THE MANY FACES OF SHORT CONVOLUTIONS IN TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Short causal convolutions improve the performance of sequence models, a benefit often attributed to the formation of single-layer induction heads. By disentangling the scaling and shifting operations within a convolution, we investigate their role from a **stability** perspective, showing that their benefits extend beyond expressivity. Starting from a simple channel-wise gain, we discover that convolutions mitigate rank-collapse by functioning as learnable **residual scaling**. However, this is not the only benefit for trainability. By extending the filter size, we show that convolutions alter the **positional information** learned through causal masks (NoPE), improving representational diversity. This enables training an otherwise unstable post-norm Transformers, while broadening the range of viable learning rates for pre-norm Transformers. Together, these findings identify channel-wise scaling and local temporal mixing as complementary contributors to the effectiveness of short convolutions.

1 INTRODUCTION

Short convolutions have proven useful as an addition to sequence models (Gu & Dao, 2023; Dao & Gu, 2024; Yang et al., 2025; Allen-Zhu, 2025; Li et al., 2025; Sieberling et al., 2026). While originally short convolutions used to serve as the main component for sequence mixing (Bai et al., 2018; Dauphin et al., 2017), their impact as auxiliary sequence mixers on linear RNNs and Transformers has proven to be substantial (Allen-Zhu, 2025). In the Mamba family (Gu & Dao, 2023; Dao & Gu, 2024; Lahoti et al., 2026) of sequence mixers and in DeltaNet (Yang et al., 2024; 2025), convolutions are usually placed on the query, key, and value projection, while later work showed further benefits by putting convolutions on the MLP projections as well (Qiu et al., 2025). In Transformers, on the other hand, putting convolutions on the query/key/value or the MLP projections is less established, with recent works showcasing benefits from all positions (Allen-Zhu, 2025; Li et al., 2025; Sieberling et al., 2026). In this paper, we primarily focus on Transformers, given their enduring popularity as sequence mixers in LLMs (Team, 2024).

However, most of the analysis on the role of convolutions in sequence models is limited to expressivity (Arora et al., 2024a;c; 2025; Okpekpe & Orvieto, 2025), and mainly focused on linear RNNs (Li et al., 2025). In particular, the impact of convolutions on the optimization and inductive biases of sequence models is yet to be studied. Convolutions combine two operations, namely **channel-wise scaling** and **token shifting**. Each of these functions may provide separate benefits. Specifically, scaling impacts the learning rate of architecture components, introducing complicated and cyclic optimization dynamics, while keeping expressivity the same (Salimans & Kingma, 2016; van Laarhoven, 2017). On the other hand, token shifting introduces local mixing in the model, which can introduce position-related inductive biases to the representation (Islam et al., 2020). Consequently, in this paper we focus on separating these two dynamics, providing an analysis on the benefits of convolutions in sequence models beyond expressivity.

Optimization benefits. We first investigate how channel-wise scaling contributes to the optimization benefits of convolutions. In our experiments, we find that scaling alone, implemented through a convolution with kernel size $k = 1$, is sufficient to make post-norm Transformers trainable. We observe further improvements in the form of learning-rate robustness in pre-norm Transformers. We connect these benefits to improved representational diversity, which can be achieved through resid-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ual scaling (Noci et al., 2022). We also provide an analysis of the optimization trajectory of these models, while showing that initialization matters in how these benefits exhibit themselves.

Positional information benefits. We then investigate the additional role of token shifting in how Transformers represent positional information. Larger convolution kernels generally improve performance beyond the training context length, motivating a closer examination of their role in positional encoding. We first show that queries do not benefit from the token shifting functionality, while the presence of explicit positional encoding in the form of RoPE reduces the use of token shifting in the value and key. We provide a theoretical analysis for this observation. Furthermore, we show that both larger kernels and RoPE reduce reliance on position-associated directions identified in the residual stream. Together, these observations suggest that convolutions provide an alternative means of representing some positional information.

Contributions. In this work, we provide the following contributions

- **Channel-wise scaling and optimization.** We show that primarily due to channel-wise scaling, convolutions make post-norm Transformers trainable, while significantly improving learning rate robustness in pre-norm Transformers.
- **Token-shifting and positional encoding.** We show that convolutions significantly improve length generalization, a property related to positional encoding. We provide evidence for convolutions at least partially altering the way Transformers encode positional information because they exhibit reduced reliance on the identified position-associated directions.
- **Theoretical result.** We show that short convolutions on keys and values provide order sensitivity, which cannot be achieved by queries through a finite size convolution.
- **Kernel-size and initialization.** We show that initialization plays a significant role in how the model interacts with layer normalization, while kernel size produces task-dependent benefits in recall, downstream performance, and length generalization.

2 PRELIMINARIES

A simple transformer block. This work considers a transformer with with B blocks, each comprising of an attention layer and a SwiGLU MLP, giving a total depth of $L = 2B$. Since attention mixes representations along the sequence axis, it is commonly referred to as a sequence mixer. The MLP, on the other hand, mixes representations across channels and is thus called a channel mixer.

We index input sequences by $n \in \{1, \dots, N\}$, each of length T, denoting the representation of token t of sample n after layer ℓ as $\mathbf{x}_{t,n}^{\ell} \in \mathbb{R}^D$. We furthermore denote by $\mathbf{X}_n^{\ell} \in \mathbb{R}^{T,D}$ the matrix obtained by stacking all layer-ℓ representations.

In this work, we study both pre-norm and post-norm architectures. In the pre-norm architecture, normalization is applied before the residual branch:

$$\mathbf{X}_n^{\ell} = \mathbf{X}_n^{\ell-1} + \left[F_{\ell}\left(\mathrm{Norm}_{\ell}\left(\mathbf{X}_n^{\ell-1}\right)\right)\right]_t,$$

where Norm_{ℓ} denotes RMSNorm. In the post-norm architecture, normalization is applied as:

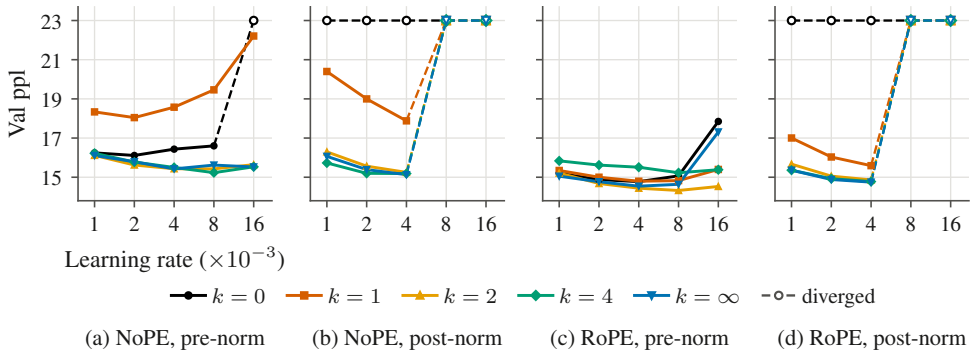
$$\mathbf{X}_n^{\ell} = \mathrm{Norm}_{\ell}\left(\mathbf{X}_n^{\ell-1} + \left(F_{\ell}\left(\mathbf{X}_n^{\ell-1}\right)\right)_t\right).$$

Above, F_{ℓ} can denote either causal self attention or a SwiGLU MLP. For the simple case of a single attention head, F_{ℓ} takes the form

$$F_{\ell}(\mathbf{X}_n^{\ell}) = \mathrm{softmax}\left(\mathbf{Q}_n^{\ell}\mathbf{K}_n^{\ell^{\top}}/\sqrt{d_h} + \mathbf{M}\right)\mathbf{V}_n^{\ell}\mathbf{W}_O^{\ell},$$

where $\mathbf{M}$ is the causal mask, $\mathbf{Q}_n^{\ell} = \mathbf{X}_n^{\ell}\mathbf{W}_Q^{\ell}$, $\mathbf{K}_n^{\ell} = \mathbf{X}_n^{\ell}\mathbf{W}_K^{\ell}$, and $\mathbf{V}_n^{\ell} = \mathbf{X}_n^{\ell}\mathbf{W}_V^{\ell}$; $\mathbf{W}_O^{\ell}$ is the output projection. The SwiGLU channel mixer instead takes the form $F_{\ell}(\mathbf{X}_n^{\ell}) = \mathbf{W}_d\left(\mathrm{SiLU}(\mathbf{W}_g\mathbf{X}_n^{\ell}) \odot \mathbf{W}_u\mathbf{X}_n^{\ell}\right)$, where $\mathbf{W}_g$ and $\mathbf{W}_u$ are the gate- and up projections, $\mathbf{W}_d$ projects back to the residual-stream dimension D, and $\odot$ denotes elementwise multiplication.

