Why Are Positional Encodings Nonessential for Autoregressive Transformers? A Clarification from "Prehistory"

Anonymous ACL submission

Abstract

Do autoregressive Transformer language models *require* explicit positional encodings (PEs)? The answer is *'no'*; as long as they have more than one layer, they can distinguish sequences with permuted tokens without requiring explicit PEs. This property has been known since early efforts adopting the Transformer from machine translation to language modeling, concurrently with GPT-2. However, this result was not widely disseminated and was rediscovered by recent work. This may be partially due to a sudden growth of the language modeling community *after* the advent of GPT-2, but perhaps also because no clear explanation has been provided in prior work. Here we present a comprehensible explanation why explicit PEs are nonessential for *multi-layer* autoregressive Transformers (in contrast, *one-layer* models require positional encodings to discern order information of their input tokens). We review the origin of this result, and hope to establish it as a common knowledge.

1 Introduction

The field of language modeling has seen new waves of interest after the promising results of GPT-2 (Radford et al., 2019), impressive capabilities of GPT-3 (Brown et al., 2020), and unprecedented versatility of ChatGPT and GPT-4 (Bubeck et al., 2023; Achiam et al., 2023), manipulating human languages in a way no machine has ever before.

About a decade before this "GPT-era," [1] neural language modeling research had also seen a smaller but significant growth after Tomáš Mikolov's breakthrough results with recurrent neural network language models (Mikolov et al., 2010, 2011). This had made neural language modeling a popular research topic, particularly among speech recognition and machine translation researchers as these were two major application areas at the time. Note that language models were not a standalone system as is the case today; they were merely a component in a larger system with a specific application (Jelinek et al., 1975; Brown et al., 1988) except in certain visionary work (Sutskever et al., 2011).

When the Transformer encoder-decoder architecture was shown to be successful for machine translation (Vaswani et al., 2017), several works investigated its application to build conventional (autoregressive) language models using the decoder, e.g., Liu et al. (2018); Radford et al. (2018); Al-Rfou et al. (2019); Dai et al. (2019); Baevski and Auli (2019), or non-autoregressive models of language using the encoder, e.g., Devlin et al. (2019); producing many methods and practical knowledge for optimizing Transformers to language modeling, concurrently to GPT-2 (Radford et al., 2019).

While the growth of interest in language modeling in the GPT-era has been very exciting for the field, this sudden growth also marked some forms of discontinuity in the language modeling research. For example, fine-tuning or prompt engineering of large pre-trained models became a mainstream type of research, while research on the model architecture became harder due to large model scales at which experiments are expected to be reported.

We also observe that certain common knowledge from pre-GPT language modeling research does not seem to have been well transferred through this rapid growth, leading to its rediscoveries.

This short review focuses on one of such results, namely the property that *explicit positional encodings are nonessential for multi-layer autoregressive Transformer language models*. We review a simple

[1] Here by "GPT-era" we refer to the time after GPT-2/3 (and analogously, "pre-GPT era" for the time before). We consider the advent of GPT-2/3 as the transition point instead of GPT-1 because the treatment of language models in the GPT-1 paradigm was different (Radford et al., 2018).

"Prehistory" in the title is a hyperbole that carries a double meaning: firstly emphasizing that ideas from just a few years ago are now largely unfamiliar to many working on LMs today; and secondly, more specifically for the positional encoding result discussed here, the explanation we describe, considered trivial by the original authors, has not been *written* before.

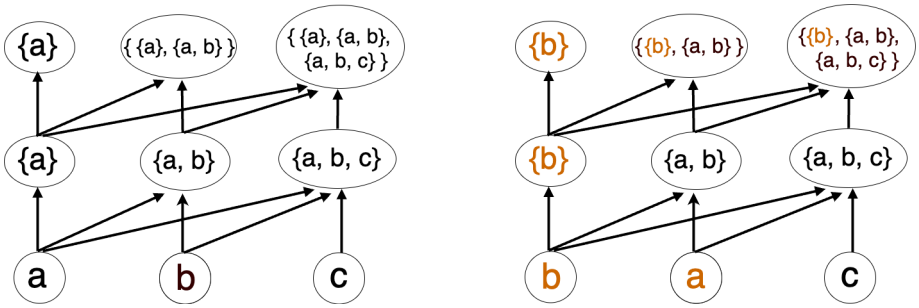

Figure 1: An illustration of autoregressive Transformers without explicit positional encodings for two input sequences: (a, b, c) and its permutation (b, a, c). The color in the right diagram highlights the differences in terms of context seen by each layer at each position (expressed as a set). With two (or more) layers, as soon as an input is different at one position (here at the first position), autoregressive Transformers see different contexts for all later positions at the top layer. In contrast, *one-layer* models can not distinguish between these two input sequences (in this example, as soon as the second token is fed) even though they are strictly speaking not permutation invariant (since the context at the first position is different). Here we only swap the two first tokens; with a more "complex" permutation, they may see different context at more positions but they systematically fail to do so in the last step.

but poorly disseminated explanation of this result, while tracing back to original work that demonstrated this property.

