Position Encoding with Random Float Sampling Enhances Length Generalization of Transformers

Atsushi Shimizu[*]
Daiwa Securities
atsushi.shimizu@daiwa.co.jp

Shohei Taniguchi
The University of Tokyo
taniguchi@weblab.t.u-tokyo.ac.jp

Yutaka Matsuo
The University of Tokyo
matsuo@weblab.t.u-tokyo.ac.jp

Abstract

Length generalization is the ability of language models to maintain performance on inputs longer than those seen during pretraining. In this work, we introduce a simple yet powerful position encoding (PE) strategy, Random Float Sampling (RFS), that generalizes well to lengths unseen during pretraining or fine-tuning. In particular, instead of selecting position indices from a predefined discrete set, RFS uses randomly sampled continuous values, thereby avoiding out-of-distribution (OOD) issues on unseen lengths by exposing the model to diverse indices during training. Since assigning indices to tokens is a common and fundamental procedure in widely used PEs, the advantage of RFS can easily be incorporated into, for instance, the absolute sinusoidal encoding, RoPE, and ALiBi. Experiments corroborate its effectiveness by showing that RFS results in superior performance in length generalization tasks as well as zero-shot commonsense reasoning benchmarks.

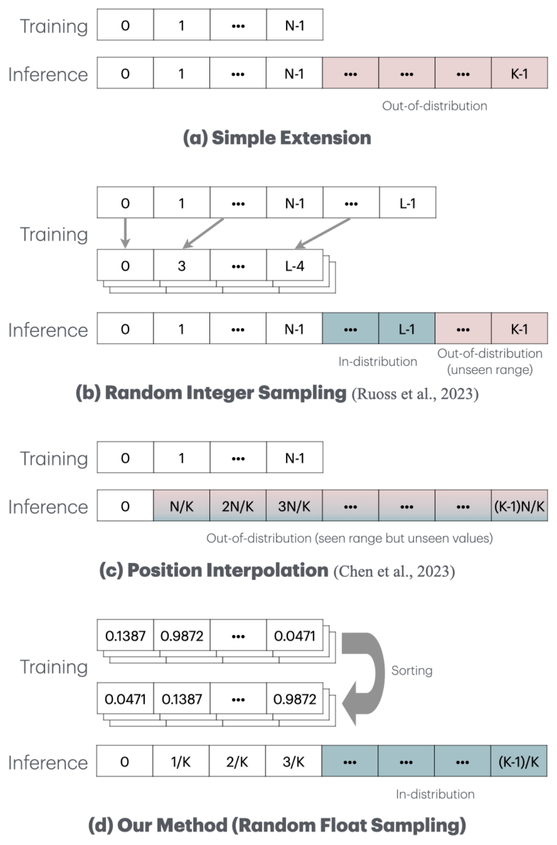

Training
0
1
N-1
Inference
K-1
Out-of-distribution
(a) Simple Extension
L-1
3
L-4
In-distribution
Out-of-distribution (unseen range)
(b) Random Integer Sampling (Ruoss et al., 2023)
N/K
2N/K
3N/K
(K-1)N/K
Out-of-distribution (seen range but unseen values)
(c) Position Interpolation (Chen et al., 2023)
0.1387
0.9872
0.0471
Sorting
1/K
2/K
3/K
(K-1)/K
In-distribution
(d) Our Method (Random Float Sampling)

Figure 1: An illustration of the existing and proposed position indexing strategies. RFS avoids the OOD issue for any context length K, which is unknown during training.

1 Introduction

Length generalization is the problem of training language models that generalize to larger context sizes than those employed for pretraining. It has been a critical challenge because simply feeding longer contexts as in Figure 1 (a) causes an out-of-distribution (OOD) issue and thus fails, as shown empirically by Press et al. (2022); Kazemnejad et al. (2023); Zhou et al. (2024), while training on stretched sequences is costly, especially for Transformer-based models (Vaswani et al., 2017) due to their quadratic complexity. The need to handle longer contexts, however, has grown with advances in inference-time compute, such as prompt engineering (Kojima et al., 2022), Chain-of-Thought (Wei et al., 2022), test-time scaling (Muennighoff et al., 2025), and the AI agents (Yao et al., 2023; Shinn et al., 2023). The rise of small language models for real-world applications (Belcak et al., 2025; LiquidAI, 2025) also necessitate the length generalization strategy as their context sizes are often smaller and prone to overflowing it, e.g., 2,048 for Phi-1.5 (Li et al., 2023), TinyLlama (Zhang et al., 2024), and SmolLM2 (Allal et al., 2025).

Extensive approaches have been proposed to address this issue, including two notable works closely related to ours. Ruoss et al. (2023) propose to sample N integers from $\{1, \cdots, L\}$ uniformly

[*]Work done as a research intern at the University of Tokyo.

at random for the position indices, given the maximum length L and the training window size N with $N < L$, as illustrated in Figure 1 (b). Position interpolation (Chen et al., 2023) down-scales the position index to match the original index range, and performs minimal fine-tuning with longer inputs, as shown in Figure 1 (c). Although both mitigate the issue by overlapping the position index range during training and inference, they still suffer from the OOD issue when the input length exceeds what was assumed during training or fine-tuning.

In this paper, we introduce a novel position indexing approach, Random Float Sampling (RFS), that technically allows the model to work on any input length by strategically selecting position indices from a shared continuous range during training and inference. Specifically, the indices are randomly sampled during training as depicted in Figure 1 (d), allowing the model to learn the attention mechanism based on relative token distances. Consequently, unlike existing PEs relying on absolute indices, the model trained under RFS is better at capturing the sequential structure even when an unexpected number of tokens are squeezed into the index range.