Convolutions. We investigate how per-channel causal convolutions, which mix tokens along the sequence axis within each channel, interact with the pre- and post-norm transformers introduced



(a) NoPE, pre-norm (b) NoPE, post-norm (c) RoPE, pre-norm (d) RoPE, post-norm

Figure 1: **Learning rate curves and kernel sizes.** Validation perplexity vs. learning rate for Transformers with and without rotary position embeddings, under pre-norm and post-norm, for various convolution kernel sizes $k \in \{0, 1, 2, 4, \infty\}$. The values $k = 0$ and $k = \infty$ correspond to no short convolutions and long convolutions, respectively.

above. We define a *short convolution* of finite filter size k with diagonal filter matrices $\mathcal{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_k) \subseteq \mathbb{R}^{D\times D}$ as an operator $\mathcal{C}_{\text{short}}$ which, applied to an input $\mathbf{X} \in \mathbb{R}^{T\times D}$, computes

$$\mathcal{C}_{\text{short}}(\mathbf{X}, \mathcal{W}) = \sum_{j=0}^{k-1} \mathbf{S}^j \mathbf{X} \mathbf{W}_j, \qquad \mathbf{S} \in \mathbb{R}^{T\times T} \quad \text{with} \quad \mathbf{S}_{ij} = \begin{cases} 1, & i = j+1, \\ 0, & \text{otherwise.} \end{cases}$$

Each output therefore depends on the current token and its $k-1$ predecessors, if existent. Recent work identifies short convolutions as an effective mechanism for local token mixing in attention-based models. (Li et al., 2025) show that convolution-augmented attention improves associative recall and length generalization by injecting local positional structure. (Allen-Zhu, 2025) similarly finds that lightweight causal convolutions improve reasoning in Transformers and can partially substitute for positional encodings.

Long convolutions convolve each token with all previous ones (we informally write $k = \infty$). We denote a long convolution operator as $\mathcal{C}_{\text{long}}$. It geometrically decays an input sequence $\mathbf{X} \in \mathbb{R}^{T\times D}$ using a diagonal matrix $\mathbf{G} \in \mathbb{R}^{D\times D}$ as diagonal gain matrices $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{R}^{D\times D}$, $\mathcal{C}_{\text{long}}(\mathbf{X}, \mathbf{G}, \mathbf{B}_0, \mathbf{B}_1) = \mathbf{X}\mathbf{B}_0 + \sum_{j=1}^{T-1} \mathbf{S}^j \mathbf{X} \mathbf{G}^j \mathbf{B}_1$ In practice, parameter-efficient long convolutions are often defined through an exponential moving average (Gu et al., 2022).

RoPE. Rotary position embeddings (RoPE) introduce positional information to the attention mechanism by pre-processing queries and keys using position-dependent rotations, making their dot products explicitly dependent on relative token positions. For even D, define the frequencies $\omega_r = \beta^{-2r/D}$ for $r = 0, \ldots, D/2-1$, where $\beta > 1$ is fixed. Pairing adjacent coordinates, define the fixed block-diagonal rotation matrix

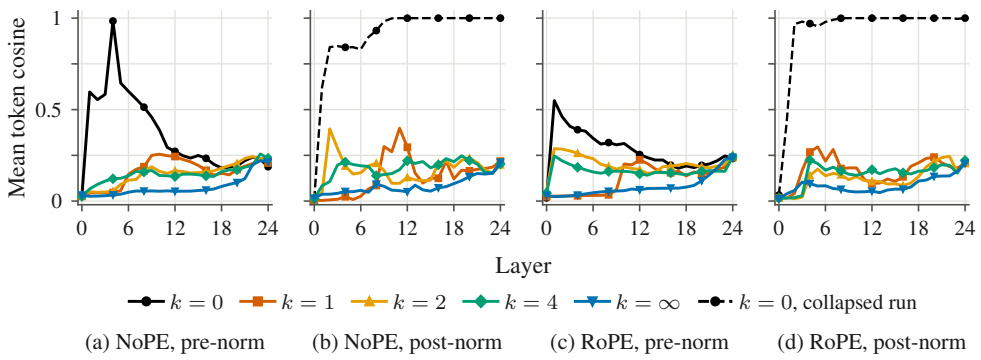
$$\mathbf{R} = \operatorname{diag}\left(\operatorname{Rot}(\omega_0), \ldots, \operatorname{Rot}(\omega_{D/2-1})\right), \qquad \operatorname{Rot}(\omega) = \begin{pmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{pmatrix}.$$

At position t, RoPE applies $\mathbf{R}^t$ to both queries and keys, rotating each adjacent coordinate pair by the angle $t\omega_r$: $(\mathbf{Q}_n^\ell)_t \mapsto (\mathbf{Q}_n^\ell)_t \mathbf{R}^{t^\top}$, $(\mathbf{K}_n^\ell)_t \mapsto (\mathbf{K}_n^\ell)_t \mathbf{R}^{t^\top}$. Since $\mathbf{R}$ is orthogonal, $(\mathbf{R}^t)^\top \mathbf{R}^s = \mathbf{R}^{s-t}$, and the resulting attention-score matrix transforms as $(\mathbf{Q}_n^\ell)_\tau (\mathbf{K}_n^{\ell^\top})_t \mapsto (\mathbf{Q}_n^\ell)_\tau \mathbf{R}^{t-\tau} (\mathbf{K}_n^{\ell^\top})_t$. When query and key convolutions are enabled, these rotations are applied after convolution. Architectures that do not use RoPE are denoted by *NoPE*. NoPE can be recovered by setting $\mathbf{R}_t = \mathbf{I}$. Fig. 7 summarizes normalization placement, the choice of short, long, or no convolutions in attention and SwiGLU, and the choice between RoPE and NoPE.

3 Convolutions and Optimization

Convolutions are known to provably aid length generalization (Li et al., 2025) and to unlock expressivity[1] in shallow models (Okpekpe & Orvieto, 2025). In this section, we show that, surprisingly,

[1]For instance, a one-layer Transformer block can perfectly solve multi-query associative recall (Arora et al., 2024b) only if convolutions are present (Okpekpe & Orvieto, 2025).



(a) NoPE, pre-norm (b) NoPE, post-norm (c) RoPE, pre-norm (d) RoPE, post-norm

Figure 2: **Rank collapse.** Mean pairwise cosine similarity between the token representations of a layer, averaged over the held-out validation sequences, for convolution kernel sizes $k \in \{0, 1, 2, 4, \infty\}$ at the best learning rate of each configuration. Layer 0 is the embedding. A value of 1 means every token maps to the same direction. Post-norm models without convolutions never train; the dashed curve is one of the collapsed runs.

convolutions can also provide drastic training benefits. For this purpose, we investigate the robustness of model optimization when it comes to the choices in learning rate, normalization type (pre-norm and post-norm), and positional encoding (NoPE or RoPE). Starting with a gain (convolution with $k = 1$) to isolate the role of channel-wise scaling, we then increase the kernel size to also introduce the impact of token shifting. We further analyze the impact of convolutions on representational collapse and optimizer updates during training.

3.1 Trainability of post-norm Transformers

Learning rate robustness. Post-norm Transformers can be difficult to train at large depth (Wang et al., 2024; Noci et al., 2022). To investigate the impact of convolutions on trainability, we first consider training post-norm Transformers with various learning rates and two choices of positional encoding: NoPE and RoPE. The results are shown in Fig. 1(b, d).

Residual scaling. We observe that with a gain ($k = 1$ convolution), post-norm already becomes trainable, while RoPE substantially improves perplexity compared to NoPE in this setting. We hypothesize that this improvement arises because the positional embeddings learned by NoPE are more vulnerable to rank-collapse (Kazemnejad et al., 2023). Furthermore, an increase in kernel size ($k = 2, 4, \infty$) generally improves model performance, although successful training is already possible with channel-wise scaling alone.

Representational diversity. In Fig. 2(b, d), we provide the average pairwise cosine similarity between token representations from the residual stream $\boldsymbol{X}^{\ell}$ as a rank-collapse metric (Noci et al., 2022). With rank-collapse, we expect the average cosine similarity to approach 1, while a smaller value generally indicates a greater representational diversity. We observe that convolutions, regardless of their kernel size, improve the diversity of representation in a post-norm Transformer, while the model without any convolutions ($k = 0$) exhibits rank-collapse. Moreover, with NoPE, larger kernel sizes seem to reduce fluctuation across layers in the rank-collapse metric, while the highest representational diversity is observed in the long convolution model. RoPE similarly reduces fluctuations across layers, which we again attribute to the positional encoding learned by NoPE to be more vulnerable to rank-collapse.

Canon layers. We also investigate placing convolutions before and after the projections, as proposed in canon layers (Allen-Zhu, 2025). The results are shown in Fig. 8(Appendix A). We observe that adding more convolutions improves the stability of the model, flattening the learning rate curve, although performance drops when the two pre-projection convolutions are introduced. We attribute the improvement in stability to further down-scaling of the residual branch, which can improve representational diversity. However, further down-scaling the residual branch may also diminish its contribution to the residual stream, potentially explaining the lower performance within a fixed training budget.

3.2 Learning rate robustness in pre-norm Transformers

Learning rate robustness. In Fig. 1(a, c) we provide validation perplexity across learning rates for pre-norm Transformers, with both NoPE and RoPE. We observe a similar trend in optimization robustness, with the learning rate curve significantly flattened at the presence of convolutions. Similarly to Section 3.1, we observe an improvement in average pair-wise cosine similarity when introducing convolutions to the architecture, as shown in Fig. 2(a, c). Interestingly, we observe some degradation in the representational diversity of the early layers in the pre-norm model with NoPE compared to the RoPE model.

Representational diversity. In order to investigate the impact of the observed degradation in representational diversity of the NoPE model with $k = 0$ in Fig. 2(a), we plot the gradient norm of the model w.r.t. the query and key parameters for each layer for 3 checkpoints on the optimization trajectory in Fig. 9. As rank-collapse introduces gradient vanishing to the query and the key weights, we expect a drop in the norm of the gradients w.r.t. those parameters in the early layers. This trend is visible in Fig. 2(a), providing further evidence for rank-collapse in NoPE. In Fig. 3(b, c, d), we observe the magnitude of the Adam update size per-layer, normalized by the magnitude of the weights. We can see that the Adam update size becomes very uneven during training for the models with $k > 1$, with the early layers suffering the most from the unevenness. We therefore conclude a relationship between NoPE and a little bit rank-collapse in the pre-norm model. And we also note that since RoPE alleviates this problem, then maybe positional encoding alleviates rank-collapse to some extent.

