On the Confounding Effects of Length Generalization With Randomized Positional Encodings

Anonymous ACL submission

Abstract

Transformers generalize exceptionally well on tasks with a fixed context length. However, this capability rapidly diminishes when test sequences are far longer than any sequence seen during training. Unfortunately, simply training on longer sequences is computationally infeasible due to the quadratic cost of attention. Randomized positional encodings were shown to alleviate this issue on algorithmic reasoning tasks, where position is of high importance, but it is unclear if their benefits also transfer to "real-world" tasks such as image classification or natural language processing, which may have different inductive biases. Therefore, in this work, we analyze these randomized encodings on such tasks. Moreover, we show that fine-tuning pretrained models with randomized positional encodings improves length generalization. Finally, we demonstrate that evaluating length generalization on natural language can be misleading due to its short-range dependencies, whereas algorithmic reasoning and vision reveal the limits of prior work and the effectiveness of randomized positional encodings.

1 Introduction

Transformers (Vaswani et al., 2017) perform exceptionally well on sequence modeling tasks across various domains, including natural language processing (NLP) (Devlin et al., 2019), reinforcement learning (Reed et al., 2022), and image recognition (Dosovitskiy et al., 2021). Accordingly, there is a growing demand to employ Transformers on longer sequences, e.g., increasing image resolution. However, it is infeasible to simply increase the length of training sequences due to the quadratic time and space complexity of the Transformer's attention mechanism. Unfortunately, Transformers also generalize less well to longer sequences than other architectures such as RNNs (Delétang et al., 2023). Consequently, boosting Transformers' length generalization capabilities is a rapidly growing research area (Ruoss et al., 2023).

Positional embeddings are one of Transformers' principal failure modes for length generalization (Shaw et al., 2018). Since attention is permutation-invariant, Transformers rely on positional embeddings to inject positional information into their computation, which is crucially important for tasks such as language modeling or algorithmic reasoning. However, traditional positional encodings are out-of-distribution at test time since the model never observed the larger test positions.

Current solutions to this problem typically rely on one of two approaches: (i) using relative instead of absolute positional information, and (ii) additionally randomizing the relative information during training (and test) time. However, while improving performance on language datasets, deterministic relative encodings simply discount faraway information, which cannot induce generic length generalization. In contrast, probabilistic encodings (Ruoss et al., 2023; Likhomanenko et al., 2021) force Transformers to operate solely on order information by decoupling a token's positional information from its position in the sequence. For example, Ruoss et al. (2023) subsample a set of ordered positions from a range that is much longer than the maximum test sequence length, thus reducing train-test distribution shift since test positions will have been observed during training.

We extend the analysis of Ruoss et al. (2023) from algorithmic reasoning to the real-world domains of natural language and vision. We show that natural language is characterized by different inductive biases than image classification or algorithmic reasoning and thus not suited for evaluating length generalization. Concretely, we demonstrate that relative encodings exploit the recency bias of language, but fail to generalize on image classification, unlike randomized encodings. Moreover, we investigate whether pretrained models trained with classical positional encodings can be fine-tuned to longer sequence lengths via randomized encodings.

Contributions Our main contributions are:

- We conduct an empirical evaluation of randomized positional encodings across two real-world data modalities: NLP and vision.
- We show that pretrained models can be fine-tuned with other (randomized) encodings.

2 Related Work

The Transformer architecture (Vaswani et al., 2017) famously replaced all recurrent computation in previous machine translation models with multi-head attention. However, while scalable and performant, dot-product attention itself is permutation invariant, which is why Vaswani et al. (2017) augmented the Transformer's token embeddings by adding scaled *sinusoids* to inject positional information.

The subsequent success of Transformers consequently sparked a flurry of attempts to improve these positional encodings: Gehring et al. (2017) added *learned* positional embeddings to the token embeddings. Dai et al. (2019) proposed to compute the attention at every layer with the *relative* distances between queries and keys to improve long-term (inter-context) dependency modeling. Su et al. (2021) suggested treating the token embeddings as a collection of 2D vectors and *rotating* them in every layer to encode positional information. Press et al. (2022) introduced *ALiBi* encodings to improve length generalization on NLP tasks by adding constant biases, inversely proportional to the key-query distance (known as ALiBi slopes), to the attention score. Chi et al. (2022) presented *KERPLE* embeddings, which replace ALiBi's constant slopes with learnable parameters. Chi et al. (2023) developed *Sandwich* encodings which drops the cross-terms between semantic and positional information in the attention, creating a parameter-free relative positional embedding.