Experiments on length generalization tasks demonstrate its effectiveness. As in Figure 2, for example, with the absolute sinusoidal PE on the copy task, RFS maintains roughly 80% accuracy on twice longer inputs, while the model without PE (NoPE), the best one in Kazemnejad et al. (2023), struggles on even 1.5 times longer inputs with accuracy around 20%. In another experiment on commonsense reasoning ability of language models, RFS shows a strong performance on unseen length test cases, which is comparable to that of a model without our method tested on seen length.

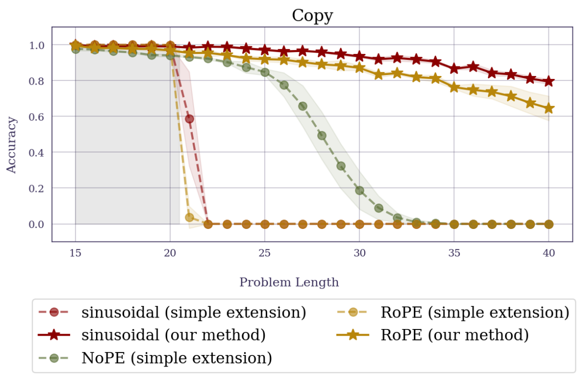


Figure 2: Results of different PEs on the copy task. The shaded area indicates that those lengths are seen during training. RFS boosts the performance significantly and leads to better results than NoPE. The experiment setup is stated in Section 4.1.

2 Related Work

2.1 Position Encoding

Transformers are permutation-equivariant; therefore, how to explicitly encode position information has been studied extensively. One common way is to form a matrix purely containing position information and add it to the token embedding matrix before they are fed into the Transformer blocks. Vaswani et al. (2017) propose the use of sinusoids so that the dot product of two position vectors is roughly proportional to the token index distance. Another approach introduced in Su et al. (2024), named RoPE, also utilizes sinusoidal functions and rotates queries and keys in each attention module to encode the token index distances. Since the above two PEs rely on sequential integer position indices, RFS can immediately replace them.

2.2 Length Generalization

RoPE-based architectures: To train a model with long context length, Dai et al. (2019) and Yu et al. (2023) propose architectures without full attention. Train-short-finetune-long is also considered in Press et al. (2021); Ding et al. (2024), position interpolation (Chen et al., 2023), and YaRN (Peng et al., 2024). Ours belongs to a train-short family that incurs no additional cost for fine-tuning. It includes position indexing strategies such as imposition of randomness for robustness, such as random integer sampling (Ruoss et al., 2023), and others (Kiyono et al., 2021; Likhomanenko et al., 2021).

Attention mechanism modification: The T5 model (Raffel et al., 2020) directly manipulates the attention scores with learnable parameters. This line of work includes ALiBi (Press et al., 2022), which is a lightweight version without learnable parameters. Chi et al. (2022); Zheng et al. (2024); Oka et al. (2025); Nakanishi (2025) also propose improved attention mechanisms.

No position encoding (NoPE): Several studies cast doubt on the necessity of explicit positional information in causal Transformers (Haviv et al., 2022; Kazemnejad et al., 2023; Irie, 2024). They argue that position information can be learned and show that NoPE models perform as well as models with PE. Note that this only applies to causal models, not to bidirectional ones for translation (Vaswani et al., 2017) or transcription (Radford et al., 2023).

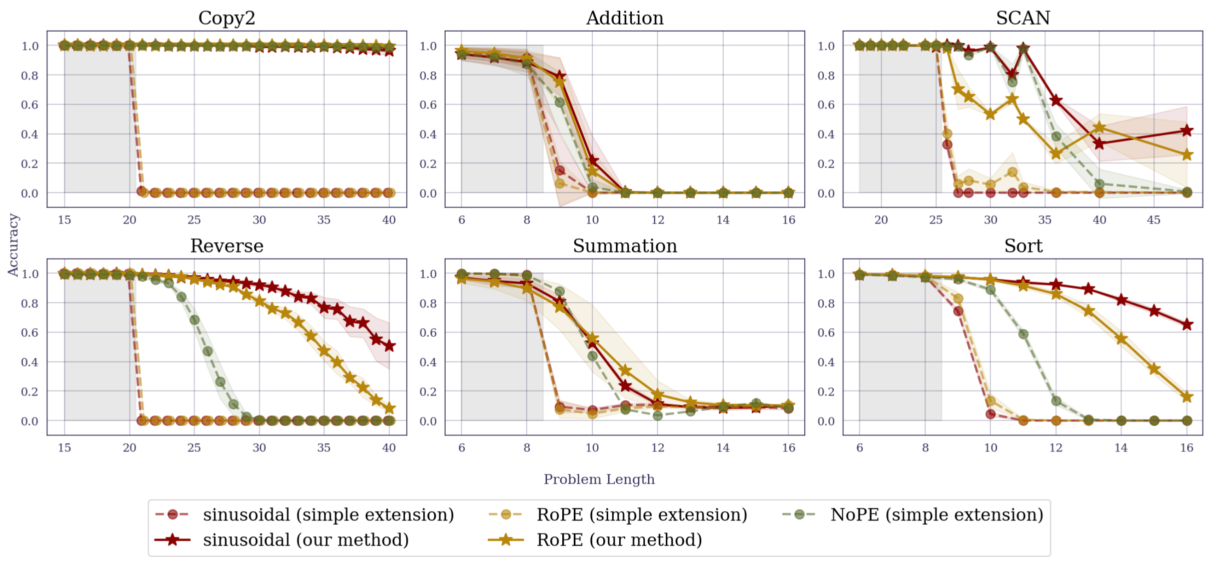


Figure 3: Results of length generalization tasks. The shaded area indicates the input lengths seen during training.