2 Background: Self-Attention

Following the original definition (Vaswani et al., 2017), one Transformer "layer" consists of two sub-layers: a self-attention layer and a feedforward block. Given that a typical feedforward block processes information at each position/time step exclusively, the self-attention layer is the only sequence processing component of the Transformer layer.

Autoregressive Self-Attention. Let d and T denote positive integers. An autoregressive self-attention layer transforms an input sequence $\{\boldsymbol{x}_t\}_{t=1}^T, \boldsymbol{x}_t \in \mathbb{R}^d$ to an output sequence $\{\boldsymbol{y}_t\}_{t=1}^T, \boldsymbol{y}_t \in \mathbb{R}^d$ as follows:

$$\boldsymbol{q}_t, \boldsymbol{k}_t, \boldsymbol{v}_t = \boldsymbol{W}_q\boldsymbol{x}_t, \boldsymbol{W}_k\boldsymbol{x}_t, \boldsymbol{W}_v\boldsymbol{x}_t \quad (1)$$

$$\boldsymbol{K}_t = [\boldsymbol{K}_{t-1}, \boldsymbol{k}_t] \in \mathbb{R}^{d\times t} \quad (2)$$

$$\boldsymbol{V}_t = [\boldsymbol{V}_{t-1}, \boldsymbol{v}_t] \in \mathbb{R}^{d\times t} \quad (3)$$

$$\boldsymbol{y}_t = \boldsymbol{V}_t \mathrm{softmax}(\boldsymbol{K}_t^\top \boldsymbol{q}_t) \quad (4)$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d\times d}$ are trainable weight matrices, $[\boldsymbol{A}, \boldsymbol{a}]$ denotes the concatenation of vector $\boldsymbol{a}$ to matrix $\boldsymbol{A}$ which increments the time dimension ($\boldsymbol{K}_0$ and $\boldsymbol{V}_0$ are initially empty), and $\mathrm{softmax}$ is along the time dimension. We omit the $1/\sqrt{d}$ scaling inside $\mathrm{softmax}$, as well as the output projection, which are irrelevant for our discussion.

While the equations above accurately describe the model conceptually, self-attention is also often expressed in the following *matrix form* that better reflects the possibility to parallelize computation over the time axis during training. By denoting the input as $\boldsymbol{X} = [x_1, ..., x_T] \in \mathbb{R}^{d\times T}$ ($\boldsymbol{X}_i = x_i$) and the output as $\boldsymbol{Y} = [y_1, ..., y_T] \in \mathbb{R}^{d\times T}$, it yields:

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = \boldsymbol{W}_q\boldsymbol{X}, \boldsymbol{W}_k\boldsymbol{X}, \boldsymbol{W}_v\boldsymbol{X} \quad (5)$$

$$\boldsymbol{Y} = \boldsymbol{V}\mathrm{softmax}(\boldsymbol{M} \odot (\boldsymbol{K}^\top\boldsymbol{Q})) \quad (6)$$

where $\boldsymbol{M} \in \mathbb{R}^{T\times T}$ is the so-called *attention mask*. For these equations to be equivalent to Eqs. 1-4 above, i.e., for autoregressive self-attention, $\boldsymbol{M}$ is set to be the upper triangular matrix, i.e., $\boldsymbol{M}_{i,j} = 1$ if $i \leq j$ and $\boldsymbol{M}_{i,j} = -\infty$ otherwise.

We also denote by $\boldsymbol{Y} = \mathrm{SelfAttn}(\boldsymbol{X}, \boldsymbol{M})$ the overall self-attention operation given input $\boldsymbol{X}$ and mask $\boldsymbol{M}$, grouping Eqs. 5-6.

Non-Autoregressive Self-Attention. The same equations (Eqs. 5-6) can also express *non-autoregressive* self-attention by removing the mask $\boldsymbol{M}$, i.e., by setting $\boldsymbol{M}_{i,j} = 1$ for all $i, j \in \{1, ..., T\}$. We denote such $\boldsymbol{M}$ as $\boldsymbol{M} = \mathbb{1}$.

