Understanding Attention Glitches with Threshold Relative Attention

Anonymous Authors[1]

Abstract

Transformers struggle with generalisation, displaying poor performance even on basic yet fundamental tasks, such as flip-flop language modeling. We test whether these limitations can be explained through two key failures of self-attention. The first is the inability to fully remove irrelevant information. The second concerns position, even when a key is completely irrelevant learned positional biases may unintentionally up-weight it - dangerous when distances fall out of distribution. To probe this we propose TRA, a novel attention mechanism with which we demonstrate that these issues underlie generalisation failures on the flip-flop task.

1. Introduction

Transformers (Vaswani et al., 2017), pre-trained at scale, display revolutionary broad-spectrum capabilities, particularly in the domains of natural language and code (Bubeck et al., 2023), and include the ability to perform complex abstract symbol manipulation entirely in-context (Brown et al., 2020; Smolensky et al., 2024). However, when tested in controlled synthetic settings transformers demonstrate dramatic generalisation failures even on tasks where the correct solution *should be trivial* (Zhou et al., 2023). A particularly notable example of this is the task of Flip-Flop language modeling (Liu et al., 2023a), which requires storing a retrieving a single bit of information according a to an instruction sequence. While trivial, this capability is foundational for all syntactic parsing and algorithmic reasoning capabilities, and yet is one which transformers fail to learn without a long tail of generalisation errors. Our contention is that generalisation errors stem from the following flaw in self-attention. The mechanism is able to perform (1) fully positional operations (i.e. attention within a local window) (2) fully content based operations (ignoring position entirely), but not (3): content followed by position, e.g. of some set of relevant items apply a positional bias. We believe (3) is a cruical capability - it allows for tracking the state of an entity over time, and is a pre-requisite for robust reasoning. However it is absent from standard attention because position is generally treated as independent of content. Which leaves the two types of information in conflict, fundamentally limiting self-attention. Our solution involves a two step process. We first mask out irrelevant keys based on threshold applied to the raw attention weights, and then compute the relative distance but only as between the query and the keys that remain. We call this mechanism **T**hreshold **R**elative **A**ttention (**TRA**), and show that it can generalisably solve the flip-flop task.

2. Model

2.1. Background: Relative Positional Encodings

Relative positional encodings (Dai et al., 2019; Raffel et al., 2019) decompose the attention operation into two steps. First the standard $\boldsymbol{QK}^T$ dot product is calculated, but this purely based on semantic relevance. Let's call that matrix $\boldsymbol{S}$ (semantic). This is then added with a learned positional bias matrix $\boldsymbol{D}$ (distance) to produce the final attention weight.

$$\boldsymbol{S} = \frac{\boldsymbol{QK}^T}{\sqrt{\boldsymbol{d}_k}} \tag{1}$$

$$\boldsymbol{A} = \text{softmax}\left(\boldsymbol{S} + \boldsymbol{D}\right) \tag{2}$$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Our contention is that a central cause of length generalisation is due to $\boldsymbol{S}$ and $\boldsymbol{D}$ being treated as independent of each other. If $\boldsymbol{D}$ is fit to training distances, then distribution shifts will necessarily begin to introduce errors. However, $\boldsymbol{S}$ is more robust, because relevance isn't impacted by changing lengths. We want to make $\boldsymbol{D}$ more robust by making it contingent on $\boldsymbol{S}$.

2.2. Contextual Relative Distance

We want to condition $\boldsymbol{D}$ on $\boldsymbol{S}$. To achieve this we first introduce **selective sparsity**, allowing keys to be fully removed from $\boldsymbol{S}$ by using ReLU as a **threshold** and which eliminates keys with negative weights. Lets call that matrix $\boldsymbol{S'}$.

$$\boldsymbol{S'} = \text{ReLU}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{\boldsymbol{d}_k}}\right) \tag{3}$$

Next calculate a boolean mask $\boldsymbol{M}$ that is true only for relevant positions:

$$M_{i,j} = \begin{cases} 1 & \text{if } S'_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We are now going to compute our distance matrix, but only consider the relative distance to keys which make it through the threshold, this give us the **contextualised relative distance**. To do this we consider the following example $\boldsymbol{M}$:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{5}$$