3 Method: Random Float Sampling

In this section, we formally define Random Float Sampling (RFS). Let p_i be the position index of the i-th token and $[0, 1)$ be the continuous range where indices are chosen. During training, given a sequence of length n_{tr}, we sample n_{tr} values from the range independently and uniformly at random, sort them in ascending order, and use them as the position indices. Namely, we have

$$p_i = \tilde{p}_{\pi(i)}, \quad \tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_{n_{\text{tr}}} \overset{\text{i.i.d.}}{\sim} [0, 1),$$

where $\tilde{p}_{\pi(i)}$ is the $\pi(i)$-th smallest in $\tilde{p}_1, \cdots, \tilde{p}_{n_{\text{tr}}}$.

In inference with a context of length n_{in}, the position indices are an evenly spaced partition of the shared range as

$$p_i = \frac{(2i - 1)}{2\max(n_{\text{tr}}, n_{\text{in}})}.$$

The indices are selected deterministically to make the model's output independent of the position index randomness. Note that when $n_{\text{in}} \ll n_{\text{tr}}$, the distances between the indices increase to a level nearly unseen during training. Using the max in the denominator prevents this degeneration.

Finally, the selected indices are scaled by L, a hyperparameter, to match the magnitude of indices commonly used in Transformers. We set it to $1,000$ in our experiments. We again emphasize that once the position indices are obtained, they can immediately be plugged into the various PEs like absolute sinusoidal encoding, RoPE, and ALiBi.

Design Rationale. As position interpolation and random integer sampling imply, and as discussed in Appendix A, we use the shared index range for training and inference. It immediately follows that, in inference, the token index distances vary depending on n_{in}. This is where randomness in training comes in, forcing the model to experience various position index distributions.

4 Experiments

4.1 Length Generalization Tasks

Setup. To test our method's length generalization ability, we follow Kazemnejad et al. (2023), train a model from scratch on sequences up to a predefined length, and test on both seen and unseen lengths. The performance is evaluated by the exact-match accuracy on 6 tasks: copy and reverse (Ontanon et al., 2022), addition (Nye et al., 2021), sort and summation (Saxton et al., 2019), and SCAN (Lake and Baroni, 2018). We refer readers to Kazemnejad et al. (2023) and Appendix B.2, B.4 for more details about the dataset.

Architecture. We use a decoder-only Transformer model with "base" configuration and the T5-Base tokenizer in the HuggingFace library (Wolf et al., 2020) and employ greedy decoding for generation. See Appendix B.4 for the full description. For position encoding, we consider the absolute sinusoidal PE and RoPE. We compare our RFS with other position indexing, such as simple extension, position interpolation (Chen et al., 2023), and random integer sampling (Ruoss et al., 2023). We do not fine-tune any model for fairness, even though Chen et al. (2023) performs minimal fine-tuning in their experiments.

Result. We run the experiment with three different random seeds and report the performance in Figure 2 and 3. The most notable case is copy2; followed by the copy, reverse, and sort, where simple extension exhibits nearly no length generalization capabilities, but RFS still generates correct answers on unseen lengths. Second, RFS performs better than or at least on par with the NoPE, demonstrating the effectiveness of explicit position bias. Figure 4 compares the RoPE-based RFS models with those of position interpolation and random integer sampling, highlighting the superiority of our method. It demonstrates that the advantage of position interpolation cannot be fully brought out without finetuning, while random integer sampling requires more training steps to converge as it underfits. Compared to random integer sampling, which also selects position indices from the predefined range, the use of floats in RFS not only relaxes the context size limitation but also accelerates the learning. Further discussion is given in Appendix C.2. The result of ALiBi, which shows that RFS enhances the length generalization ability beyond absolute sinusoidal encoding and RoPE, is also relegated to Appendix C.3.

Ablation. In Section 3, RFS is defined as being trained with indices taken from the uniform distribution, and $L = 1,000$ is employed. To test the alternative design choices, we run the experiment on the copy task using RoPE and report the average OOD accuracy. Firstly, the uniform distribution is replaced with the normal distribution with 0.5 mean and 0.2 standard deviation, or the beta distribution with $\alpha = 0.5$ and $\beta = 0.5$ for training, while inference-time indices follow those in Section 3. Other variants with inference indices taken from the cumulative density functions of $[0, 1)$ even interpolation are also considered. Secondly, L is modified to 10 and $100,000$. The result, summarized in Table 1, indicates that while simply supplanting the uniform distribution with the normal or beta distribution degrades the system, matching the inference indices distribution using the cumulative density functions performs well. Nevertheless, their OOD accuracy, 81.9% and 81.2% for the normal and beta distribution, respectively, falls short of the baseline 83.7%. The ablation on L results in a slight degradation for both $L = 10$ and $L = 100,000$, showing that L has a minimal impact on the OOD accuracy. Overall, these ablation studies justify the architectural choices of our RFS.

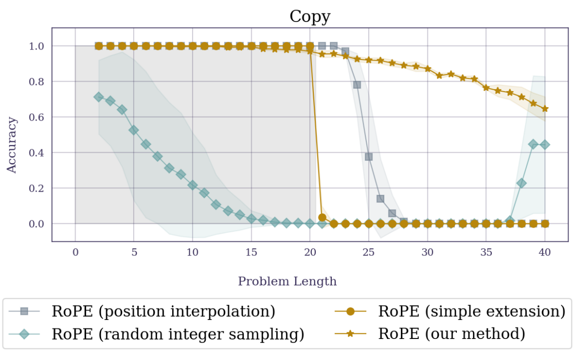


Figure 4: Results of RFS, position interpolation, and random integer sampling on the copy task. The shaded area indicates the input lengths seen during training.