Positional Encodings. When positional encodings (Gehring et al., 2017; Vaswani et al., 2017) are used, a vector representing discrete position t is added to input token $\boldsymbol{x}_t$. The exact choice of PE design is irrelevant to our discussion.

3 Main Results on Positional Encodings

The goal of this short review paper is to provide a summary of results on the necessity/needlessness of positional encodings for Transformers with comprehensible explanations; and to discuss the original references (Sec. 5).

Definitions. We first define two key properties:

(1) A sequence processor $f : \mathbb{R}^{d\times T} \to \mathbb{R}^{d\times T}$ is said to be *permutation invariant* when for any input $\boldsymbol{X} \in \mathbb{R}^{d\times T}$, and its arbitrary permutation along the time/token axis $\boldsymbol{X'} \in \mathbb{R}^{d\times T}$, $f(\boldsymbol{X}) = f(\boldsymbol{X'})$.

(2) f is *fully position-sensitive* when for any inputs $\boldsymbol{X}, \boldsymbol{X'} \in \mathbb{R}^{d\times T}$, if $\boldsymbol{X}_i \neq \boldsymbol{X}'_i$, then $f(\boldsymbol{X})_j \neq f(\boldsymbol{X'})_j$ for all $j \in \{i, ..., T\}$; meaning that as soon as one input is different at position i, f produces different outputs at all the "future" positions $j \geq i$.

Necessity vs. needlessness of explicit PEs is tied to the model's capability to distinguish between permutated sequences[2], which can be characterized using the definitions above. (1) If a sequence processor f is *permutation invariant*, positional encodings are needed. (2) If f is *fully position-sensitive*, positional encodings are not needed. (3, Remark) Not *permutation invariant* is not enough to conclude on the necessity of positional encodings (as all positions matters).

We present the main results in the form of question/answer pairs as follows.

Question 1 (Back to basics)**.** *Why are positional encodings **needed** for **non-autoregressive** Transformers?*

This is because non-autoregressive self-attention is *permutation invariant*, i.e., for any input $\boldsymbol{X}$, and its arbitrary permutation along the time/token axis $\boldsymbol{X'}$, $\mathrm{SelfAttn}(\boldsymbol{X}, \mathbb{1}) = \mathrm{SelfAttn}(\boldsymbol{X'}, \mathbb{1})$.

We can straightforwardly check this by directly looking at Eqs. 5-6. Without the mask, i.e., $\boldsymbol{M} = \mathbb{1}$, keys and queries from all positions interact regardless of their positions to yield attention scores (i.e., $\mathrm{softmax}(\boldsymbol{K}^\top \boldsymbol{Q}) \in \mathbb{R}^{T\times T}$ in Eq. 6), which are used to compute weighted average, which is commutative, of values.

Question 2 (Knowledge Bias)**.** *Why are positional encodings believed to be crucial for "self-attention" by "default" in the first place?*

This is partly because without specification of being autoregressive or non-autoregressive, the default self-attention is often (somehow) assumed to be non-autoregressive (as in the encoder for machine translation (Vaswani et al., 2017)) which requires positional encodings as discussed above.

Secondly, the explanation for the necessity of positional encodings in the original paper was the fact that the "*model contains no recurrence and no convolution*" (Vaswani et al., 2017). As we'll see below, this explanation is incomplete, but if one assumed this to be true, it would imply that the autoregressive self-attention also requires positional encodings (we emphasize that this is not true).

Question 3. *Why are positional encodings **nonessential** for **multi-layer autoregressive** Transformers?*

This is because multi-layer autoregressive Transformers are *fully position-sensitive*.

A simple method to check this is to compare the model outputs when we feed two sequences which are permutations of each other. While we could also provide a mathematical proof here, this can better be visualized as in Figure 1: even for this extreme case where we feed two sequences that only differ from each other by a permutation of their two first tokens, the autoregressive model sees different contexts at all positions at the top layer.

Question 4. *Does the multi-layer autoregressive Transformer language models effectively learn to use positional signals in practice?*

For this question, we refer to Irie et al. (2019) which demonstrated good general performance of multi-layer autoregressive Transformer language models without PEs and provided visualization of attention weights (see figures in Irie et al. (2019)). They report that, interestingly, the first attention layer mainly attends to the new input, while the second layer uniformly attends to the context. Uniform attention in early layers is intuitively good as it allows the model to grasp all the available context, which is crucial to distinguish similar sequences (as illustrated in Figure 1).