While most of the above approaches aimed at improving Transformers' performance for a fixed-length setting in a deterministic manner, a different line of work tried to boost their length generalization performance via probabilistic positional encodings. Ruoss et al. (2023) developed the *randomized positional encoding* (RPE) scheme, which is compatible with all the above approaches, and randomizes the position associated with each token while maintaining the relative order between tokens. Concurrently, Li and McClelland (2022) introduced a special case of RPEs (for learned positional encodings). However, both works only investigated length generalization on algorithmic reasoning tasks. In contrast, Kiyono et al. (2021) presented *SHAPE* encodings, which only randomize the offset of the sequence's start position instead of randomizing the distances betweeen tokens, and showed improved BLEU performance on NLP tasks. In a similar vein, Likhomanenko et al. (2021) proposed *CAPE* encodings, which first scale the positions into the range $[-1, 1]$ and then apply a set of randomization stages similar to RPEs, and demonstrated that they boost generalization on machine translation, image and speech recognition. Finally, Kazemnejad et al. (2023) showed that positional encodings are unnecessary for length generalization of *decoder-only* Transformers since their causal attention masking is sufficient to represent absolute and relative positional embeddings.

3 Methods

We investigate the length generalization performance of randomized positional encodings on natural language processing and image classification.

3.1 Randomized Positional Encodings

The motivation for randomized positional encodings (Ruoss et al., 2023) stems from the observation that the distribution over token positions is different at training and test time in the context of length generalization, leading to a distribution shift that current Transformer architectures cannot handle.

Concretely, consider the case where the length of the longest sequence in the training set is N. The goal of length generalization is to achieve good performance on sequences of length $M \gg N$. To that end, the randomized positional encodings for token $1 \leq j \leq N$ are given by $\mathrm{RPE}(j, \cdot) := \mathrm{PE}(i_j, \cdot)$, where i_j is a randomly sampled index from a much larger range $\{1, \dots, L\}$ for a configurable hyperparameter L such that $M \leq L$. Note that PE refers to an arbitrary positional encoding scheme (such as $\sin / \cos$) and $\cdot$ refers to the model dimension.

To sample the indices, consider the discrete uniform distribution $\mathcal{U}(S)$ over some set S and let $P_k := \{S \subseteq \{1, \dots, L\} \mid |S| = k\}$. At each training step, for a sequence of length $n \in \{1, \dots, N\}$, randomized positional encodings sample a random set of indices $I \in \mathcal{U}(P_n)$ and then sort I in ascending order such that $I = \{i_1, i_2, \dots, i_n\}$ for $i_1 < i_2 < \cdots < i_n$. Note that, by construction of the set of sets P_k, the indices forming I are distinct.

3.2 Natural Language Processing

While evaluating positional encodings on algorithmic tasks can provide us with interesting insights, they cannot be substitutes for assessment on "real-world" tasks. NLP is the primary use case of Transformers and thus a task of paramount importance when assessing their length generalization capabilities. To that end, we consider the enwik8 dataset, which is a byte (i.e., character)-level dataset formed from the first 100 million bytes of an English Wikipedia XML dump (Hutter, 2006).

We train decoder-only Transformer models with 8 blocks of 8 heads each ($d_{\text{model}} = 256$) on text sequences of length 256 and evaluate on length 1024. We consider 10 different positional encoding schemes (Vaswani et al., 2017; Press et al., 2022; Dai et al., 2019; Su et al., 2021; Gehring et al., 2017; Chi et al., 2022, 2023) and their randomized variants (Ruoss et al., 2023), yielding 18 different models. We train each model for 1 000 000 steps with a batch size of 64 using the Adam optimizer (Kingma and Ba, 2015) with gradient clipping (to an L2 norm of 1), a learning rate of 1×10^{-4}, and 3 parameter initialization seeds.