Model	OOD accuracy
Baseline	**83.7**
$\mathcal{N}(0.5, 0.2)$	0.0
$\mathcal{B}(0.5, 0.5)$	15.5
$\mathcal{N}(0.5, 0.2)$-cdf	81.9
$\mathcal{B}(0.5, 0.5)$-cdf	81.2
$L = 10$	83.4
$L = 100,000$	83.6

Table 1: Copy task performance over different design choices.

4.2 Language Modeling

Setup. To evaluate the effectiveness of our method in language modeling, we train language models from scratch on OpenWebText (Gokaslan et al., 2019) and test their zero-shot commonsense reasoning ability on the following 7 tasks: HellaSwag (Zellers et al., 2019), RACE (Lai et al., 2017), ARC-Easy and ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2019), and BoolQ (Wang et al., 2019). Additional details about the dataset are provided in Appendix B.2. To examine the length generalization ability, we set the model's context window to the median number of tokens in each task's test set, splitting the set into in-distribution (ID) and out-of-distribution (OOD) subsets.

Architecture. The models are based on GPT-2 (Radford et al., 2019) with 12 layers, each with 12 heads, and an embedding dimension of 768. We use the same tokenizer with GPT-2. Details are summarized in Appendix B.5.

	Task	**HellaSwag**	**RACE**	**ARC-e**	**ARC-c**	**OBQA**	**WinoGrande**	**BoolQ**	**Avg**
	Context size	87	439	28	33	13	21	130	
ID	Baseline	**35.82**	26.59	35.47	**23.39**	**21.40**	52.50	**55.74**	35.84
	Ours	34.95	**28.52**	**36.48**	22.51	20.96	**52.80**	55.37	**35.94**
OOD	Baseline	**25.24**	**27.38**	**36.75**	20.70	28.78	46.20	56.25	34.47
	Ours	24.56	27.19	36.59	**22.37**	**29.15**	**50.17**	**60.42**	**35.78**

Table 2: Zero-shot performance on commonsense reasoning tasks. ARC-e, ARC-c, and OBQA are short for ARC-Easy, ARC-Challenge, and OpenbookQA, respectively.

Result. Table 2 reports the result. Our RFS showcases the better average accuracy in the OOD test cases, while performing on par with the baseline in the ID cases. Notably, the average accuracy of our method in OOD 35.78% is comparable to that of the baseline in ID tests 35.84%, highlighting the length generalization ability of our PE strategy. Namely, the degraded performance of the baseline on OOD samples, which stems from the lack of length generalization ability, is almost recovered by RFS. Although RFS does not improve OOD accuracy on HellaSwag, RACE, and ARC-Easy, this may come from training data or architectural factors rather than indexing itself; we leave a deeper analysis to future work.

5 Conclusion

In this paper, we introduce Random Float Sampling, a random position indexing strategy in Transformers that places no intrinsic upper bound on context length at the indexing level. RFS is simple enough to be integrated into many commonly used PEs, and its advantage is evaluated from various angles. RFS demonstrates the strongest performance against existing methods on length generalization tasks with a significant margin. It also improves the accuracy on OOD samples in commonsense reasoning tasks to nearly the same level as on their ID counterparts, showcasing the practicality of RFS in language modeling.

Limitations

Firstly, we conducted the experiments on language models with around 100 million parameters, but not on the commercial-size LLMs, due to computational resource constraints. To fully discover its potential in real-world applications, testing on larger LLMs is necessary. Secondly, even though our method technically accepts contexts of any length, its performance deteriorates as the input becomes longer. To compensate for degradation, combining random position indexing with other orthogonal works, such as Scalable-Softmax (Nakanishi, 2025) and interleaving deployment of RoPE and NoPE (Yang et al., 2025), would be an interesting direction for future research.

Acknowledgments

This research was conducted using the FUJITSU Supercomputer PRIMEHPC FX1000 and FUJITSU Server PRIMERGY GX2570 (Wisteria/BDEC-01) at the Information Technology Center, the University of Tokyo.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *Preprint*, arXiv:2506.02153.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. 2022. Kerple: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems*, volume 35, pages 8386–8399. Curran Associates, Inc.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongRoPE: Extending LLM context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11091–11104. PMLR.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1382–1390, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kazuki Irie. 2024. Why are positional encodings nonessential for deep autoregressive transformers? revisiting a petroglyph. *Preprint*, arXiv:2501.00659.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 24892–24928. Curran Associates, Inc.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3309–3321, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *Preprint*, arXiv:2309.05463.

Tatiana Likhomanenko, Qiantong Xu, Gabriel Synnaeve, Ronan Collobert, and Alex Rogozhnikov. 2021. Cape: Encoding relative positions with continuous augmented positional embeddings. In *Advances in Neural Information Processing Systems*, volume 34, pages 16079–16092. Curran Associates, Inc.

LiquidAI. 2025. Introducing liquid nanos — frontier-grade performance on everyday devices.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Ken M. Nakanishi. 2025. Scalable-softmax is superior for attention. *Preprint*, arXiv:2501.19399.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *Preprint*, arXiv:2112.00114.

Yui Oka, Taku Hasegawa, Kyosuke Nishida, and Kuniko Saito. 2025. Wavelet-based positional representation for long context. In *The Thirteenth International Conference on Learning Representations*.

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5493–5505, Online. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *J. Mach. Learn. Res.*, 21(1).

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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1889–1903, Toronto, Canada. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bowen Yang, Bharat Venkitesh, Dwarak Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. 2025. Rope to nope and back again: A new hybrid attention strategy. *Preprint*, arXiv:2501.18795.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. In *Advances in Neural Information Processing Systems*, volume 36, pages 78808–78823. Curran Associates, Inc.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. 2024. DAPE: Data-adaptive positional encoding for length extrapolation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. 2024. Transformers can achieve length generalization but not robustly. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

A Why Does Simple Extension Fail?