Our threshold mask $\boldsymbol{M}$ is a $m \times n$ matrix, which we are going to use to get our contextualised relative distance matrix $\bar{\boldsymbol{D}}$, by applying the cumsum operation in the right to left direction:

$$\bar{D}_{ij} = \sum_{k=j}^{n} M_{ik} \tag{6}$$

For our example $\boldsymbol{M}$ from (5) this would yield the following output, for clarity we omit keys that do not meet the threshold:

$$\bar{\boldsymbol{D}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 0 \end{bmatrix} \tag{7}$$

$\bar{\boldsymbol{D}}$ now contains distances which are informed by context. For example, the query at position 4 it will view the key at position three as the closest possible item and the key at position 1 as the next nearest, skipping over the irrelevant intervening keys.

2.3. Parameterising $\bar{D}$

We need some way of turning the contextualised relative distances into an actual weight. In order to do so we utilise a parameterised forget gate. Lets denote the residual stream of size *E* for position *i* as x_i, with $\boldsymbol{W}_f \in \mathbb{R}^{E\times 1}$ as the forget projection, b as a scalar bias, and σ as the sigmoid non-linearity. Then the weight for each position is given as:

$$\delta_i = \sigma\left(W_f x_i + b\right) \tag{8}$$

$$D'_{ij} = \delta_i^{\bar{D}_{ij}} \tag{9}$$

As each δ_i is a sigmoided scalar value, the more times it is multiplied with itself the smaller it gets. The closer δ_i is to 1 the longer the decay takes, the closer to zero the shorter. This provides the model a temporal memory. Enabling it to forget irrelevant past information if necessary, or to treat all timesteps uniformly if temporal ordering is not a concern. We opt for a recency bias for several reasons. Firstly, it has been generally shown to be useful when modeling natural language (Shen et al., 2018; Press et al., 2021; Zhu et al., 2021; Clark et al., 2025). Secondly, it is unclear where position provides useful information beyond causal ordering. The first occurrence can be identified by proximity to the start of sequence token, and middle positions are better differentiated via their unique semantic context (Ebrahimi et al., 2024). Finally, decaying memory is the mechanism empowering the resurgence in state space models (Peng et al., 2023; Orvieto et al., 2023; De et al., 2024), recursive networks (Opper & Siddharth, 2024), and has shown promise with transformers (Csordás et al., 2021).

2.4. The Final Attention Weights

To complete the TRA mechanism, we compute the final attention weights $\boldsymbol{A'}$ as follows:

$$\hat{\boldsymbol{A}} = \boldsymbol{S'} + \log(\boldsymbol{D'}) \tag{10}$$

$$\bar{A}_{ij} = \begin{cases} -1e11 & \text{if } M_{ij} = 0 \\ \hat{A}_{ij} & \text{otherwise} \end{cases} \tag{11}$$

$$\boldsymbol{A'} = \text{softmax}\left(\bar{\boldsymbol{A}}\right) \tag{12}$$

We use $\log(\boldsymbol{D'})$ for improved numerical stability and weight scale following (Lin et al., 2025). Note keys which do not meet the threshold are completely removed from the final softmax and do not count towards distance. This means that TRA exhibits both selective sparsity and allows semantic content to determine, and consequently synergise with, positional bias.

3. Experiments

3.1. Flip-Flop Language Modelling

As stated in the introduction, Flip-Flop language modeling Liu et al. (2023a) is a algorithmic reasoning benchmark designed to test transformers ability to glitchlessly handle sequential dependencies. Sequences consist of symbol alphabet of three instructions: write (w), ignore (i) and read (r) each of which is followed by a bit. To solve the task the model has to recall the bit that follows the nearest write instruction while ignoring all irrelevant information (e.g. in: w1i0i1i1i0i1i0i0i1i0r out: 1). The model cannot rely on memorisation but instead has to learn a program to succeed, and achieving this capacity is a pre-requisite to being able to maintain a state hierarchically (Merrill et al., 2022; Liu et al., 2023b). The task consists of three test sets: IID, and two out of distribution sets. OOD sparse increases the number of ignore instructions and requires the ability to handle increased dependency distance. OOD dense decreases the number of ignore instructions and tests if the model retain focus in the presence of an increased number of attractors (i.e. write instructions). Input examples consist of strings of length 512 with the probability of an ignore instruction being generated set to 0.98 in the sparse set and 0.1 in the dense set. The criterion for success is perfection (i.e. 100% accuracy across all test sets). The authors find that transformers exhibit reasoning errors across many architectural variants and that the issue is *scale-invariant*, persisting even up to GPT-4.