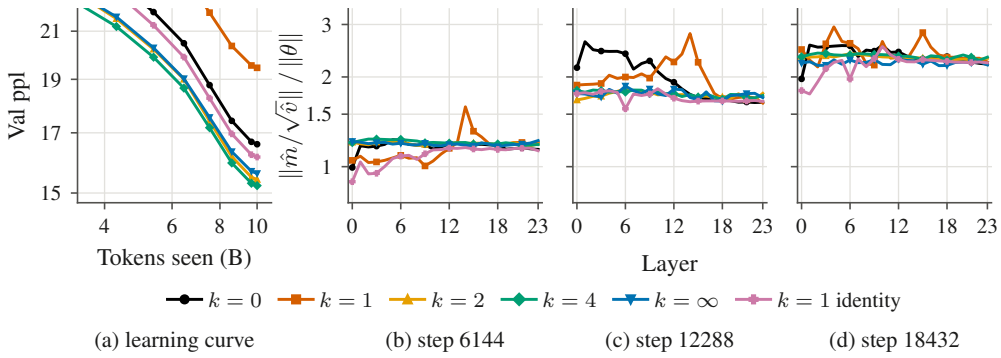
To investigate the implications of reduced representational diversity on optimization in the NoPE model with $k = 0$ shown in Fig. 2(a), we examine the query and key gradient norms in each layer at three training checkpoints (Fig. 9, Appendix A). Motivated by the connection between rank-collapse and vanishing query and key gradients at initialization (Noci et al., 2022), we test whether the early layers with reduced representational diversity also exhibit suppressed gradients during training. We observe lower query and key gradient norms in these layers, consistent with this mechanism.

We further examine the relative Adam update magnitudes across layers, normalized by the corresponding parameter norms (Fig. 3(b, c, d)). The updates are particularly uneven in these early layers, suggesting an association between reduced representational diversity and uneven optimization dynamics. Although gradient suppression alone does not explain the variation in Adam updates, these observations motivate the hypothesis that partial rank-collapse contributes to the learning rate sensitivity of the NoPE model.RoPE partially alleviates the degradation in representational diversity, suggesting that the way positional information is represented may contribute to these optimization difficulties. We hypothesize that NoPE's reliance on learned absolute positional information makes it more susceptible to this degradation. However, the comparison with RoPE alone does not isolate the role of absolute position.

Initialization. The initialization of the convolution layers also impacts model performance. Initializing the filters to the identity eliminates the gap between the $k = 1$ and $k = 0$ models in Fig. 1(a, c), while further improving the performance of the $k = 4$ model relative to the $k = 0$ baseline Fig. 10 (Appendix A). This suggests that the performance gap observed between the models with $k > 0$ and the $k = 0$ model depends on initialization. However, identity initialization leads to divergence in our post-norm experiments, highlighting the need to choose the convolution initialization jointly with the normalization method.

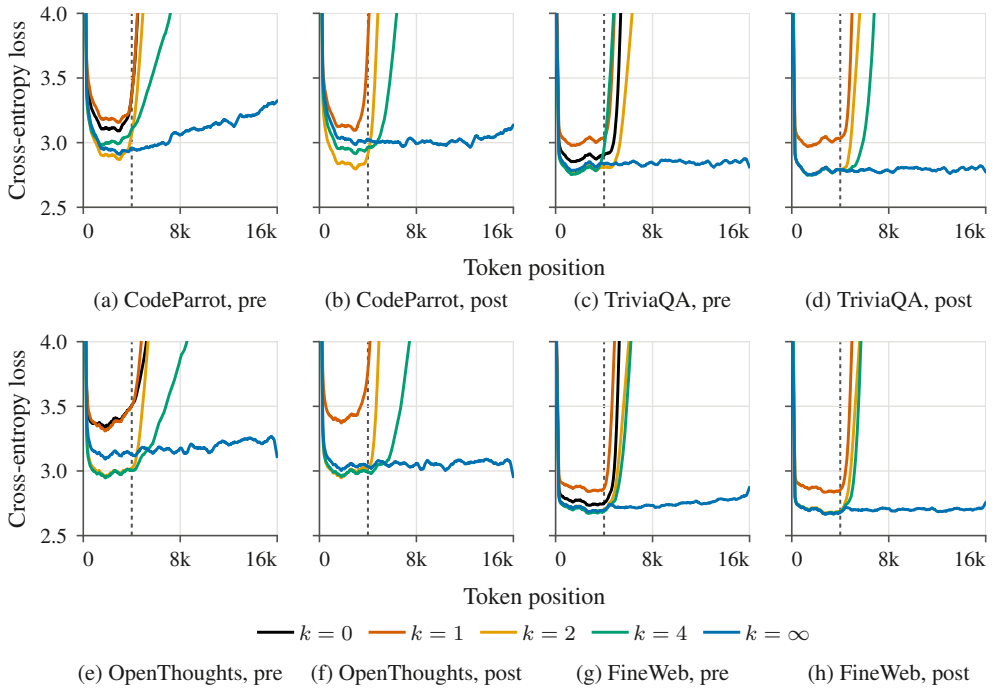
4 Convolutions and Positional Encoding

In the previous section, we observed that RoPE flattens the learning rate curves of the $k = 0$ and $k = 1$ models, making them more similar to those of models with $k > 1$ (Fig. 1(a, c)). Together with the improvements in representational diversity observed with RoPE, this motivates us to ask whether token shifting in convolutions also provides positional information. We investigate this possibility by first examining length generalization, where convolutions improve performance beyond the training context length. We then study how introducing RoPE changes the model's reliance on token shifting within the convolution filters. Finally, we identify components in the residual stream that vary primarily with position, and examine how the model's reliance on these components changes when convolutions or RoPE are introduced. Together, these analyses explore whether convolutions provide an alternative means of representing positional information.



(a) learning curve (b) step 6144 (c) step 12288 (d) step 18432

Figure 3: **Size of the optimizer update through training.** Norm of the AdamW update applied to each block, divided by the norm of that block's parameters, at three equidistant points of the 19075-step run, for pre-norm NoPE Transformers at a common learning rate of 8×10^{-3}. The update direction is $\hat{m}/(\sqrt{\hat{v}} + \epsilon)$ with bias correction, summed in quadrature over the block's tensors and excluding weight decay. The scheduled learning rate is divided out, so the quantity plotted depends only on the ratio of Adam's first and second moments relative to the size of the block's parameters, and is comparable across checkpoints on a decaying schedule. The vertical axis is logarithmic and shared.



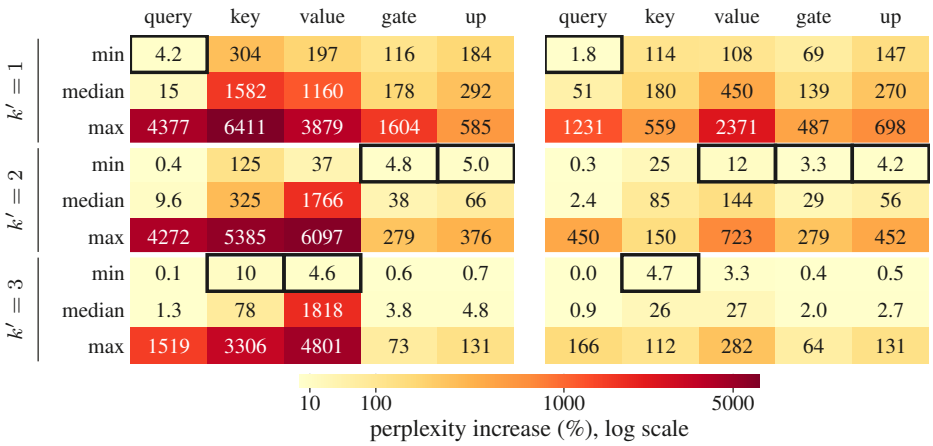
(a) CodeParrot, pre (b) CodeParrot, post (c) TriviaQA, pre (d) TriviaQA, post

(e) OpenThoughts, pre (f) OpenThoughts, post (g) FineWeb, pre (h) FineWeb, post

Figure 4: **Length extrapolation.** Per-token cross-entropy loss against position for NoPE Transformers trained at 4096 tokens (dashed line) and evaluated at 16384, for convolution kernel sizes $k \in \{0, 1, 2, 4, \infty\}$. Each adjacent pair of panels is one evaluation set, with the *pre*-norm model on the left and the *post*-norm model on the right; the sets are CodeParrot, TriviaQA, OpenThoughts-114k-Math and held-out FineWeb. Losses are a 501-token moving average.

4.1 Length Generalization

Length generalization is a well-known test-bed for positional encoding (Lin et al., 2025; Kazemnejad et al., 2023; Press et al., 2022), with short convolutions shown to improve length generalization (Li et al., 2025). This motivates using length generalization to investigate the role of convolutions in representing positional information in sequence models. In Fig. 4, we evaluate length generalization across convolution kernel sizes.