Let $\mathbf{x} \in \mathbb{R}^d$ be the vector representation of a token, which is then fed into the attention block and projected to a query and a key by computing $\mathbf{q} = f_q(\mathbf{x})$ and $\mathbf{k} = f_k(\mathbf{x})$ where f_q and f_k represent the linear transformation and rotary positional encoding (RoPE). Chen et al. (2023) shows that, given position differences, there always exist f_q and f_k that produce well-behaved attention scores $\mathbf{q}^T\mathbf{k}$, but the attention scores can explode for unseen position differences, which ruin the self-attention mechanism. They also proved that attention scores behave nicely under position interpolation, where position differences can be unseen but within the range seen during training. Their proof can be applied to our RFS as well, guaranteeing the attention scores won't spike.

For the absolute sinusoidal encoding, let $\mathbf{P} \in \mathbb{R}^{n \times d}$ be the position matrix where i-th row $\mathbf{p}_i$ is the position vector defined as

$$\begin{aligned} \mathbf{p}_{i,2k} &= \sin(p_i/10000^{2k/d}) \\ \mathbf{p}_{i,2k+1} &= \cos(p_i/10000^{2k/d}). \end{aligned} \tag{1}$$

We argue that the simple extension fails because the rank of the position matrix increases as the input sequence gets longer, interfering with the subspace purely for the semantic information during training. Figure 5 clearly illustrates this, where we perform SVD on the position matrix and plot the singular values in descending order. We set $d = 256$ and choose 8 values for n. Note that the rank of a matrix is the same as the number of its non-zero singular values, and no position matrix in Figure 5 is a full rank. Considering that the position matrix is added to the token embedding $\mathbf{X} \in \mathbb{R}^{n \times d}$, the input to the self-attention block $\mathbf{X} + \mathbf{P}$ stores the position information only in the smaller subspace, and the rest only holds the semantic information. We observe that the simple extension increases the rank of the position matrix as the sequence length grows, which is problematic as it undermines the subspace in which the model assumes no position information is contained. This is not the case for our RFS.

Takeaway. In both RoPE and the absolute sinusoidal encoding, utilizing unseen position indices during training ruins the model because they are out-of-distribution, and random integer sampling and position interpolation overcome this issue by matching the position index range in training and inference. Thus, squeezing the indices into the range $[0, L]$ is the natural starting point for us. Further, inspired by the observation that position indices used in the inference must be seen during training, we design our position indexing strategy so that the model is forced to experience not only various position indices within the range but also diversified position distances where the extra randomness comes in. This differentiates our method from position interpolation and leads to the distinctive performance as demonstrated in Section 4.

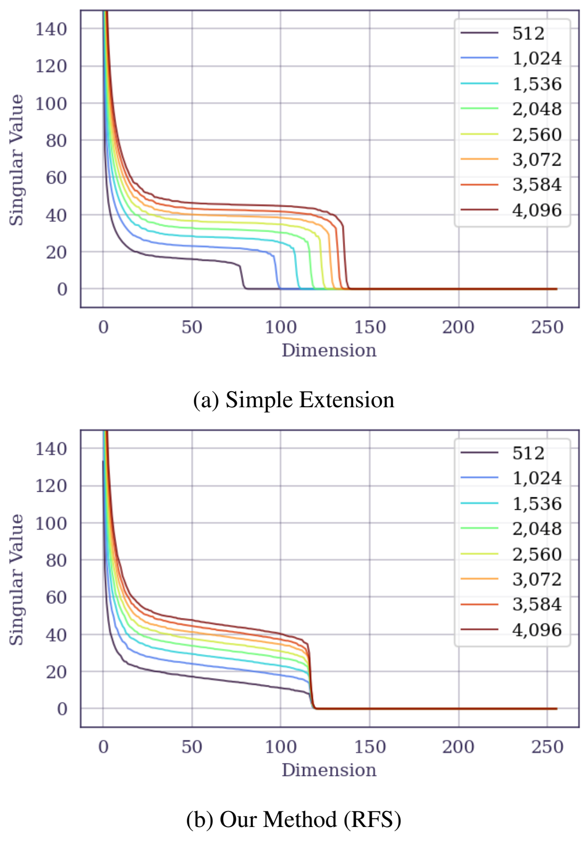


Figure 5: The singular values of the additive sinusoidal position matrix. The dimensionality is 256. The position index range is set to $[0, 2048]$ in our method (bottom). With the simple extension, the longer the input gets, the higher the rank of the position matrix becomes. RFS effectively controls the rank no matter the input length.

B Experiment Details

B.1 Code

The code for this research is available at https://anonymous.4open.science/r/RFS-arr-submission/.

B.2 Datasets

Table 3 provides a summary of datasets used in the experiments.

Name	Split	# training samples	# test samples	License
Copy	cmc_tr20_ts40	100,000	10,000	MIT License
Copy2	rsc_tr20_ts40	100,000	10,000	MIT License
Reverse	mc_tr20_ts40	100,000	10,000	MIT License
Addition	len_tr8_ts16	100,000	10,000	MIT License
Sort	len_mltd_tr8_ts16	100,000	10,000	MIT License
Summation	len_tr8_ts16	100,000	10,000	MIT License
SCAN	len_tr25_ts48	15,997	3,136	BSD License
OpenWebText	-	-	-	Creative Commons
HellaSwag	-	-	10,042	MIT License
RACE	-	-	1,045	-
ARC-e	-	-	2,376	Creative Commons
ARC-c	-	-	1,172	Creative Commons
OpenbookQA	-	-	500	-
WinoGrande	-	-	1,267	-
BoolQ	-	-	3,270	Creative Commons

Table 3: Dataset description.