Finally, being nonessential does not mean that some sophisticated extra positional encodings may not improve Transformer language models, we discuss corresponding references in Sec. 5.

Question 5. *Why are positional encodings **needed** for **one-layer autoregressive** self-attention?*

This is also well illustrated in Figure 1 by looking at the first layer. Depending on the specific permutation, one-layer model's outputs at some positions are sensitive to the input permutation, but the output at the last position (when the entire sequence is seen) is the same for any permutations; implying that they are not *fully position-sensitive*.

[2]Once it is clear that the model can distinguish between permuted sequences, there is no reason to introduce extra explicit PEs. A common wisdom is to let the model learn to exploit the positional signals on its own. For example, it is rather unnatural to add extra PEs to recurrent neural networks.

4 An Intriguing Linear Transformer Case

Here we discuss an *intriguing* case of linear Transformers (Katharopoulos et al., 2020; Schmidhuber, 1992; Schlag et al., 2021). One representative example of linear Transformers can be obtained by simply removing $\mathrm{softmax}$ in Eq. 4. The resulting model can be equivalently expressed as the following fast weight programmer (see Appendix B):

$$\boldsymbol{q}_t, \boldsymbol{k}_t, \boldsymbol{v}_t = \boldsymbol{W}_q \boldsymbol{x}_t, \boldsymbol{W}_k \boldsymbol{x}_t, \boldsymbol{W}_v \boldsymbol{x}_t \tag{1}$$

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} + \boldsymbol{v}_t \otimes \boldsymbol{k}_t \tag{7}$$

$$\boldsymbol{y}_t = \boldsymbol{W}_t \boldsymbol{q}_t \tag{8}$$

where $\otimes$ denotes outer product and $\boldsymbol{W}_0 = 0$. Because of the state update rule of Eq. 7 giving an impression of "recurrence" (as in the title of Katharopoulos et al. (2020)), it may not be immediately clear if this model requires PEs. In reality this is not true recurrence (for further discussions, we refer to Irie et al. (2021, 2023); Merrill et al. (2024)). Since this model is equivalent to the autoregressive self-attention layer discussed in Sec. 3, it inherits the same properties, i.e., PEs are needed for one-layer models, while they are nonessential for multi-layer models.

5 Literature Review

To the best of our knowledge, Shen et al. (2018) were the first to explore non-symetric "attention masks" (Eq. 6) to encode positional information for neural networks whose sequence processing ability solely relies on the attention mechanism. While the above work does not specifically discuss autoregressive language models (LMs), their insights about masking could have been directly extended to answer the question whether positional encodings are needed for auto-regressive Transformer LMs.

Irie et al. (2019) effectively demonstrated that multi-layer autoregressive Transformer LMs perform well without positional encodings. To be more specific, the corresponding ablation study was conducted for 12, 24, and 42 layer models; while other deeper models (up to 112 layers) were also trained without PEs. This result was later rediscovered/confirmed by Haviv et al. (2022) and Kazemnejad et al. (2023)[3]. Irie et al. (2019)'s argument was the fact that the autoregressive setting itself encodes the positional information due to increasing context over time, which directly connects to Shen et al. (2018)'s argument (while they failed to cite it). Also, while they specifically state that the results are valid for the multi-layer models, they did not explicitly discuss the one-layer case. This no-positional encoding scheme has been immediately adopted in other works on speech recognition; e.g., Zeyer et al. (2019) removed PEs from the decoder of their encoder-decoder speech recognizer. A popular open-source speech toolkit, ESPnet (Watanabe et al., 2018) also integrated Transformer LMs without PEs as part of the standard recipe.

Lee et al. (2019) discussed permutation invariance of non-autoregressive self-attention. Tsai et al. (2019) extended this discussion and showed that autoregressive self-attention is not permutation invariant. However, as discussed above, permutation invariance alone is not enough to conclude on the necessity of PEs (as shown in Sec. 3, one-layer autoregressive models are not *permutation invariant* but also not *fully position-sensitive* and require PEs).

Length Generalization. Empirically, whether removal of PEs yields performance improvements depends on the task. Irie et al. (2019) reported general performance gain by removing PEs for Transformer LMs trained on books, for various numbers of layers. In constrast, Haviv et al. (2022) and Scao et al. (2022) reported slight degradation.