(a) NoPE (b) RoPE

Figure 5: **Compressing the convolution filter.** The changes in validation perplexity when compressing a convolution of size $k = 4$ into a convolution of size $k' \in \{1, 2, 3\}$ at inference time, for each component (query, key, value, and the up-projection and gate of the SwiGLU layer), in pre-norm NoPE (left) and RoPE (right) Transformers. Each of the four coefficients of the trained model's filter is either assigned to a position of the reduced filter or dropped, with coefficients sharing a position summed together; the minimum, median, and maximum changes in perplexity (in %) are reported. For each component, the outlined cell marks the smallest k' whose best pattern increases perplexity by at most 15%. For both architectures, the unmodified perplexity is 15.23.

We observe that larger convolution kernels generally perform better beyond their training sequence length. The long convolution models perform exceptionally well on this problem, with the loss curve remaining almost flat for up to $\times 4$ training sequence length. These observations can be interpreted as convolutions playing a role in positional encoding, thus motivating further investigation into whether convolutions provide positional information, which we pursue in the remainder of this section.

4.2 The effect of RoPE on convolutional filters

To investigate the impact of explicit initialization on the behavior of convolutions, we compare the effect of compressing a pre-trained convolution of size $k = 4$ to $k' \in \{1, 2, 3\}$ for NoPE and RoPE positional encodings in a pre-norm Transformer. The results are shown in Fig. 5. We construct shorter filters at inference time by reassigning or removing filter coefficients, summing coefficients assigned to the same position. We apply this intervention separately to the query, key, value, and SwiGLU convolutions and measure the resulting change in validation perplexity. We report the lowest, the highest, and the median amount of change in perplexity. This allows us to assess how much each component relies on token shifting and the impact of RoPE on it, as opposed to channel-wise scaling, which will be preserved by the way we merge the filters.

As we observe, the query does not utilize the token shifting capabilities of the convolution, with even a gain ($k' = 1$) being enough to keep the changes in perplexity $< 15\%$. This behavior does not change with RoPE. On the other hand, the key and the value are both very sensitive to kernel size, with both of them needing at least $k' = 3$ with NoPE to keep the same level of performance. However, we observe that RoPE changes this behavior, with the value convolutions becoming slightly more short-ranged ($k' = 2$), and the maximum and the median drop in performance significantly lowered in both for $k' \leq 2$. This suggests a change in the way the model utilizes these convolutions. Theorem 1 provides some theoretical insight into why this is the case.

Proposition 1. *Fix $L \geq 2$ linearly independent embeddings $x_1, \ldots, x_L \in \mathbb{R}^m$ and a nonzero terminal embedding $x_\star \in \mathbb{R}^m$ attending only to these L inputs. A single softmax attention head without biases or positional encodings can distinguish all permutations at its terminal output using suitable linear projections and a zero-padded causal filter of size 2 on keys alone or, separately,*

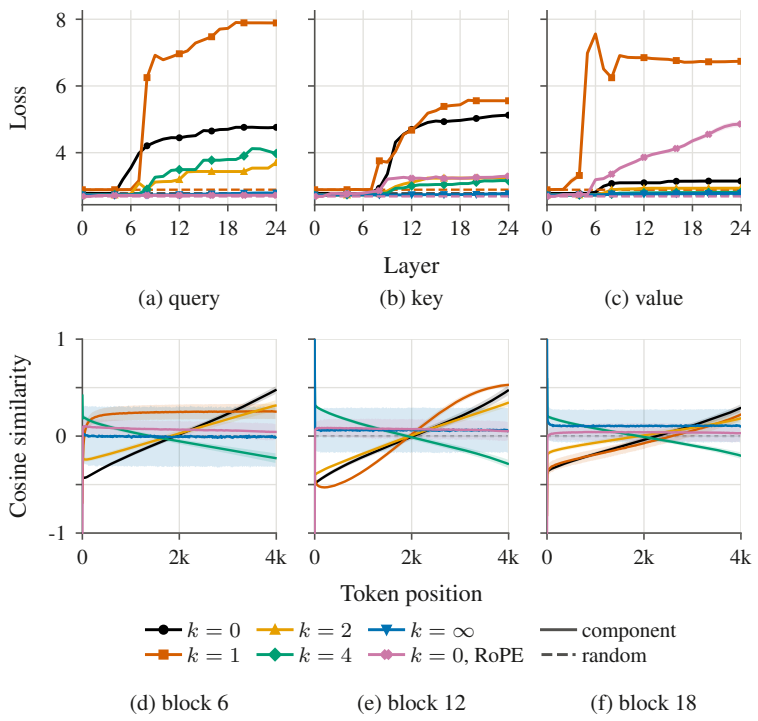


Figure 6: **The first position component.** *Top* (a–c): next-token loss when the first PCA component is removed from the input of query/key/value projections in every layer up to the depth on the x axis, for pre-norm NoPE Transformers. At depth 0 each curve is its unmodified baseline; shading is the paired 95% sequence-bootstrap interval, and dashed curves remove a random direction instead. *Bottom* (d–f): cosine similarity between each token representation and the first PCA direction against token position, at three block outputs, shaded ± 1 sample standard deviation across the 4096 evaluation sequences.

values alone, with all parameters fixed across permutations. For query-only filtering, the minimum required size is instead L. A proof is given in Appendix B

The proposition shows that only the key and the value can encode position with a finite filter size. Together with the result proposed by Li et al. (2025), where they show only the key can use convolutions to solve associative recall in a single layer, we conclude that at least when it comes to positional encoding and expressivity, queries do not benefit from short convolutions.

Finally, we point out that the MLP components seem to require $k' = 2$ to retain the same level of performance, with RoPE having very little impact on their behavior. This observation hints at the MLP components mostly using convolutions as either channel-wise scalers and local sequence mixers, with their effect being mostly local and limited to the first 2 filters.

4.3 Content-independent directions

We next try to identify directions in the residual stream whose activation primarily vary with time t and not the content. In other words, we define the output of the model at layer ℓ for sequence n at time t as $\boldsymbol{x}^\ell_{t,n} = \boldsymbol{c}^\ell_{t,n} + \boldsymbol{p}^\ell_t + \hat{\boldsymbol{x}}^\ell$, where $\boldsymbol{c}^\ell_{t,n}$ is the content-dependent component of the representation, $\boldsymbol{p}^\ell_t$ is the position dependent component of the representation, and $\hat{\boldsymbol{x}}^\ell$ the mean component of the representation. This view of the residual stream, as introduced by (Song & Zhong, 2023), allows us to detect these directions via principle component analysis (PCA). Specifically, for each layer ℓ we define:

$$\hat{\boldsymbol{x}}^\ell = \frac{1}{TN} \sum_{t,n=1}^{T,N} \boldsymbol{x}^\ell_{t,n}, \quad \bar{\boldsymbol{x}}^\ell_t = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}^\ell_{t,n}, \quad \boldsymbol{P}^\ell = \begin{bmatrix} \bar{\boldsymbol{x}}^\ell_1 - \hat{\boldsymbol{x}}^\ell & \bar{\boldsymbol{x}}^\ell_2 - \hat{\boldsymbol{x}}^\ell & \dots & \bar{\boldsymbol{x}}^\ell_T - \hat{\boldsymbol{x}}^\ell \end{bmatrix}^\top.$$

We perform PCA on $\boldsymbol{P}^{\ell}$, deriving a set of directions $\mathcal{V} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_T\}$. We then remove the contribution along a selected direction from $\mathcal{V}$ from either the query, the key, or the value up to a certain layer, and measure the changes in loss. Comparing these interventions across layers allows us to investigate the interaction between the model and a position dependent component (Song & Zhong, 2023). We provide the results for this experiment for the first PCA component in Fig. 6(a, b, c), and the results for the top three PCA components in Fig. 12. In Fig. 11, we provide the results for interventions in one layer. For all models except long convolutions, the cosine similarity between token representations and the top directions in $\mathcal{V}$ remain largely a function of t, as is observed for the first PCA component in Fig. 6(d, e, f) and the top three PCA components in Fig. 13.

The results suggest that the $k = 0$ and $k = 1$ models are mostly impacted by these directions, with the impact getting less significant for convolutions with larger kernels. Furthermore, we observe that RoPE introduces a similar effect, although the top PCA component seems to still be slightly influential on the value vector. The comparison with $k = 1$ helps distinguish this effect from the optimization benefits of channel-wise scaling. Although a $k = 1$ gain convolution (gain) improves trainability, the resulting model continues to rely on these directions. The lower sensitivity observed with $k > 1$ therefore suggests an additional role for token shifting. This interpretation complements the filter-compression results in Fig. 5. Introducing RoPE makes the key and value convolutions more tolerant to compression (Fig. 5), while introducing convolutions with larger kernels reduces reliance on position-dependent directions in the residual stream (Fig. 6(a, b, c)). Together, these observations support the hypothesis that *token shifts performed by convolutions can introduce positional encoding to the model.*

5 Convolutions and Massive Activations

Massive activations are extreme outliers concentrated in a few hidden channels and token positions, which can act as implicit biases and shape attention allocation (Sun et al., 2024). Such outliers complicate low-precision quantization, making their architectural origins practically important (Dettmers et al., 2022). Prior work identifies feed-forward amplification, residual accumulation, normalization placement, and attention-output gating as factors governing their emergence and magnitude (Sun et al., 2026; Qiu et al., 2025). Here, we investigate how convolutions affect massive activations in the residual stream. In Fig. 14(Appendix A), we compare the maximum-to-median activation ratio across convolution kernel sizes for NoPE and RoPE Transformers. We observe lower ratios for all convolution variants than for their corresponding baselines without convolutions, with larger reductions under RoPE. A reduction is already present at $k = 1$, which is effectively only channel-wise scaling. These reductions are also positively correlated with the optimization stability observed in Fig. 1.