3.2. Baselines and Experimental Setup

For both TRA and the baselines we use the same core transformer backbone based on Llama 3 (Dubey et al., 2024); consisting on SwiGLU layers (Shazeer, 2020) and RMSNorm (Zhang & Sennrich, 2019). For all models we use the mini configuration from Turc et al. (2019): four heads, four layers and a 256 dimensional embedding size. The MLP is set 2x embedding size for both the linear and gate hidden units. Our focus is on small models following prior work (Zhou et al., 2023). Furthermore, attention glitches have been shown to persist at scale (Liu et al., 2023a), and the true solution for these tasks should not require additional layers. We use cosine decay for all models. We compare TRA with:

No positional encoding (NoPE): Our first baseline is causal attention with no positional embedding, following Kazemnejad et al. (2023)'s claim that decoder only transformers can implicitly represent position so explicitly encoding is unnecessary.
Absolute positional encoding (APE): Positional information for each index is represented by a learned embedding which is added to the residual stream at layer zero. This was the standard in the original variant (Vaswani et al., 2017) as well as GPT-2 (Radford et al., 2019). Lengths not encountered during training will not have the corresponding embeddings trained.
Relative Positional Encodings (REL): Introduced by Transformer-XL (Dai et al., 2019) and also adopted by T5 (Raffel

et al., 2019). Relative Positional Encodings model position via a learnable additive bias term. Position is considered as to the relative distance of key to the query. OOD distances are all assigned the same learned value for maximum length.
Rotational Positional Encoding (RoPE): Introduced by Su et al. (2021). This approach models positional information by applying a rotation to the keys proportional to the distance from the query. Recent work by Barbero et al. (2025) demonstrates that rather than encoding a gradual distance based decay RoPE actually learns either fully positional attention (attend to predecessor or attend to first token) or full semantic attention. One coming at the expense of the other.
Label Encoding (Label): Introduced by Li & McClelland (2023) and Ruoss et al. (2023) these seek to mitigate the OOD issues faced by APE by randomly selecting and then sorting indices from a range greater than sequence length. This motivated by the contention that encodings learn to respect relative ordering rather than absolute value.
Contextual Positional Encoding (CoPE): Perhaps conceptually the most related to TRA, CoPE (Golovneva et al., 2024) utilises spikes in the $\boldsymbol{S}$ matrix in order to calculate positions. Tokens are assigned fractional positions by interpolating betweeen a fixed set of absolute embeddings. Unlike TRA no thresholding is employed to remove irrelevant keys.
Forgetting Transformer (FoT): The recent Forgetting Transformer (FoT) (Lin et al., 2025) utilises a forget gate as its positional encoding. The crucial difference being that it uses standard relative distance compared with TRA's contextualised alternative. FoT therefore serves as the truest test as to whether the issues we hypothesise regarding position are valid.
Differential Transformer (Diff): Introduced by Ye et al. (2024) the differential transformer utilises a twin heads approach where an auxiliary attention head looks to perform noise reduction by down-weighting attention to irrelevant keys. It tests to whether increased specificity in the attention dot product is beneficial - without any further consideration to position.

3.3. Results

Table 1. Results represent the average across four random initialisations. Metric is full sequence accuracy (exact match).