Task Name	Input Example	Output Example
Copy	Copy: Code Bank southern 8	Code Bank southern 8
Copy2	Copy: Angle Angle Angle	Angle Angle Angle
Reverse	situated Con branches 8	8 branches Con situated
Addition	Compute 3 5 3 2 + 6 2 7 6 5 4 ?	The answer is 6 3 1 1 8 6.
Sort	Sort the following list [9 8 3 0, 5 1 5].	The sorted list is [5 1 5, 9 8 3 0].
Summation	Compute: 3 + 9 + 2 + 1 + 7 .	The answer is 2 .
SCAN	jump twice after run right	Right Run Jump Jump

Table 4: Examples of the input and output of the length generalization tasks.

B.3 Computational Resources

All model training is performed on eight Nvidia A100 (40GB) instances. For the length generalization tasks, one training run is completed in 4 GPU hours. For the language modeling, one training run is completed in 160 GPU hours.

B.4 Length Generalization Tasks

In Section 4.1, we evaluated our method on 7 tasks: copy, copy2, reverse, addition, sort, summation, and SCAN. The examples of them are provided in Table 4. Note that we ask to answer only one's place of the result in the summation task following Kazemnejad et al. (2023). Table 5 presents the model architecture and the training parameters. A hyperparameter L in our method, the range from which position indices are selected, is set to $1,000$ for absolute sinusoidal encoding and RoPE, while for ALiBi, we select the maximum input length of the entire dataset.

Parameter	**Value**
Architecture	
# Layers	12
# Attention Heads	12
Model Dimension	768
Rotary Dimension	16 (25% of the head dim)
Normalization	LayerNorm
Dropout	0.1
Decoding Strategy	Greedy
Hyperparameters	
Optimizer	AdamW
Learning Rate	3e-5
Weight Decay	0.05
β_1	0.9
β_2	0.999
Gradient Clipping	1.0
Batch Size	64
# Train Steps	40,000
# Warm Ups	2,400
Index Range L (ALiBi)	max input length
Index Range L (others)	1,000

Table 5: Summary of model architecture and the training parameters for the length generalization tasks.

B.5 Language Modeling Experiment Details

The architecture and the hyperparameters are detailed in Table 6. After being pretrained on OpenWebText (Gokaslan et al., 2019), the models are evaluated using the Language Model Evaluation Harness (Gao et al., 2024).

Parameter	**Value**
Architecture	
# Layers	12
# Attention Heads	12
Model Dimension	768
Position encoding	RoPE
Normalization	LayerNorm
Dropout	0.0
Hyperparameters	
Optimizer	AdamW
Learning Rate	6e-4
Weight Decay	0.05
β_1	0.9
β_2	0.999
Gradient Clipping	1.0
Batch Size	480
# Train Steps	100,000
# Warm Ups	2,000
Index Range L	1,000

Table 6: Summary of model architecture and the training parameters for the language modeling.

C Length Generalization Tasks Additional Results

This section provides a qualitative error analysis and two additional experiments to further clarify the advantages of the proposed method. The first experiment compares RFS to the performance of position interpolation and random integer sampling on the length generalization tasks. Then, an experiment of testing the compatibility of ALiBi and our method follows. The experimental settings are identical to those in Section 4.1.

C.1 Error Analysis

Table 7 reports validation set samples that both the baseline and RFS with RoPE predict incorrectly. A sample of the copy2 task is not included, as RFS made no errors. Though incorrect, these are not completely nonsense, and RFS answers tend to be closer to the target. For instance, in the copy task, RFS prediction has only one word error, while the baseline additionally skips two words. Another example in the sort task, RFS answers only one number incorrectly, but the baseline additionally forgets to generate a comma. It suggests that when other metrics other than the exact match are employed, the performance gap between the baseline and RFS may get larger. This motivates practitioners to equip RFS in a situation where minor errors are acceptable.

C.2 Position Interpolation and Random Integer Sampling

For position interpolation, we evaluate the performance without fine-tuning for fair comparison, even though Chen et al. (2023) states that minimal fine-tuning is necessary to fully leverage its architectural edge. In random integer sampling, the maximum length L is 512. We run the experiment 3 times for each setup and take the average accuracy for visualization.

The results are given in Figure 6. Overall, our method presents better length generalization ability in most of the tasks, although for the addition task, position interpolation exhibits a slightly better performance than ours. Random integer sampling often struggles to achieve high accuracy for problem lengths seen during training, implying that this method underfits and requires more training steps.

C.3 Random Float Sampling and ALiBi

ALiBi is designed to enhance the length generalization ability by not modifying the queries and keys in the attention mechanism but adjusting the attention scores according to the token distances before the softmax operator. Thus, it avoids the OOD issues discussed in A that are inherent to the absolute sinusoidal encoding and RoPE. However, attention score adjustments can be unseen if the input length exceeds the training context size, and it may cause performance degradation, which could be mitigated by incorporating our method.

This hypothesis is tested and reported in Figure 7 by adding two lines, ALiBi with our method and with the simple extension, to Figure 3. It clarifies that our method successfully adds performance gain to the simple extension in all the tasks, suggesting that experiencing possible attention score adjustments during training leads to better length generalization ability. That being said, the performance gain is smaller than that in the absolute sinusoidal encoding and RoPE, and it shows limited effectiveness than NoPE in copy2, reverse, addition, and SCAN.

D Use of LLM

We employed a large language model (ChatGPT; “GPT-5 Thinking”) only for English-language polishing and light copy-editing. The model was not used to generate ideas and experimental design. All technical content and claims were authored and verified by the human authors. No non-public data, confidential information, or personally identifiable information were provided to the model.