6 Discussion and Conclusion

Our paper support the following broader claims. **Convolutions and optimization.** With proper initialization, convolutions improve optimization through channel-wise scaling. This improvement can be observed in the form of flat learning rate curves and stable post-norm Transformers.

Convolutions alter positional encoding. Larger kernel sizes improve length generalization in convolution augmented Transformers. Further analysis shows that convolutions modify how NoPE Transformers encode positional information, which can directly be attributed to the token shifting functionality provided by convolutions.

Convolutions improve recall and downstream performance. Downstream performance favors short convolutions, while recall tasks heavily benefit from the presence of long convolutions Table 1.

Future work. For future work, we aim to consider dynamic convolutions in our analysis (Sieberling et al., 2026; Lahoti et al., 2026). Furthermore, an important direction to explore is the role of scale on the benefits of convolutions. Convolutions also exhibit lower maximum-to-median activation ratios, although whether these reductions contribute to training stability remains unresolved. Given this observation, we also aim to explore better quantization as another possible benefit of convolutions.

REFERENCES

Zeyuan Allen-Zhu. Physics of language models: Part 4.1, architecture design and the magic of canon layers. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/3c4141c12660ad3625eb4ae845e0a6f9-Abstract-Conference.html.

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. Mechanistic evaluation of transformers and state space models. *CoRR*, abs/2505.15105, 2025. doi: 10.48550/ARXIV.2505.15105. URL https://doi.org/10.48550/arXiv.2505.15105.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=LY3ukUANko.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International conference on learning representations*, volume 2024, pp. 15664–15730, 2024b.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1763–1840. PMLR / OpenReview.net, 2024c. URL https://proceedings.mlr.press/v235/arora24a.html.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. URL http://arxiv.org/abs/1803.01271.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 2017. URL http://proceedings.mlr.press/v70/dauphin17a.html.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/c3ba4962c05c49636d4c6206a97e9c8a-Abstract-Conference.html.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023. doi: 10.48550/ARXIV.2312.00752. URL https://doi.org/10.48550/arXiv.2312.00752.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in neural information processing systems*, 35:35971–35983, 2022.

Md. Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJeB36NKvB.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/4e85362c02172c0c6567ce593122d31c-Abstract-Conference.html.

Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. *CoRR*, abs/2603.15569, 2026. doi: 10.48550/ARXIV.2603.15569. URL https://doi.org/10.48550/arXiv.2603.15569.

Mingchen Li, Xuechen Zhang, Yixiao Huang, and Samet Oymak. On the power of convolution-augmented transformer. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 18393–18402. AAAI Press, 2025. doi: 10.1609/AAAI.V39I17.34024. URL https://doi.org/10.1609/aaai.v39i17.34024.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron C. Courville. Forgetting transformer: Softmax attention with a forget gate. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=q2Lnyegkr8.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurélien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ae0cba715b60c4052359b3d52a2cff7f-Abstract-Conference.html.

Destiny Okpekpe and Antonio Orvieto. Revisiting associative recall in modern recurrent models. *arXiv preprint arXiv:2508.19029*, 2025.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/904e89bb4e632e75fb47f093b620b257-Abstract-Conference.html.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 901, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/ed265bc903a5a097f61d3ec064d96d2e-Abstract.html.

Oliver Sieberling, Bharat Runwal, Rameswar Panda, and Yoon Kim. Dynamic short convolutions improve transformers. *CoRR*, abs/2606.03825, 2026. doi: 10.48550/ARXIV.2606.03825. URL https://doi.org/10.48550/arXiv.2606.03825.

Jiajun Song and Yiqiao Zhong. Uncovering hidden geometry in transformers via disentangling position and context. *CoRR*, abs/2310.04861, 2023. doi: 10.48550/ARXIV.2310.04861. URL https://doi.org/10.48550/arXiv.2310.04861.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *CoRR*, abs/2402.17762, 2024. doi: 10.48550/ARXIV.2402.17762. URL https://doi.org/10.48550/arXiv.2402.17762.

Shangwen Sun, Alfredo Canziani, Yann LeCun, and Jiachen Zhu. The spike, the sparse and the sink: Anatomy of massive activations and attention sinks. *CoRR*, abs/2603.05498, 2026. doi: 10.48550/ARXIV.2603.05498. URL https://doi.org/10.48550/arXiv.2603.05498.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *CoRR*, abs/1706.05350, 2017. URL http://arxiv.org/abs/1706.05350.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(10):6761–6774, 2024. doi: 10.1109/TPAMI.2024.3386927. URL https://doi.org/10.1109/TPAMI.2024.3386927.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d13a3eae72366e61dfdc7eea82eeb685-Abstract-Conference.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

A Appendix

B Proof of Proposition 1

Notation. Let $L \geq 2$ be an integer. Fix linearly independent vectors $x_1, \ldots, x_L \in \mathbb{R}^m$ and a nonzero vector $x_\star \in \mathbb{R}^m$. We study softmax attention's awareness of positional information by considering sequences of length $L+1$ ending in $x_\star$, constructed by permuting the x_is. A permutation $\pi = (\pi_1, \ldots, \pi_L)$ is a vector of integers containing each integer in $\{1, \ldots, L\}$ exactly once. A permutation π defines the sequence $x_j^\pi = x_{\pi_j}$ for $1 \leq j \leq L$ and $x_{L+1}^\pi = x_\star$.

Choose linear projections $W_Q, W_K \in \mathbb{R}^{d \times m}$ and $W_V \in \mathbb{R}^{d_v \times m}$, and filter exactly one of the three projected sequences. For example, for key filtering, define

$$q_j^\pi = W_Q x_j^\pi, \qquad k_j^\pi = \sum_{a=0}^{h-1} c_a \odot W_K x_{j-a}^\pi, \qquad v_j^\pi = W_V x_j^\pi. \tag{1}$$

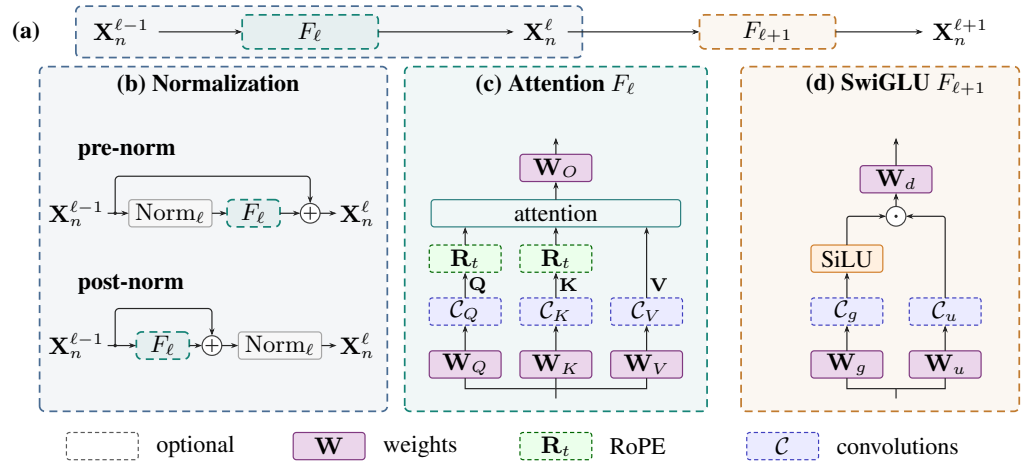


Figure 7: **Modular Transformer block.** (a) Two residual layers with (b) pre- or post-norm, comprising (c) attention and (d) SwiGLU. Matching colors identify expanded components. Optional convolutions are short or long.

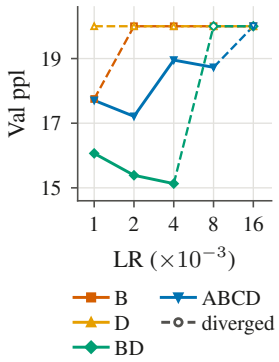


Figure 8: **Canon positions and post-norm.** Validation perplexity against learning rate ($\times 10^{-3}$) for post-norm NoPE Transformers with the long convolution at different Canon positions: A and B before and after q, k, v projection, C and D before and after SwiGLU projection, respectively.

For query or value filtering, we apply the sum to that projection instead, leaving the other two unfiltered. Here $x_j^\pi = 0$ for $j \le 0$, $\odot$ denotes coordinatewise multiplication, and the coefficients c_a are vectors of the filtered projection's dimension, shared across positions. We require $h \ge 1$ and $c_{h-1} \neq 0$.

With one attention head, no biases and no positional embeddings, let only the first L positions contribute to the terminal output $z_\star(\pi) \in \mathbb{R}^{d_v}$:

$$z_\star(\pi) = \sum_{j=1}^{L} \frac{\exp((q_{L+1}^\pi)^\top k_j^\pi / \sqrt{d})}{\sum_{\ell=1}^{L} \exp((q_{L+1}^\pi)^\top k_\ell^\pi / \sqrt{d})} v_j^\pi. \tag{2}$$

Precise Statement of the Result. For every $L \ge 2$, every fixed collection of linearly independent vectors $x_1, \ldots, x_L \in \mathbb{R}^m$ and every fixed nonzero $x_\star \in \mathbb{R}^m$, the following holds.