Model	NoPE	APE	REL	RoPE	Label	FoT	Diff	CoPE	TRA
Flip Flop IID	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$
Flip Flop OOD (Dense)	0.15 ± 0.0	27.38 ± 13.1	41.2 ± 47.29	$\mathbf{100 \pm 0.0}$	12.58 ± 15.87	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$	$\mathbf{100 \pm 0.0}$
Flip Flop OOD (Sparse)	99.97 ± 0.1	90.61 ± 5.64	70.48 ± 10.92	72.82 ± 1.27	86.93 ± 9.93	93.4 ± 4.21	77.8 ± 6.73	95.1 ± 4.4	$\mathbf{100 \pm 0.0}$

Most PE methods struggle OOD: Absolute and relative positional embeddings consistently struggle in out of distribution settings. Positional information is attached to specific indices. Best case they learn a bias in a particular direction (recency or the opposite), but even then this bias is input solely index dependent and only applicable for IID indices. Consequently, they struggle with OOD settings. RoPE also appears to learn a strategy which is fit to the training distribution and therefore struggles with out of distribution settings. This issue extends even to the differential transformer, which despite having a more flexible selection mechanism is still limited by its underlying positional encoding (RoPE).
NoPE Generalises Weakly: The NoPE hypothesis (Kazemnejad et al., 2023) states that decoder only transformers do not need positional embeddings because they can approximate a counter. They do so by selecting a token which serves as an anchor and sends some signal via the value message. Intervening tokens send a null signal. The position of a given token is then determined by the extent to which the anchor token's signal is diminished by the softmax. From Table 1 we can see that this strategy generalises quite well on the sparse setting, but totally collapses on the dense OOD set. This indicates that too many anchor signals lead to errors in the count, which means that NoPE is insufficient for true generalisation.
Flexible PE Yields Improvements but is Insufficient: The strongest contenders of the baselines are FoT and CoPE, which utilise more flexible positonal encodings. However, both models fail to learn consistent generalising solutions to the task. FoT lacks the capability to synergise position and content which invitably leads to failures (see A), while CoPE has the flexibility, but cannot fully remove irrelevant information which makes the generalising circuit harder to form.
TRA succeeds: It is fully able to solve the benchmark because it can combine positional and semantic information in a complementary fashion. It achieves 100% accuracy across seeds, which till this point has been only been achievable by LSTM networks. We provide mechanistic analysis as to how it succeeds while other baselines fail in Appendix A.

4. Conclusion

In this work we investigate whether fundamental issues with length generalisation occur due to a conflict between positional and semantic information. As a probe for this hypothesis we present TRA, a simple modification to the attention mechanism designed to allow both types of information to operate in tandem. We find that the introduction of such a change significantly improves out of distribution generalisation. Finally, we present some preliminary results assessing TRA's suitability for language in modeling in Appendix B and find evidence of promising generalisation trends.

References

Barbero, F., Vitvitskyi, A., Perivolaropoulos, C., Pascanu, R., and Veličković, P. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GtvuNrk58a.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL https://arxiv.org/abs/2303.12712.

Clark, C., Oh, B.-D., and Schuler, W. Linear recency bias during training improves transformers' fit to reading times. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S. (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7735–7747, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.517/.

Csordás, R., Irie, K., and Schmidhuber, J. The neural data router: Adaptive control flow in transformers improves systematic generalization. *CoRR*, abs/2110.07732, 2021. URL https://arxiv.org/abs/2110.07732.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860, 2019. URL http://arxiv.org/abs/1901.02860.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., Desjardins, G., Doucet, A., Budden, D., Teh, Y. W., Pascanu, R., Freitas, N. D., and Gulcehre, C. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ebrahimi, M., Panchal, S., and Memisevic, R. Your context is not an array: Unveiling random access limitations in transformers, 2024. URL https://arxiv.org/abs/2408.05506.

Golovneva, O., Wang, T., Weston, J., and Sukhbaatar, S. Contextual position encoding: Learning to count what's important, 2024. URL https://arxiv.org/abs/2405.18719.

Kazemnejad, A., Padhi, I., Natesan, K., Das, P., and Reddy, S. The impact of positional encoding on length generalization in transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Drrl2gcjzl.

Li, Y. and McClelland, J. Representations and computations in transformers that support generalization on structured tasks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=oFC2LAqS6Z.