Fine-tuning As pretrained foundation models are becoming increasingly available (Touvron et al., 2023a,b), a key question is whether they can be efficiently fine-tuned to longer sequences lengths without a performance drop. Unfortunately, the straightforward approach of fine-tuning on longer sequences only yields limited success (Anil et al., 2022; Jelassi et al., 2023). Instead, we investigate whether pretrained models, trained with classical positional encodings, can be fine-tuned *on short sequences* via randomized positional encodings. To that end, we fine-tune a pretrained (via the same setup as above) decoder-only Transformer that uses rotary embeddings (Su et al., 2021), which are commonly employed in foundation models (Touvron et al., 2023a,b). We fine-tune with rotary, ALiBi (Press et al., 2022) and randomized rotary encodings (Ruoss et al., 2023) on sequences of length 256 for 1 000 000 steps and evaluate length generalization on sequences of length 1024.

3.3 Image Classification

Natural language is characterized by a strong recency bias – faraway words rarely tend to have a big impact on predicting the next token (Khandelwal et al., 2018). Therefore, we also consider a real-world dataset that requires the effective use of distant context for correct output. To that end, we investigate image classification in a sequence-to-sequence setting (i.e., with flattened images). Accurate classification requires aggregating the pixel information surrounding each pixel, which will be located in remote places for a flattened image.

We consider the ImageNet dataset (Russakovsky et al., 2015). We preprocess the images by converting them to grayscale and resizing them to 22×23 (yielding a flattened sequence length of 506) for training and 45×45 (i.e., length 2025) for evaluation. However, since flattening the image removes the information of where a row begins, we append a row delimiter (i.e., a black pixel) to the end of every row (leading to a considerable improvement on training sequences). We train an encoder-decoder Transformer by feeding the flattened images to the encoder and a beginning-of-sequence token to the decoder to predict the correct class (out of 1000). We use the same architectures as in Section 3.2 and train them for 1 000 000 steps with a batch size of 32. We use the Adam optimizer (Kingma and Ba, 2015) with gradient clipping and a learning rate of 1×10^{-5} and 3 parameter initialization seeds.

Table 1: The minimum cross-entropy loss on enwik8 (3 random seeds) for a decoder-only Transformer with different positional encodings. We trained on sequences of length 256 and evaluated on length 1024. Randomized positional encodings significantly degrade the performance due to the inductive bias of this natural language dataset where queries only need to attend to nearby keys.

	Length 256		Length 1024	
Positional Encoding	**Det.**	**Rand.**	**Det.**	**Rand.**
None	232.8	NA	4194.0	NA
sin / cos	222.5	225.9	3641.5	3326.3
ALiBi	218.1	228.1	859.6	1603.4
Relative	**216.2**	219.1	854.4	1379.9
Rotary	218.8	222.4	4259.9	1883.9
Learned	223.8	230.7	3160.0	5234.1
Power KERPLE	216.8	221.8	**845.3**	1151.0
Log KERPLE	217.0	221.0	850.8	1559.0
Sandwich	220.0	225.3	1337.7	1715.2
SHAPE	225.4	NA	5844.8	NA

4 Results

We now present our extensive experimental evaluation on natural language and vision datasets.

4.1 Natural Language

Table 1 shows our evaluation of the decoder-only Transformer on enwik8 with different positional en-

Table 2: The minimum cross-entropy loss on enwik8 (3 random seeds) when fine-tuning a decoder-only Transformer that is pretrained with rotary positional encodings. We pretrained and fine-tuned on sequences of length 256 and evaluate on length 1024. ALiBi achieves the best length generalization performance.

		Fine-tuned		
Length	**Pretrained**	**Rotary**	**Rand. Rotary**	**ALiBi**
256	218.8	**215.2**	219.6	217.4
1024	4259.9	4653.7	1847.5	**955.0**

coding schemes. We observe that non-randomized relative positional embeddings (i.e., KERPLE, relative, and ALiBi) achieve the best length generalization performance. This is expected due to the dataset's character-level nature: Given the string "... appl", a model can correctly predict the character 'e' without needing to consider the long-range context the word lies in. Therefore, the windowed inductive bias of relative encodings (queries simply attend to nearby keys) leads to favorable results.