1. For key-only filtering and, separately, value-only filtering, a filter of size $h = 2$ suffices to distinguish all permutations. Specifically, there exist finite dimensions d, d_v, matrices $W_Q, W_K \in \mathbb{R}^{d \times m}$ and $W_V \in \mathbb{R}^{d_v \times m}$, and filter coefficients c_0, c_1 with $c_1 \neq 0$ such that, for every pair of permutations π and σ,

$$\pi \neq \sigma \Longrightarrow z_\star(\pi) \neq z_\star(\sigma). \tag{3}$$

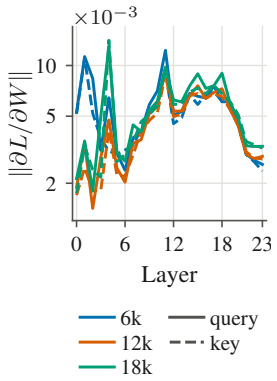


Figure 9: **Query and key gradients by depth.** Norm of the loss gradient with respect to the query (solid) and key (dashed) projection weights of each block, for the pre-norm NoPE Transformer without convolutions, at three equidistant checkpoints (steps in thousands).

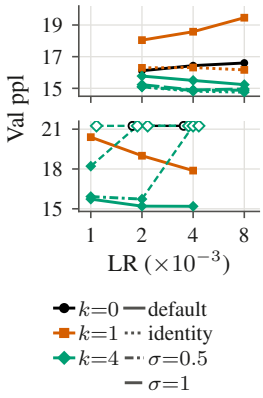


Figure 10: **Convolution inilization.** Validation perplexity of NoPE Transformers against learning rate. Colour and marker are the kernel size and the line style the filter init $\mathcal{N}(0, \sigma^2)$: the default $\sigma = 0.02$, identity, and $\sigma = 0.5$ and 1. *Top*, pre-norm; *Bottom*, post-norm. Hollow markers at the top of a panel are runs that collapsed.

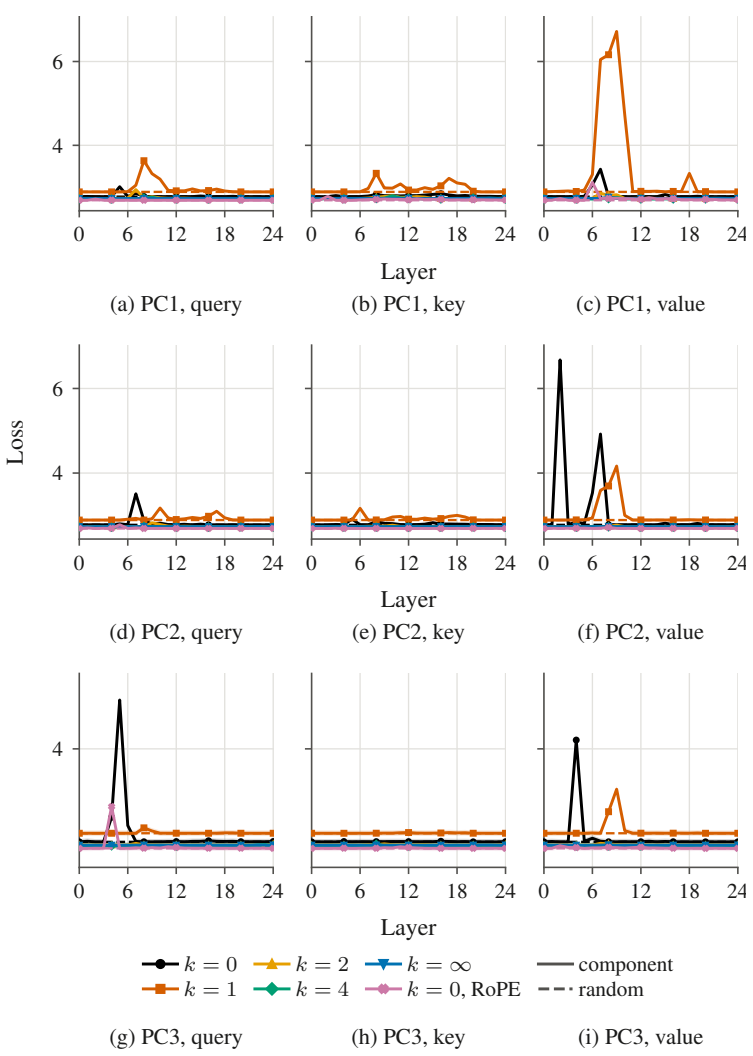


Figure 11: **Cost of deleting the position components from a single layer.** As in the cumulative figure, but the direction is removed from one layer only, the one on the x axis, with every other layer left untouched, for pre-norm NoPE Transformers at each model's best learning rate. Point 0 is the unmodified baseline. Rows are the first three principal components, columns the projection the direction is removed from. Shading is the paired 95% sequence-bootstrap interval and dashed curves are the matching random-direction controls, which are drawn afresh for this experiment. The additional curve is the no-convolution model trained with rotary embeddings, on the same axes, as a reference for how much of this cost an explicit positional encoding removes. Its rotary embedding stays active throughout, so it too shows reliance on position information carried by the residual stream, not on the rotary encoding itself.

2. For query-only filtering, the minimum size is instead $h = L$. There exist matrices of the same dimensions and filter coefficients $c_0, \ldots, c_{L-1}$ with $c_{L-1} \neq 0$ satisfying the same implication for every pair of permutations, but no filter of size $h < L$ can do so.

In each case, the embeddings stay fixed. The matrices and filter coefficients may depend on these embeddings and on which projection is filtered, but not on the permutation.

Lower bounds. These bounds hold for every choice of projections and filter coefficients in arbitrary finite dimensions; they do not require linear independence. To start, recall that for $h = 1$ $z(\pi)$

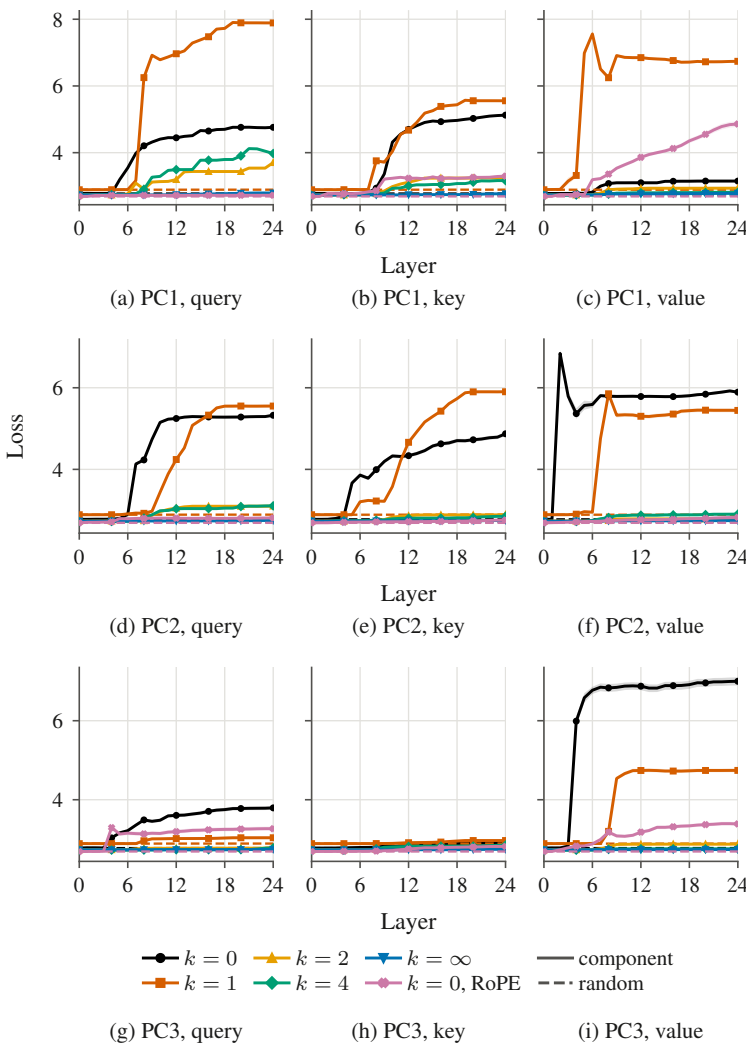


Figure 12: **Cost of deleting the position components from the attention inputs.** Next-token loss when a position-PCA direction is removed from the input of one projection in every layer up to the depth on the x axis, for pre-norm NoPE Transformers at each model's best learning rate. At depth 0 each curve is its unmodified baseline. Rows are the first three principal components, columns the projection the direction is removed from. Shading is the paired 95% sequence-bootstrap interval. Dashed curves remove a random direction instead, as a control; the source reuses one control per model across the three components. Removal acts on a private copy of the residual, leaving the residual stream and the other projections untouched. The additional curve is the no-convolution model trained with rotary embeddings, on the same axes, as a reference for how much of this cost an explicit positional encoding removes. Its rotary embedding stays active throughout, so it too shows reliance on position information carried by the residual stream, not on the rotary encoding itself.

is a set operation; thus, size one cannot distinguish any two permutations, so key and value filtering requires $h \geq 2$.

For query filtering, fix arbitrary projection dimensions, matrices and filter coefficients, and suppose $1 \leq h < L$. Given any permutation π, define σ by $\sigma_1 = \pi_2$, $\sigma_2 = \pi_1$ and $\sigma_j = \pi_j$ for $3 \leq j \leq L$. Then $\sigma \neq \pi$, but $x_j^\sigma = x_j^\pi$ for every $3 \leq j \leq L + 1$.