Lin, Z., Nikishin, E., He, X., and Courville, A. Forgetting transformer: Softmax attention with a forget gate. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=q2Lnyegkr8.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Exposing attention glitches with flip-flop language modeling, 2023a. URL https://arxiv.org/abs/2306.00946.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=De4FYqjFueZ.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016. URL http://arxiv.org/abs/1609.07843.

Merrill, W., Sabharwal, A., and Smith, N. A. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022. doi: 10.1162/tacl_a_00493. URL https://aclanthology.org/2022.tacl-1.49/.

Opper, M. and Siddharth, N. Banyan: Improved representation learning with explicit structure. *arXiv preprint arXiv:2407.17771*, 2024.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences, 2023.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., and Zhu, R.-J. Rwkv: Reinventing rnns for the transformer era, 2023.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *CoRR*, abs/2108.12409, 2021. URL https://arxiv.org/abs/2108.12409.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

Ruoss, A., Delétang, G., Genewein, T., Grau-Moya, J., Csordás, R., Bennani, M., Legg, S., and Veness, J. Randomized positional encodings boost length generalization of transformers, 2023. URL https://arxiv.org/abs/2305.16843.

Shazeer, N. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020. URL https://arxiv.org/abs/2002.05202.

Shen, Y., Lin, Z., wei Huang, C., and Courville, A. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkgOLb-0W.

Smolensky, P., Fernandez, R., Zhou, Z. H., Opper, M., and Gao, J. Mechanisms of symbol processing for in-context learning in transformer networks. *arXiv preprint arXiv:2410.17498*, 2024.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864, 2021. URL https://arxiv.org/abs/2104.09864.

Turc, I., Chang, M., Lee, K., and Toutanova, K. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019. URL http://arxiv.org/abs/1908.08962.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Ye, T., Dong, L., Xia, Y., Sun, Y., Zhu, Y., Huang, G., and Wei, F. Differential transformer, 2024. URL https://arxiv.org/abs/2410.05258.

Zhang, B. and Sennrich, R. Root mean square layer normalization. *CoRR*, abs/1910.07467, 2019. URL http://arxiv.org/abs/1910.07467.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S., and Nakkiran, P. What algorithms can transformers learn? a study in length generalization, 2023. URL https://arxiv.org/abs/2310.16028.

Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. *CoRR*, abs/2107.02192, 2021. URL https://arxiv.org/abs/2107.02192.

A. Mechanistic Analysis

In order to analyse the impact of contextualised relative distance we train two layer one head TRA and FoT models on the flip-flop language modeling task and compare the attention heatmaps for the final layer in Figure 3. TRA fully generalises in this limited setting, the only model to be able to do so out of all the baselines, and consistent with the optimal solution demonstrated in theory by (Liu et al., 2023a). On the other hand FoT fails, and the reason why is clearly visible by contrasting the attention weights of the two.

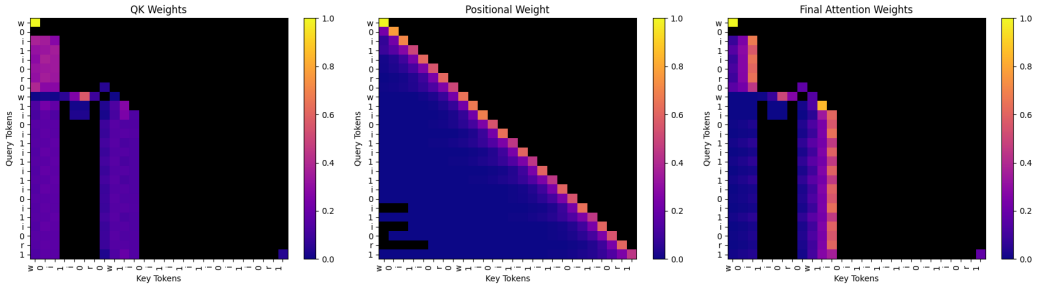

Figure 1. TRA final layer attention in FFLM task. Black means null attention weight.