Copy | Copy: 1 southern I M J $ high Hersteller conform J B J F Positive heroic Men 3 A + 4 registered	
Target	1 southern I M J $ high Hersteller conform J B J F Positive heroic Men 3 A + 4 registered
Baseline	1 southern I M J $ high Hersteller conform J F J F Positive heroic Men 3 A registered
RFS	1 southern I M J $ high Hersteller conform J F J F Positive heroic Men 3 A + 4 registered

Reverse | 2 tears tears S neighborhoods Cover = geschaffen C apa tears 8 5 exp R 2 despair E (5 A branches 1	
Target	1 branches A 5 (E despair 2 R exp 5 8 tears apa C geschaffen = Cover neighborhoods S tears tears 2
Baseline	E 2 2 R 8 8 tears apa C geschaffen geschaffen = Cover neighborhoods neighborhoods tears tears 2 2 2
RFS	1 branches A 5 (E despair 2 R exp 5 8 tears apa C geschaffen = Cover neighborhoods S tears 2 2

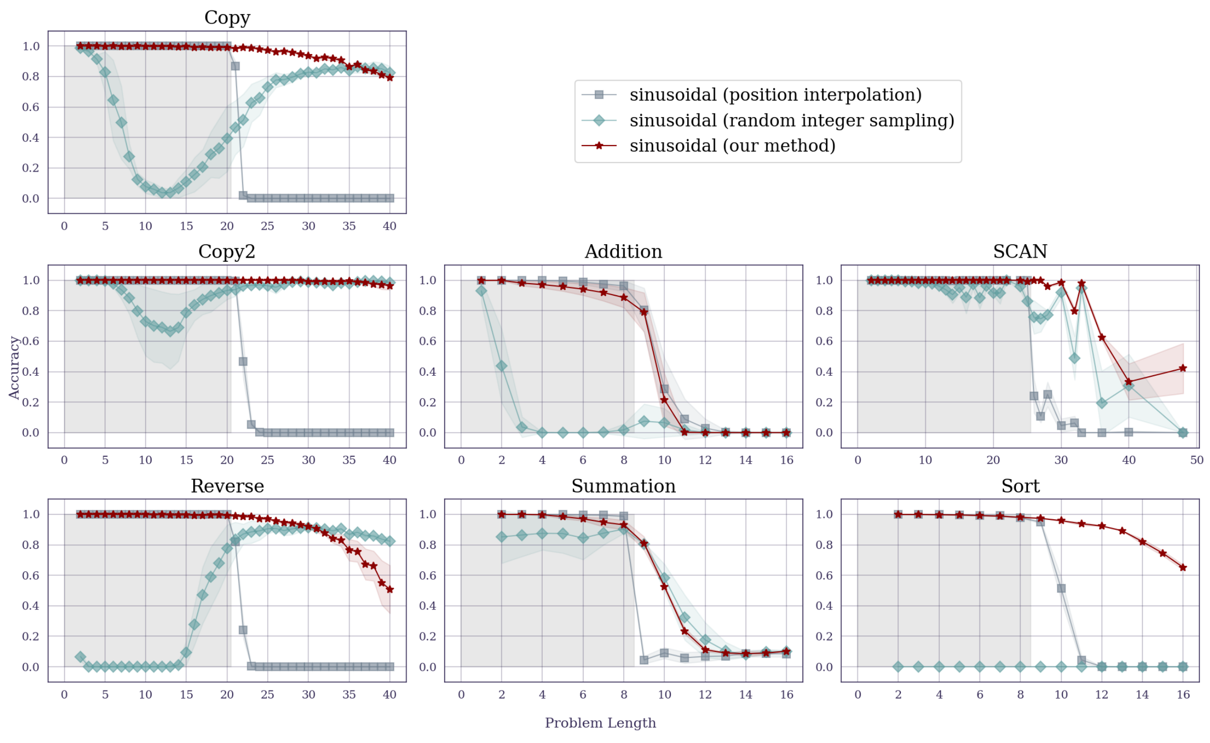
Addition | Compute 5 0 + 6 0 1 8 7 7 3 1 1 0 5 ?	
Target	60187731155
Baseline	6 0 1 8 7 7 6 5. 5. 5. 5.
RFS	6 0 1 8 7 7 3 1 5 5. 5.

Sort | Sort the following list [9 7 6 6, 4 3 5 9, 4 2 2 9, 1 4 4 4, 6 4 6 3, 7 1 4 5, 4 7 5 4, 4 5 5, 4 1 3 9, 4 0 4 5].	
Target	455, 1444, 4045, 4139, 4229, 4359, 4754, 6463, 7145, 9766
Baseline	4 5 5, 1 4 4 4, 4 0 4 5, 4 1 3 9, 4 2 2 9, 4 3 5 9, 6 7 5 4, 6 4 6 3, 7 1 4 5 9 7 6 6
RFS	4 5 5, 1 4 4 4, 4 0 4 5, 4 1 3 9, 4 2 2 9, 4 7 5 9, 4 7 5 4, 6 4 6 3, 7 1 4 5, 9 7 6 6