Note that randomized positional encodings do help to improve the performance of absolute embeddings (sin / cos and rotary). This confirms the hypothesis from Ruoss et al. (2023), which states that randomization allows the model to train on positions that would otherwise be out-of-distribution at evaluation time. We visualize the change in attention patterns after randomization in Fig. B.1.

Fine-Tuning Table 2 shows the results of fine-tuning pretrained models trained with rotary embeddings on enwik8. Fine-tuning with randomized encodings considerably reduces the length generalization loss compared to the pretrained model or fine-tuning with the same (i.e., rotary) embeddings. However, fine-tuning with ALiBi decreases the loss even further, particularly for length generalization. Thus, fine-tuning pretrained models with a different positional encoding scheme appears to be a viable strategy. However, we recall that this natural language dataset is characterized by short-range dependencies (as evidenced by ALiBi's superior performance), with positive results not necessarily indicative of true length generalization.

4.2 Image Classification

In Table 3 we present our evaluation of the encoder-decoder Transformer on ImageNet with different positional encodings. This task uncovers the limits of relative encodings (and, by extension, win-

Table 3: The minimum cross-entropy loss on ImageNet (3 random seeds) for an encoder-decoder Transformer with different positional encodings. We converted the images to grayscale and flattened them using delimiters to distinguish between rows. We trained on sequences of length 506 (images of size 22×23) and evaluated on length 2025 (images of size 45×45). Randomized encodings substantially improve the length generalization performance since the classification task requires attending to faraway information (i.e., across rows).

	Length 506		Length 2025	
Positional Encoding	**Det.**	**Rand.**	**Det.**	**Rand.**
None	5.741	N/A	6.115	N/A
sin / cos	5.257	5.371	7.369	6.133
ALiBi	4.949	5.017	7.120	5.373
Relative	4.397	4.921	7.934	**5.141**
Rotary	**4.325**	5.117	7.946	5.341
Learned	5.262	5.754	7.194	6.140
Power KERPLE	4.929	5.252	6.221	5.427
Log KERPLE	5.152	5.283	5.526	5.876
Sandwich	5.161	5.443	5.497	7.011
SHAPE	5.563	N/A	6.332	N/A

dowed attention) in terms of length generalization since they perform worse than a bag-of-pixels (i.e., no positional encodings) approach. In contrast, randomized positional encodings significantly improve length generalization across the board (except for Sandwich). Finally, note that SHAPE performs rather poorly, showing that randomizing only the absolute sequence offset is insufficient.

5 Conclusion

We conducted an extensive empirical investigation of the length generalization capabilities of randomized positional encodings on natural language processing and image recognition. We showed that relative positional embeddings triumph on enwik8 but fail to generalize on ImageNet classification, unlike randomized encodings. Thus, our results indicate that the absence of true length generalization is often hidden by the use of language benchmarks but becomes apparent when evaluating tasks with long-range dependencies, e.g., vision or algorithmic reasoning. Moreover, we showed that offset randomization alone is insufficient to generalize to longer sequences (SHAPE's performance is mediocre on ImageNet). Finally, we demonstrated that models pretrained with classical embeddings can be fine-tuned with a different (randomized) encoding scheme to boost their length generalization.

Limitations

Our work provides a comprehensive comparison of the length generalization capabilities of different positional encoding schemes on the enwik8 language modeling task and ImageNet image classification (phrased as a sequence-to-sequence modeling task). Nevertheless, some limitations have to be considered. First, these two datasets do not capture the full diversity and complexity of either domain, and further domains need to be evaluated for a more complete analysis of the strengths and shortcomings of randomized positional encodings on real-world data. Second, the quadratic cost of attention induces a memory bottleneck and it is unclear whether and how our results would extend to longer sequence lengths. This is less of a problem when evaluating algorithmic reasoning tasks, which share the same structure across all sequence lengths (i.e., their complexity is independent of the sequence length). In contrast, for real-world data, complexity is often more related to sequence length (e.g., evaluating $45 \times 45 = 2025$ pixel grayscale images is far from the current state-of-the-art in image recognition). Finally, it has been shown that certain capabilities of LLMs only emerge with scale (Wei et al., 2022), and thus our empirical evaluation would have to be repeated with models of increasingly larger size to investigate how model scale impacts the length generalization capabilities of randomized positional encodings. Overall, we believe that our study's limitation open up several interesting avenues for future research.