Nevertheless, one of the common benefits of removing PEs is the improved length generalization. Bhattamishra et al. (2020) showed that Transformer LMs without PEs can generalize on certain formal languages with test sequences that are longer than the training ones. Kazemnejad et al. (2023) showed that LMs without PEs yields the best length generalization performance on reasoning-like tasks compared to sophisticated positional encoding methods. Schlag et al. (2021) successfully trained deep linear Transformers without PEs (Sec. 4) by carrying context across training batches to enable them to process arbitrarily long sequences.

For readers interested in length generalization of Transformers with *non*-autoregressive components, we refer to, e.g., Csordás et al. (2021, 2022).

6 Conclusion

We provide a didactic explanation on why positional encodings are nonessential for multi-layer autoregressive Transformers—well-known to pre-GPT practitioners but never described before. We review original work that introduced this result, correcting wrong attributions in recent work.

[3]Many recent works wrongly attribute the origin of this result to these works (see Appendix A for examples).

Limitations

The literature review provided here reflects the authors' best knowledge. It is not impossible that there is prior work discussing the needlessness of positional encodings in autoregressive Transformers (we plan to make this paper updatable on arXiv if the paper is accepted, so that we will be able to integrate such findings).

Also, we only focused on the specific topic of needlessness of positional encodings for autoregressive Transformer language models. There are also other results about language modeling from the pre-GPT era, which did not survive the transition to the GPT-era. This includes discussions on methods to manage/reduce the size of key value-memory cache in auto-regressive Transformers (e.g., compare recent works by Liu et al. (2023); Ge et al. (2024) with a pre-GPT one (Irie et al., 2020)). The scope of this short paper is not an attempt at providing an exhaustive list of such results.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *Preprint arXiv:2303.08774*.

Mark A. Aizerman, Emmanuil M. Braverman, and Lev I. Rozonoer. 1964. Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, 25(6):917–936.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 3159–3166, Honolulu, HI, USA.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*.

Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 4331–4339, Barcelona, Spain.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *Int. Conf. on Learning Representations (ICLR)*, New Orleans, LA, USA.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116, Virtual only.

Peter F. Brown, John Cocke, Stephan A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Rossin. 1988. A statistical approach to language translation. In *Int. Conf. on Computational Linguistics*, pages 71–76, Buffalo, NY, USA.

Tom B Brown et al. 2020. Language models are few-shot learners. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Virtual only.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *Preprint arXiv:2303.12712*.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2022. The Neural Data Router: Adaptive control flow in Transformers improves systematic generalization. In *Int. Conf. on Learning Representations (ICLR)*, Virtual only.

Hugo Cui, Freya Behrens, Florent Krzakala, and Lenka Zdeborová. 2024. A phase transition between positional and semantic learning in a solvable model of dot-product attention. *Preprint arXiv:2402.03902*.

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. In *Proc. Association for Computational Linguistics (ACL)*, pages 2978–2988, Florence, Italy.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. North American Chapter of the Association for Computational Linguistics on Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, MN, USA.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model tells you what to discard: Adaptive kv cache compression for llms. In *Int. Conf. on Learning Representations (ICLR)*, Vienna, Austria.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, Sydney, Australia.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. In *Proc. Findings of Conf. on Empirical Methods in Natural Language Processing (EMNLP-Findings)*, pages 1382–1390, Abu Dhabi, United Arab Emirates.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2022. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *Proc. Int. Conf. on Machine Learning (ICML)*, Baltimore, MD, USA.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2023. Practical computational power of linear transformers and their recurrent and self-referential extensions. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Sentosa, Singapore.

Kazuki Irie, Alexander Gerstenberger, Ralf Schlüter, and Hermann Ney. 2020. How much self-attention do we need? Trading attention for feed-forward layers. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6154–6158.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. 2021. Going beyond linear transformers with recurrent fast weight programmers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Virtual only.

Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. Language modeling with deep Transformers. In *Proc. Interspeech*, pages 3905–3909, Graz, Austria.

Frederick Jelinek, Lalit Bahl, and Robert Mercer. 1975. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 21(3):250–256.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *Preprint arXiv:2307.10169*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 3744–3753, Long Beach, CA.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2022. Transformers learn shortcuts to automata. *Preprint arXiv:2210.10749*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *Int. Conf. on Learning Representations (ICLR)*, Vancouver, Canada.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA.

William Merrill, Jackson Petty, and Ashish Sabharwal. 2024. The illusion of state in state-space models. In *Proc. Int. Conf. on Machine Learning (ICML)*, Vienna, Austria.

Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proc. Interspeech*, pages 1045–1048, Makuhari, Japan.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan H Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531, Prague, Czech Republic.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. [Available Online] : `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. [Available Online] : `https://openai.com/index/better-language-models/`.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code Llama: Open foundation models for code. *Preprint arXiv:2308.12950*.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Stas Bekman, M. Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, Ofir Press, Colin Raffel, Victor Sanh, Sheng Shen, Lintang Sutawika, Jaesung Tae, Zheng Xin Yong, Julien Launay, and Iz Beltagy. 2022. What language model to train if you have one million GPU hours? In *Proc. Findings of Conf. on Empirical Methods in Natural Language Processing (EMNLP-Findings)*, pages 765–782, Abu Dhabi, UAE.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear Transformers are secretly fast weight programmers. In *Proc. Int. Conf. on Machine Learning (ICML)*, Virtual only.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 5446–5455, New Orleans, LA, USA.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 1017–1024, Bellevue, WA, USA.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4344–4353, Hong Kong, China.

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. Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, Long Beach, CA, USA.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. Interspeech*, pages 2207–2211.

Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2019. A comparison of transformer and LSTM encoder decoder models for asr. In *Proc. IEEE Automatic Speech Recog. and Understanding Workshop (ASRU)*, Sentosa, Singapore.

A Some examples of wrong attributions

It is very unfortunate how easily misinformation about the origin of the no PE result has propagated in the current machine learning community (see our Sec. 5 for proper references). Here are some example quotes from the corresponding papers to illustrate the problem:

- Transformers Learn Shortcuts to Automata (Liu et al., 2022): *"Note that removing positional encoding does not mean having no position information, since the use of the causal mask implicitly encodes the position, which is also noted in Bhattamishra et al. (2020) and concurrent work by Haviv et al. (2022)."*

  This statement is wrong. Not only Haviv et al. is not at all concurrent to Bhattamishra 2020. The older works are not referenced.

- Code Llama: Open Foundation Models for Code (Roziere et al., 2023): *"Recent work suggests that causal models do not require an explicit encoding of position information (Haviv et al., 2022; Kazemnejad et al., 2023)"*

  This statement is misleading, since these results are not "recent" findings.

- Flamingo: a Visual Language Model for Few-Shot Learning (Alayrac et al., 2022): *"recent work has shown that such disambiguation is still possible implicitly through the causal attention mechanism [36]."*

  We see the same problem here ([36] is Haviv et al.). This is wrong. The result is not "recent".

- A Phase Transition between Positional and Semantic Learning in a Solvable Model of Dot-Product Attention (Cui et al., 2024): *"While some transformers can leverage implicit positional information through causal masks in training (Haviv et al., 2022; Sinha et al., 2022; Kazemnejad et al., 2023)"*

  Again the authors seem to believe that such results are only known from these recent works, which is wrong.

- Challenges and Applications of Large Language Models (Kaddour et al., 2023): *"Surprisingly, Haviv et al. [192] find that causal LLMs without positional encodings are competitive compared to models with positional encodings and accredit this success to the causal attention mask leaking positional information into the model."*

  We see the same issue here.

B Reminder: Derivation of the dual form of linear Transformers

Here we briefly review the derivation (Katharopoulos et al., 2020; Schlag et al., 2021; Ba et al., 2016) connecting the fast weight programmer of Sec. 4 and its attention form (Eqs. 1-4). Starting from

Eq. 8, and by using the definition of $\boldsymbol{W}_t$ from Eq. 7, we obtain:

$$\boldsymbol{y}_t = \boldsymbol{W}_t \boldsymbol{q}_t \tag{9}$$

$$= \left(\sum_{\tau=1}^{t} \boldsymbol{v}_\tau \otimes \boldsymbol{k}_\tau \right) \boldsymbol{q}_t \tag{10}$$

$$= \sum_{\tau=1}^{t} \boldsymbol{v}_\tau \boldsymbol{k}_\tau^\top \boldsymbol{q}_t \tag{11}$$

$$= \boldsymbol{V}_t \boldsymbol{K}_t^\top \boldsymbol{q}_t \tag{12}$$

where the definitions of $\boldsymbol{K}_t$ and $\boldsymbol{V}_t$ are as in Sec. 2.

The last equation is effectively Eq. 4 without $\mathrm{softmax}$.

Note that this relation is analogous to the famous *duality* that connects the perceptron to kernel machines (Aizerman et al., 1964; Irie et al., 2022).