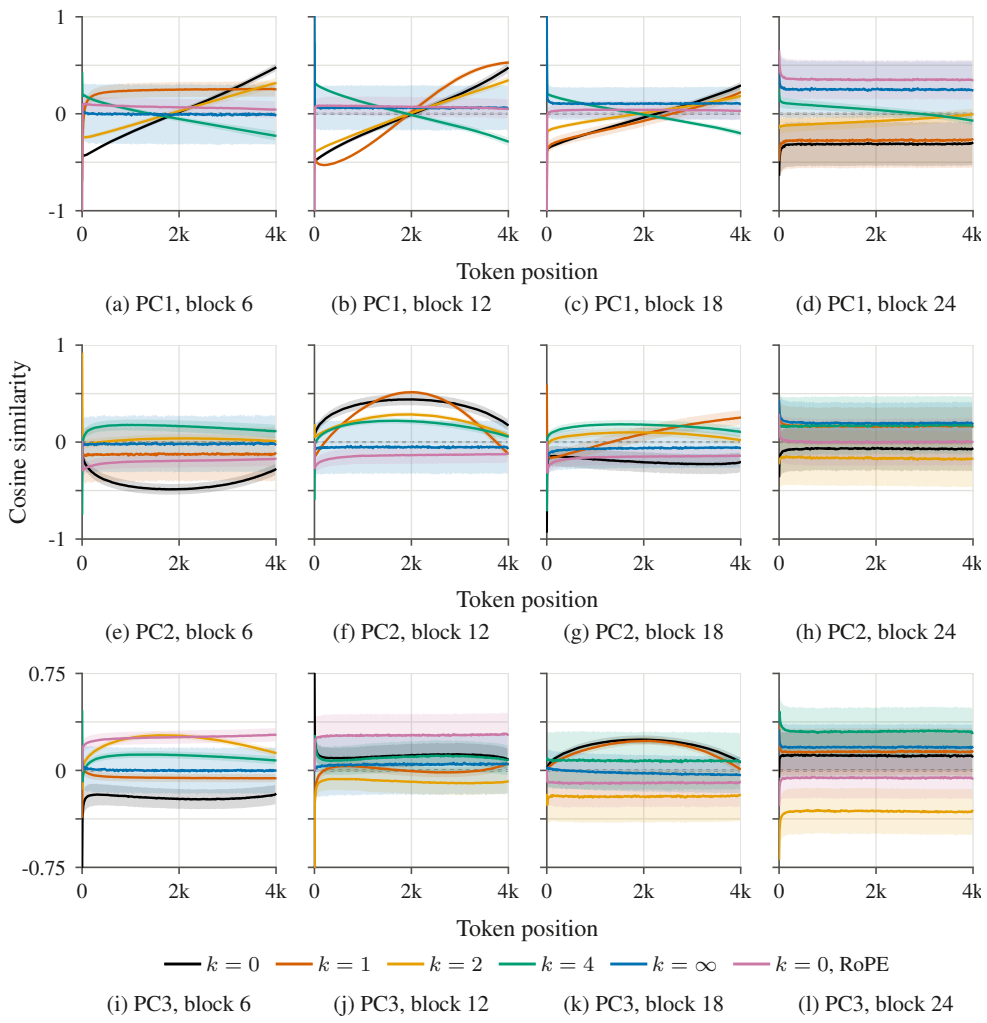


Figure 13: **Position information in the principal components.** Cosine similarity between each token representation and the position-PCA directions fitted on held-out calibration sequences, against token position, for pre-norm NoPE Transformers at each model's best learning rate. Rows are the first three principal components, columns the block output at depth $6, 12, 18, 24$. Shading is ± 1 sample standard deviation across the 4096 evaluation sequences. Cosines are computed per token before averaging, with no mean subtracted.

For every $0 \le a \le h-1$, we have $L+1-a \ge L-h+2 \ge 3$. Consequently,

$$q^{\sigma}_{L+1} = \sum_{a=0}^{h-1} c_a \odot W_Q x^{\sigma}_{L+1-a} = \sum_{a=0}^{h-1} c_a \odot W_Q x^{\pi}_{L+1-a} = q^{\pi}_{L+1}. \tag{4}$$

The keys and values are unfiltered. Reindexing the attention sum by $b = \pi_j$ we get

$$z_\star(\pi) = \sum_{b=1}^{L} \frac{\exp((q^{\pi}_{L+1})^\top W_K x_b/\sqrt{d})}{\sum_{\ell=1}^{L} \exp((q^{\pi}_{L+1})^\top W_K x_\ell/\sqrt{d})} W_V x_b = z_\star(\sigma). \tag{5}$$

Thus two distinct permutations have the same output for every parameter choice when $h < L$.

Upper bound: key-only filtering. Take $d = 1$, $d_v = m$, and let $X = (x_1, \dots, x_L) \in \mathbb{R}^{m \times L}$ be the matrix whose columns are the fixed embeddings. Since these columns are linearly independent,

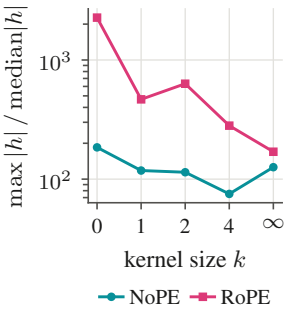


Figure 14: **Convolutions shrink massive activations.** Residual-stream maximum-to-median activation ratio versus kernel size k, averaged across depth using 128 held-out FineWeb sequences of 4096 tokens.

Table 1: **Downstream evaluation.** `lm-evaluation-harness` results at the best learning rate and convolution initialization per cell, chosen by held-out validation perplexity. Perplexities are word-level for WikiText and token-level for LAMBADA (↓); the accuracy columns are percentages (↑) with AVG their mean, and the recall columns are substring-match rates (↑), which are not included in AVG. LAMBADA is the OpenAI variant. Within each block the best value per column is **bold** and the second best underlined; ⋆ marks the best value in that column across the whole table. A dash marks a configuration with no trained run.

			Perplexity ↓		Accuracy (%) ↑							Recall (%) ↑		
		k	Wiki.	LMB.	LMB.	PIQA	Hella.	Wino.	ARC-e	ARC-c	AVG	SWDE	SQuAD	FDA
NoPE	Pre-norm	0	30.12	28.05	35.0	66.1	36.6	51.8	44.1	23.3	42.8	54.3	1.4	44.5
		1	29.27	27.72	35.3	66.4	36.5	**52.3**	44.6	**25.0**	43.4	**59.0**	0.2	51.6
		2	27.83	29.18	33.4	**68.4**	38.7	51.1	47.1	23.8	43.8	43.0	12.7	42.8
		4	**25.99**	**23.01**	**37.8**	68.3	**40.7**	50.6	**48.0**	24.3	**44.9**	12.5	19.8	15.9
		∞	28.25	29.84	32.7	67.6	38.7	51.3	47.3	23.3	43.5	57.8	**31.7**	**61.3**
	Post-norm	0	-	-	-	-	-	-	-	-	-	-	-	-
		1	33.90	46.71	29.1	65.2	33.6	50.0	44.6	22.3	40.8	42.9	0.1	36.8
		2	**27.13**	27.46	35.7	**68.6**	39.0	49.5	47.1	24.0	43.9	6.2	17.1	16.6
		4	27.41	30.95	33.8	68.0	39.4	52.2	**48.0**	23.7	44.2	6.1	15.1	16.1
		∞	27.58	**24.66**	**36.7**	67.3	**39.7**	**52.6**⋆	47.3	**24.4**	**44.7**	**60.8**	**22.5**	**60.3**
RoPE	Pre-norm	0	26.24	19.80	41.6	67.3	40.4	51.4	47.8	25.7	45.7	57.5	1.5	35.6
		1	26.51	20.62	40.9	68.5	39.9	**52.0**	47.7	24.3	45.6	59.8	3.4	41.6
		2	**25.14**⋆	**18.58**⋆	**42.3**⋆	69.0	**42.0**⋆	51.5	**48.4**⋆	**25.9**⋆	**46.5**⋆	16.0	20.5	20.3
		4	25.68	19.51	40.6	68.2	41.1	51.7	46.4	24.0	45.3	32.1	21.2	33.5
		∞	26.12	20.16	40.8	**69.3**⋆	40.9	50.6	**48.4**⋆	25.1	45.8	**61.8**⋆	**37.8**⋆	**63.8**⋆
	Post-norm	0	-	-	-	-	-	-	-	-	-	-	-	-
		1	28.69	29.64	35.4	68.0	38.7	49.8	47.0	23.8	43.8	55.1	3.4	38.7
		2	**26.07**	23.53	**40.6**	67.9	40.1	49.7	47.3	23.4	44.8	29.8	**34.6**	48.9
		4	26.36	23.98	38.3	**68.4**	40.3	51.0	**47.9**	24.3	45.0	7.7	21.1	14.8
		∞	26.35	**21.26**	40.2	67.2	**40.8**	**51.5**	47.6	**24.8**	**45.3**	**57.7**	33.3	**52.5**

$X^\top X$ is invertible. Choose

$$W_Q = \frac{x_\star^\top}{x_\star^\top x_\star}, W_K = (1, \ldots, L)(X^\top X)^{-1} X^\top, W_V = I_m. \tag{6}$$

The first choice gives $W_Q x_\star = 1$. Moreover,

$$W_K X = (1, \ldots, L)(X^\top X)^{-1} X^\top X = (1, \ldots, L), \tag{7}$$

so $W_K x_b = b$ for every $b = 1, \ldots, L$.