References

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay V. Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *NeurIPS*.

Ta-Chung Chi, Ting-Han Fan, Peter J. Ramadge, and Alexander Rudnicky. 2022. KERPLE: kernelized relative positional embedding for length extrapolation. In *NeurIPS*.

Ta-Chung Chi, Ting-Han Fan, Alexander Rudnicky, and Peter J. Ramadge. 2023. Dissecting transformer length extrapolation via the lens of receptive field analysis. In *ACL (1)*.

Noam Chomsky. 1956. Three models for the description of language. *IRE Trans. Inf. Theory*.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2022. The neural data router: Adaptive control flow in transformers improves systematic generalization. In *ICLR*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL (1)*.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. Neural networks and the chomsky hierarchy. In *ICLR*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Marcus Hutter. 2006. 500'000€ prize for compressing human knowledge.

Samy Jelassi, Stéphane d'Ascoli, Carles Domingo-Enrich, Yuhuai Wu, Yuanzhi Li, and François Charton. 2023. Length generalization in arithmetic transformers. *arXiv:2306.15400*.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *arXiv:2305.19466*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *ACL (1)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE : Shifted absolute position embedding for transformers. In *EMNLP (1)*.

Yuxuan Li and James L. McClelland. 2022. Systematic generalization and emergent structures in transformers trained on structured tasks. *arXiv:2210.00400*.

Tatiana Likhomanenko, Qiantong Xu, Gabriel Synnaeve, Ronan Collobert, and Alex Rogozhnikov. 2021. CAPE: encoding relative positions with continuous augmented positional embeddings. In *NeurIPS*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A generalist agent. *Trans. Mach. Learn. Res.*

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. 2023. Randomized positional encodings boost length generalization of transformers. In *ACL (2)*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL-HLT (2)*.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv:2104.09864*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*

A Experimental Details

We run every task-encoding-hyperparameter triplet on a single NVIDIA V100 GPU from our internal cluster. As a result we used 18 (positional encodings) $\cdot$ 3 (seeds) $=$ 54 GPU-units for the results in Tables 1 and 3, 3 (positional encodings) $\cdot$ 3 (seeds) $=$ 9 GPU-units for Table 2, and 15 (tasks) $\cdot$ 7 (positional encodings) $\cdot$ 3 (learning rates) $\cdot$ 3 (seeds) $=$ 945 GPU-units for Table B.1.

B Algorithmic Reasoning Tasks

Randomized positional encodings (Ruoss et al., 2023) were originally only evaluated on the algorithmic reasoning benchmark proposed by Delétang et al. (2023). However, the evaluation conducted by Ruoss et al. (2023) did not include a comparison with the more recent positional encoding schemes SHAPE (Kiyono et al., 2021), KERPLE (Chi et al., 2022), and Sandwich (Chi et al., 2023). Thus, we complement the results of Ruoss et al. (2023) with an evaluation of these positional encodings on the same algorithmic reasoning tasks.