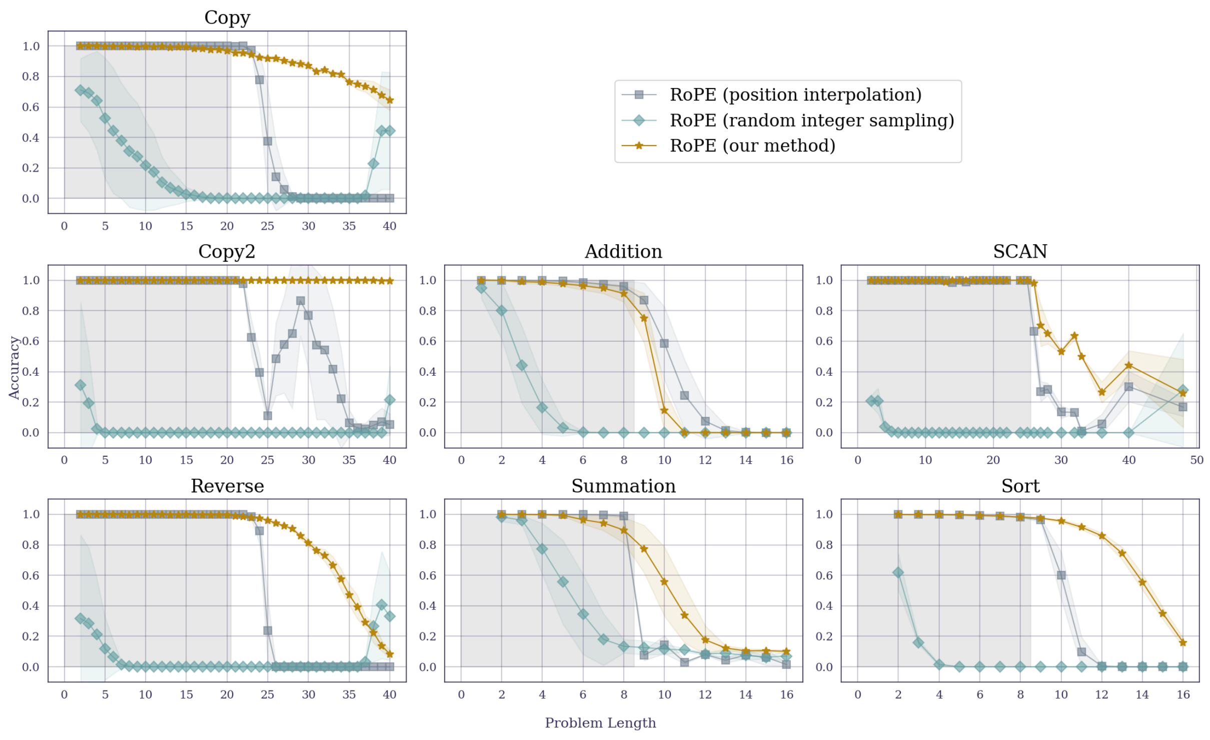
Summation | Compute: 2 + 4 + 7 + 9 + 9 + 4 + 2 + 9 + 2 .	
Target	8
Baseline	6
RFS	9

SCAN | look around right thrice and turn opposite right	
Target	Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Right
Baseline	Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Right Right Right
RFS	Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Look Right Right Right Right

Table 7: Typical errors on algorithmic tasks.



(a) Sinusoidal



(b) RoPE

Figure 6: The results of position interpolation without fine-tuning, random integer sampling, and our method on the length generalization tasks.

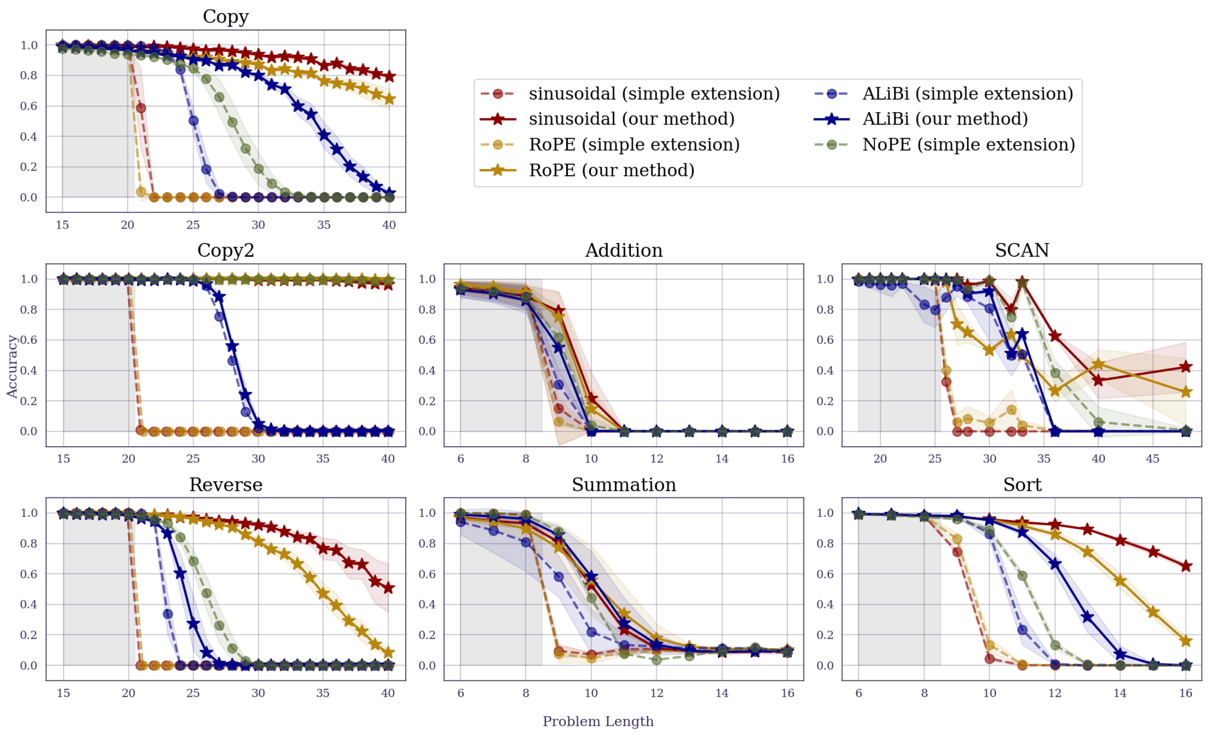


Figure 7: The impact of the proposed method on ALiBi (blue lines) on the length generalization tasks.