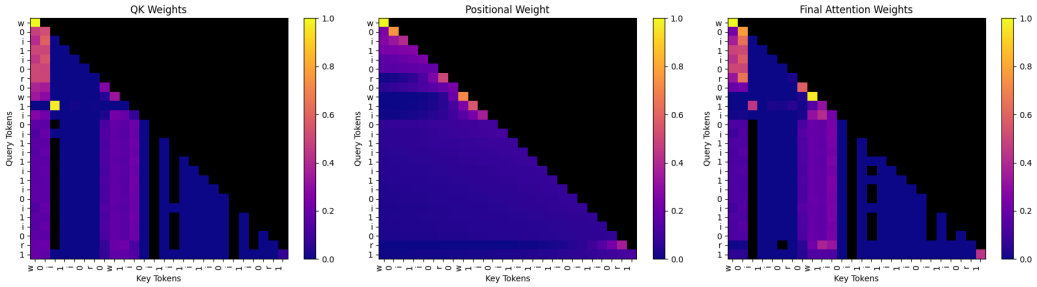

Figure 2. FoT final layer attention in FFLM task. Black means null attention weight.

Figure 3. Contrasting Attention Heatmaps between TRA and FoT. TRA synergises semantic and positional information while FoT must trade one for the other.

Figure 1 shows how TRA is able to combine both semantic content with relative positions. It first filters out irrelevant keys fully and is then able to apply a strong positional bias to the remainder that truly matter, allowing it to cleanly select the most recent write instruction. Its recency bias (1 centre) is applied in a shifted manner only to what is relevant, which allows it express a strong positional preference. FoT on the other hand (2), has no such mechanism and as a result must diminish its recency bias to be able to focus on semantic content (contrast the strength of 2 centre with 1 centre). This fact, coupled with its inability to completely remove irrelevant tokens, means that it is unable to handle long distance dependencies and consequently cannot learn semantic and positional preference such that they complement each other. We use FoT here as a clean illustrative example, but note that the same failure mode *must* be true of all schemes that do not employ contextualised distance.

B. Language Modeling:

TRA enables improved generalisation in controlled synthetic settings, which we attribute to its capacity for contextualised relative distance. What happens when we expose it to more complex data like natural language? Investigating this question has two core motivations. Firstly, contextualised relative distance requires selective sparsity to operate. This operation cuts the connection between columns and we must make sure that this does not harm learning in more complex domains by potentially limiting exploration. Secondly, it allows us to investigate whether the generalisation patterns observed in synthesis also occur in natural language. For our experiments we turn to the WikiText-103 benchmark (Merity et al., 2016), which consists of full Wikipedia articles comprising circa 100 million tokens, probing both scale and ability to handle long distance dependencies.

Hyperparameters: For language modelling we increase model size to the medium configuration of (Turc et al., 2019). This is eight layers, eight heads and a 512 dimensional embedding size. The MLP is set 2x embedding size for both the linear and gate hidden units as before. We use the GPT-2 tokenizer (Radford et al., 2019) which leads to a vocabulary size of roughly 52k. Totaling circa 80 million parameters for both TRA and the baselines. We train using: window size 128, batch size 64, 100k steps. Our scheduling regime remains the same as before. Our evaluation consist of two parts. In distribution: we measure perplexity on the test set using the training window of size of 128. Out of distribution: we increase the window size to 4096 and observe the extent to which perplexity changes with increased length.

Table 2. Test Perplexity on WikiText-103 (↓). Mean and standard deviation taken over four random initalisations. Models are trained for 100k steps with a context window of 128 tokens.

Model	NoPE	APE	Rel	RoPE	Label	FoT	Diff	CoPE	TRA
Perplexity	31.74 ± 0.11	31.61 ± 0.16	31.27 ± 0.11	30.11 ± 0.04	32.09 ± 0.07	30.30 ± 0.04	29.46 ± 0.12	30.20 ± 0.09	30.04 ± 0.32