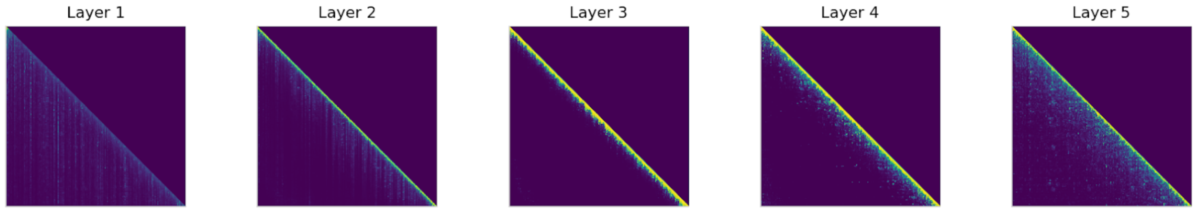
Experimental Setup We consider the same experimental setup as proposed by Delétang et al. (2023) and used by Ruoss et al. (2023). The benchmark consists of 15 algorithmic reasoning tasks spanning the Chomsky hierarchy (Chomsky, 1956), the details of which are irrelevant for the purposes of this study. The benchmark is publicly available at `https://github.com/deepmind/neural_networks_chomsky_hierarchy` under the Apache 2.0 License. The tasks are not composed of fixed-sized datasets but are sampled from data-generating distributions. We considered encoder-only Transformers (Vaswani et al., 2017) with 5 blocks of 8 heads each with $d_{\text{model}} = 64$. We train all models for 2 000 000 steps with a batch size of 128, corresponding to 256 000 000 (potentially non-unique) training

Table B.1: Accuracy (in percentage) averaged over all test lengths and maximized over 3 random seeds and 3 learning rates. The random accuracy is 50% except for MODULAR ARITHMETIC (SIMPLE), CYCLE NAVIGATION, BUCKET SORT, and MODULAR ARITHMETIC, where it is 20%. As reported by Ruoss et al. (2023), randomized positional encodings increase the test accuracy (by 7.4% on average). † denotes permutation-invariant tasks, which can be solved without positional information. SHAPE is a probabilistic positional encoding (it samples the position offset) and thus cannot be randomized further via the randomized positional encoding scheme of Ruoss et al. (2023). For ease of comparison, we report the highest accuracy per task from Ruoss et al. (2023) in the right-most column (marked with $\star$). This column thus represents the highest accuracy achieved across the (randomized) versions of $\sin / \cos$, relative, ALiBi, RoPE, and learned encodings, as well as not using a positional encoding at all.

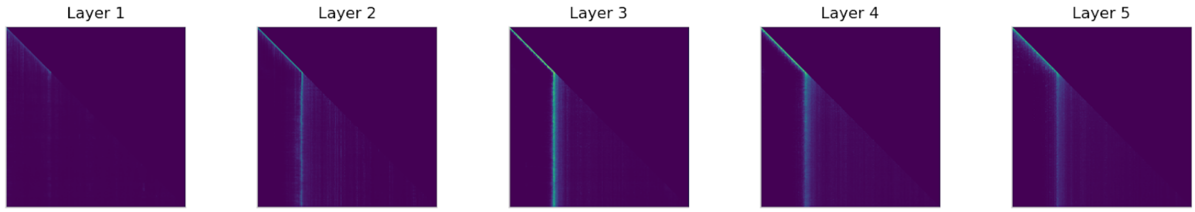
						Randomized			
Level	**Task**	**Power KERPLE**	**Log KERPLE**	**Sandwich**	**SHAPE**	**Power KERPLE**	**Log KERPLE**	**Sandwich**	**Ruoss et al. (2023)★**
R	EVEN PAIRS	69.3	**100.0**	93.0	81.3	99.9	75.5	61.4	**100.0**
	MODULAR ARITHMETIC (SIMPLE)	27.6	30.5	29.0	31.4	29.4	**33.6**	26.3	28.1
	PARITY CHECK†	56.1	55.6	55.6	55.9	55.0	**56.6**	54.6	52.6
	CYCLE NAVIGATION†	42.2	60.1	41.6	48.5	60.3	71.8	36.3	**73.6**
DCF	STACK MANIPULATION	59.9	61.5	62.2	54.7	72.3	75.7	62.9	**77.9**
	REVERSE STRING	66.3	63.9	78.7	57.3	77.9	81.5	61.9	**95.1**
	MODULAR ARITHMETIC	38.2	37.0	**38.7**	37.5	38.5	**38.7**	36.8	34.9
	SOLVE EQUATION	29.3	**30.9**	30.3	28.1	30.3	29.9	28.9	28.1
CS	DUPLICATE STRING	56.5	59.6	58.5	56.1	73.4	74.1	58.4	**75.1**
	MISSING DUPLICATE	58.1	61.8	68.9	55.1	91.3	85.5	68.1	**100.0**
	ODDS FIRST	56.0	59.5	56.8	56.0	**69.9**	68.6	57.5	69.3
	BINARY ADDITION	55.3	57.1	58.7	55.5	61.8	63.0	56.5	**64.5**
	BINARY MULTIPLICATION	55.6	55.5	**55.8**	54.5	55.3	55.3	53.3	52.1
	COMPUTE SQRT	56.0	**57.0**	55.5	55.4	55.7	54.9	53.5	53.3
	BUCKET SORT†	49.0	94.9	98.1	38.8	99.8	99.9	93.5	**100.0**

