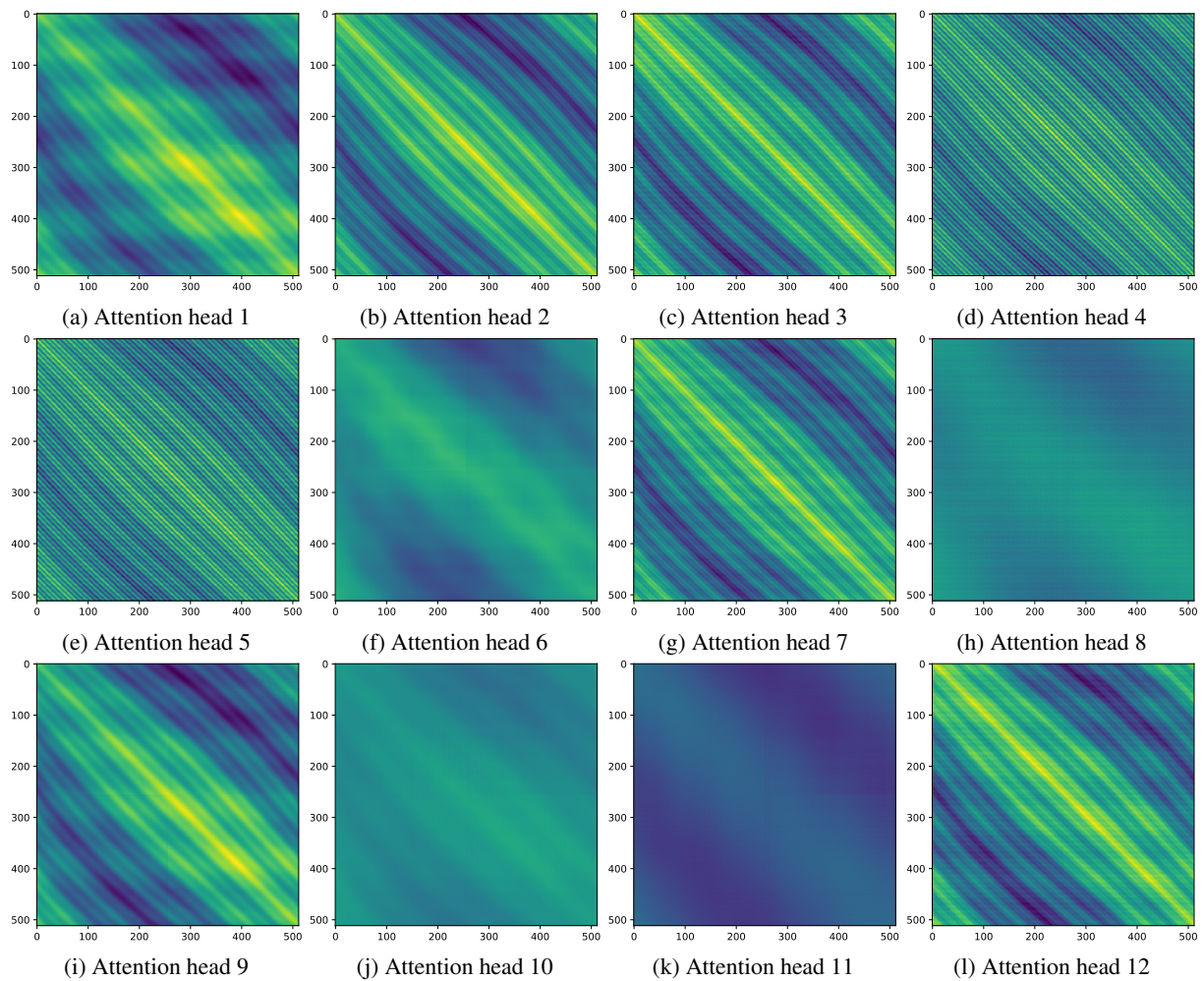


Figure 8: Values extracted from the positional attention matrices for all ALBERT base v2 first-layer attention heads. Some heads are seen to be sensitive to position, while others are not. Note that these visualizations deliberately use a different color scheme from other (red) matrices, to emphasize the fact that the matrices visualized here represent a different phenomenon and are not inner products.