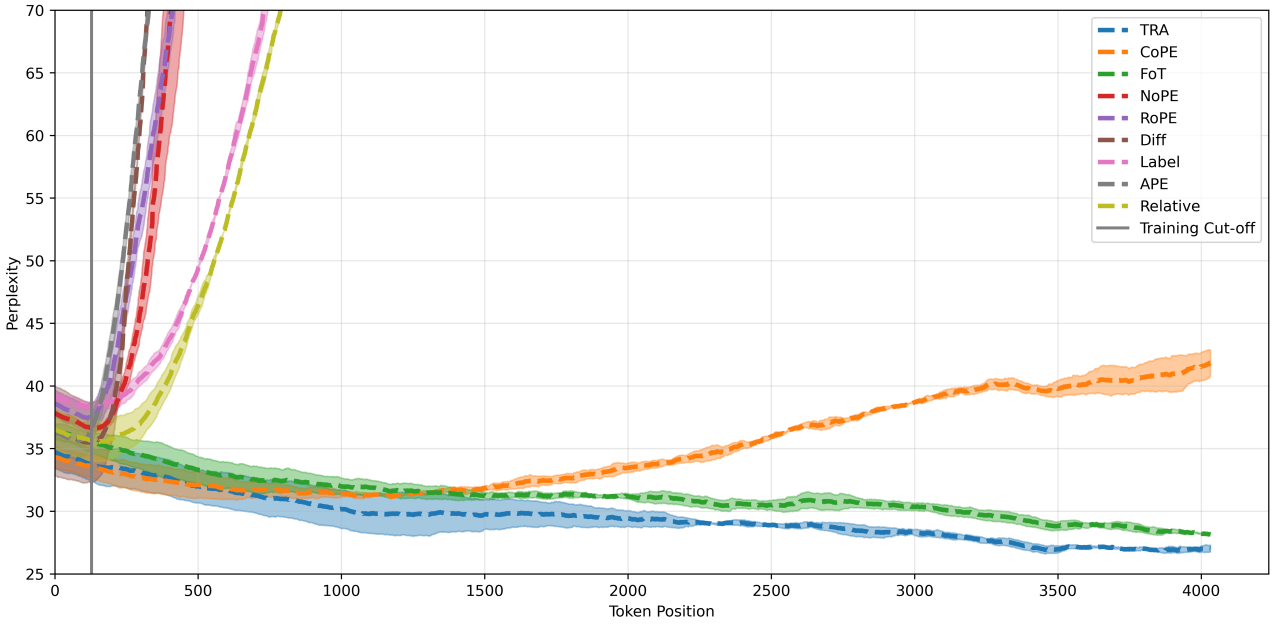

Figure 4. WikiText-103 Test Perplexity (↓) on OOD sequences lengths. Models trained for 100k steps with a window size of 128, and evaluated with window size up to 4098. Results taken over four random seeds.

Results: In-distribution results are shown in table 2. TRA remains competitive with the strongest baselines. This means that we can introduce selective sparsity without hurting learning. Notably the strongest performing model *in-distribution* is the differential transformer (Ye et al., 2024), which contains a more sophisticated mechanism for relevancy determination. Replacing the simple ReLU we use here with such a more sophisticated approach could prove a fruitful avenue for future work. The out of distribution case is shown in figure 4. Most PE methods very rapidly begin to decline in quality, with similar degradation patterns emerging compared to the synthetic settings. The three chief exceptions to this rule are TRA, CoPE and FoT. CoPE is able to maintain/slightly improve perplexity for up to roughly a 10x increase in context length, before performance begins to deteriorate. We believe this is because its use of fractional positions only temporarily alleviates generalisation difficulties, but ultimately leaves the core issue unaddressed. Furthermore, increasing the number of positions in CoPE may delay degradation further, but any increase comes with a heavy computational burden and consequently makes CoPE difficult to scale as the sole mechanism for representing position. On the other hand, both FoT and TRA are capable of strong length generalisation (up to 32x greater than training length). Moreover, they display a promising trend of improved perplexity with increased length which indicates that they are able to *make use of* increased context rather than simply being robust to it. Between the two, TRA demonstrates a consistent perplexity edge compared with FoT. We believe this is due to its improved capacity for handling out of distribution dependency lengths as demonstrated by the flip-flop language modeling results. In sum, we show that TRA can introduce selective sparsity to the attention weights without compromising

performance, and that its improved generalisation capabilities in synthetic settings also appear to be mirrored in the more complex setting of natural language.