examples. We sample the length of every training sequence uniformly from the range $\{1, \ldots, 40\}$. We evaluate the length generalization on a single batch of 500 testing examples for all sequence lengths in $\{41, \ldots, 500\}$. We used the Adam optimized (Kingma and Ba, 2015) with gradient clipping (to an L2 norm of 1) and sweeped over three learning rates (1×10^{-4}, 3×10^{-4}, 5×10^{-4}) using 3 different parameter initialization seeds.

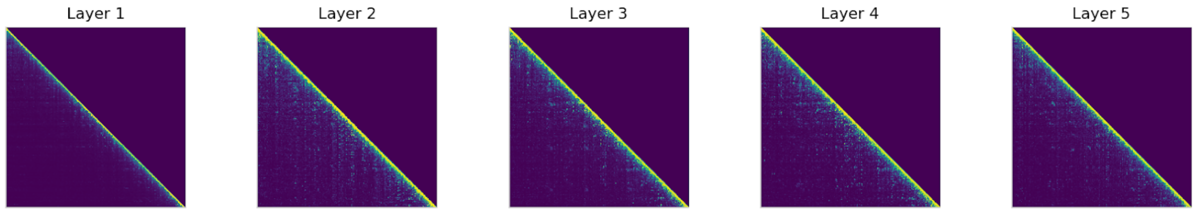
Results Table B.1 shows the accuracy of the different encodings across all different tasks. We observe that the randomized variants increase the test accuracy by 7.4% on average. Interestingly, Sandwich are the only encodings that do not seem to benefit from randomization. Finally, note that SHAPE fails to perform significantly better than random all tasks apart from EVEN PAIRS. This failure shows that only randomizing the positional offset of the sequence during training (and not also the distances between tokens) is insufficient to achieve good length generalization.



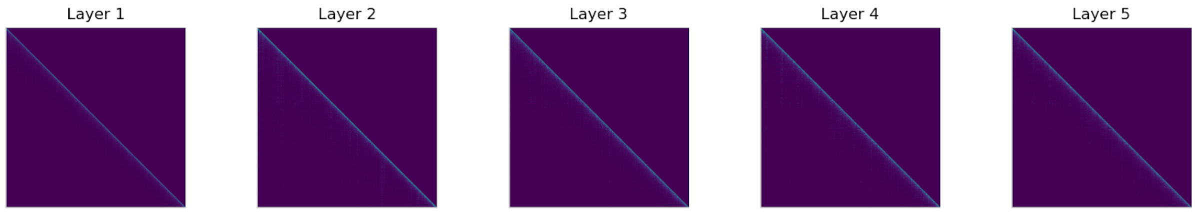
(a) $\sin$ / $\cos$ with a sequence length of 256 (in-distribution).



(b) $\sin$ / $\cos$ with a sequence length of 1024 (out-of-distribution)



(c) Randomized $\sin$ / $\cos$ with a sequence length of 256 (in-distribution).



(d) Randomized $\sin$ / $\cos$ with a sequence length of 1024 (out-of-distribution)

Figure B.1: Analysis of the attention matrices for the $\sin$ / $\cos$ and randomized $\sin$ / $\cos$ positional encodings on enwik8 using sequences of length 256 (training length) and 1024 (evaluation length). We visualize the maximum over the 8 heads per layer (following Csordás et al. 2022) and observe a clear diagonal pattern, which corresponds to the short-range dependencies observed in natural language (Khandelwal et al., 2018). The randomized positional encodings maintain the pattern on longer sequences, while it breaks down for the standard positional encoding.