Use the size-two key filter $c_0 = 0$, $c_1 = 1$. Then $k_1^\pi = 0$ and $k_j^\pi = W_K x_{\pi_{j-1}} = \pi_{j-1}$ for $2 \le j \le L$. The values remain $v_j^\pi = x_{\pi_j}$. Therefore

$$z_\star(\pi) = \frac{x_{\pi_1} + \sum_{j=2}^{L} e^{\pi_{j-1}} x_{\pi_j}}{1 + \sum_{i=1}^{L-1} e^{\pi_i}}. \tag{8}$$

where the 1 at the denominator comes from $k_1^\pi = 0$. Equivalently, write $z_\star(\pi) = \sum_{b=1}^{L} w_b^\pi x_b$, where w_b^π is the coefficient of the fixed vector x_b, and

$$w_{\pi_1}^\pi = \frac{1}{1 + \sum_{i=1}^{L-1} e^{\pi_i}}, \qquad w_{\pi_j}^\pi = \frac{e^{\pi_{j-1}}}{1 + \sum_{i=1}^{L-1} e^{\pi_i}} \tag{9}$$

with the second formula holding for $2 \le j \le L$. By linear independence, for any permutations π, σ,

$$z_\star(\pi) = z_\star(\sigma) \iff w_b^\pi = w_b^\sigma \text{ for every } b = 1, \dots, L. \tag{10}$$

Suppose $z_\star(\pi) = z_\star(\sigma)$. By linear independence, the coefficient of each fixed vector x_b is the same in both outputs. Denote these common coefficients by

$$w_b = w_b^\pi = w_b^\sigma, \qquad b = 1, \dots, L. \tag{11}$$

We prove $\pi_j = \sigma_j$ at every position by induction. First, note that the coefficients $w_1, \dots, w_L$ are pairwise distinct: their numerators for π are $1, \exp(\pi_1), \dots, \exp(\pi_{L-1})$, with a common positive denominator. The uniquely smallest coefficient is at index π_1 for π and at index σ_1 for σ. Since the coefficient vectors are equal by construction, these indices must coincide:

$$\pi_1 = \sigma_1. \tag{12}$$

Now let $2 \le j \le L$ and suppose the permutations agree at all positions before j. The coefficient formulas give

$$w_{\pi_j} = w_{\pi_1} \exp(\pi_{j-1}), \qquad w_{\sigma_j} = w_{\sigma_1} \exp(\sigma_{j-1}). \tag{13}$$

In words, the coefficient at the current entry equals the smallest coefficient multiplied by the exponential of the preceding entry. The right-hand sides are equal because $\pi_1 = \sigma_1$ and $\pi_{j-1} = \sigma_{j-1}$. Therefore

$$w_{\pi_j} = w_{\sigma_j}. \tag{14}$$

Since distinct indices have distinct coefficients, this forces $\pi_j = \sigma_j$. By induction, we then have that $z_\star(\pi) = z_\star(\sigma)$ implies $\pi = \sigma$. Thus $z_\star$ is injective.

Upper bound: value-only filtering. Take $d = 1$, $d_v = m$ and choose the same projections as for key-only filtering:

$$W_Q = \frac{x_\star^\top}{x_\star^\top x_\star}, \qquad W_K = (1, \dots, L)(X^\top X)^{-1} X^\top, \qquad W_V = I_m, \tag{15}$$

where $X = (x_1, \dots, x_L)$ as above. We leave queries and keys unfiltered, so $q_{L+1}^\pi = 1$ and $k_j^\pi = \pi_j$. Filter only the values with size two: take c_0 to be the zero vector and every coordinate of c_1 to be 1. Then $v_1^\pi = 0$ and $v_j^\pi = x_{\pi_{j-1}}$ for $2 \le j \le L$. Hence

$$z_\star(\pi) = \sum_{j=2}^{L} \frac{\exp(\pi_j)}{\sum_{b=1}^{L} \exp(b)} x_{\pi_{j-1}} = \sum_{j=1}^{L-1} \frac{\exp(\pi_{j+1})}{\sum_{b=1}^{L} \exp(b)} x_{\pi_j}. \tag{16}$$

The denominator contains all L keys, since $\sum_{\ell=1}^{L} \exp(\pi_\ell) = \sum_{b=1}^{L} \exp(b)$. The zero value at position 1 removes its contribution to the output, but not its key from the denominator.

Equivalently, write $z_\star(\pi) = \sum_{b=1}^{L} w_b^\pi x_b$, where w_b^π is the coefficient of the fixed vector x_b, and

$$w_{\pi_L}^\pi = 0, \qquad w_{\pi_j}^\pi = \frac{\exp(\pi_{j+1})}{\sum_{b=1}^{L} \exp(b)}. \tag{17}$$

with the second formula holding for $1 \le j < L$.

Suppose $z_\star(\pi) = z_\star(\sigma)$. By linear independence, we again have

$$w_b^\pi = w_b^\sigma \text{ for every } b = 1, \dots, L. \tag{18}$$

Within each coefficient vector, all entries are distinct, and the unique zero occurs at index π_L for π and at index σ_L for σ. Equality of the coefficient vectors therefore implies $\pi_L = \sigma_L$.

Suppose $\pi \ne \sigma$, and let j be the last position where they differ. Then $j < L$ and $\pi_{j+1} = \sigma_{j+1}$. Hence,

$$w_{\pi_j}^\pi = \frac{\exp(\pi_{j+1})}{\sum_{b=1}^{L} \exp(b)} = \frac{\exp(\sigma_{j+1})}{\sum_{b=1}^{L} \exp(b)} = w_{\sigma_j}^\sigma. \tag{19}$$

But $w_{\sigma_j}^\sigma = w_{\sigma_j}^\pi$, so $w_{\pi_j}^\pi = w_{\sigma_j}^\pi$. This contradicts the distinctness of the coefficients, since $\pi_j \ne \sigma_j$.

Upper bound: query-only filtering. Take $d = 1$, $d_v = m$ and choose

$$W_Q = W_K = (1, \dots, L)(X^\top X)^{-1} X^\top, W_V = I_m. \tag{20}$$

Thus $W_Q x_b = W_K x_b = b$ for every $b = 1, \dots, L$. Filter only the queries, using size L with $c_0 = 0$ and $c_a = (L+1)^{a-1}$ for $1 \le a \le L-1$. The terminal query is

$$q^\pi_{L+1} = \sum_{a=1}^{L-1} (L+1)^{a-1} \pi_{L+1-a} = \sum_{j=2}^{L} (L+1)^{L-j} \pi_j. \tag{21}$$

The contribution of $x_\star$ vanishes because $c_0 = 0$. The keys and values are unfiltered, so $k^\pi_j = \pi_j$ and $v^\pi_j = x_{\pi_j}$. Therefore

$$z_\star(\pi) = \sum_{b=1}^{L} w^\pi_b x_b, w^\pi_b = \frac{\exp(b q^\pi_{L+1})}{\sum_{\ell=1}^{L} \exp(\ell q^\pi_{L+1})}. \tag{22}$$

Suppose $z_\star(\pi) = z_\star(\sigma)$. By linear independence, $w^\pi_b = w^\sigma_b$ for every $b = 1, \dots, L$. Taking the ratio of the coefficients at indices 2 and 1 cancels the common denominator:

$$\exp(q^\pi_{L+1}) = \frac{w^\pi_2}{w^\pi_1} = \frac{w^\sigma_2}{w^\sigma_1} = \exp(q^\sigma_{L+1}). \tag{23}$$

Since the exponential is injective on $\mathbb{R}$, $q^\pi_{L+1} = q^\sigma_{L+1}$. Thus

$$\sum_{j=2}^{L} (L+1)^{L-j} \pi_j = \sum_{j=2}^{L} (L+1)^{L-j} \sigma_j. \tag{24}$$

Both sides are base-$(L+1)$ representations, with digits $\pi_2, \dots, \pi_L$ and $\sigma_2, \dots, \sigma_L$, respectively. All digits lie in $\{1, \dots, L\}$, so uniqueness of this representation implies $\pi_j = \sigma_j$ for $2 \le j \le L$. Since both permutations contain each integer in $\{1, \dots, L\}$ exactly once, their remaining entries also satisfy $\pi_1 = \sigma_1$. Hence $\pi = \sigma$, proving injectivity.

C Experimental setup

This appendix describes every run and measurement behind the figures, in enough detail to repeat them.

C.1 Architecture

All models are the decoder-only Transformer defined in the preliminaries: $B = 24$ blocks, each an attention sublayer followed by a SwiGLU MLP, residual width $D = 1024$, 16 heads of dimension 64, and an MLP inner width of $4D$. Normalisation is RMSNorm with $\epsilon = 10^{-6}$, placed before the residual branch (pre-norm) or after the residual addition (post-norm). The vocabulary is the 32,000-token Mistral-7B tokenizer; input and output embeddings are untied. Weights are initialised from $\mathcal{N}(0, 0.02^2)$. There is no query–key normalisation, no bias in any projection or convolution, and no attention or MLP dropout. This gives $\approx$ 340M parameters. RoPE uses $\theta = 5 \times 10^5$ applied to all head dimensions; NoPE removes the rotation entirely, so the only positional signal is the causal mask.

Convolutions are depth-wise and causal, and are applied to the query, key and value projections and to the gate and up projections of the MLP, as shown in Fig. 7. A short convolution of size k uses k taps, where $k = 1$ degenerates to a per-channel gain; the long convolution uses the geometric parameterisation $h = (b_0, b_1 g, b_1 g^2, \dots)$ with $b_0, b_1 \sim \mathcal{N}(0, 0.02^2)$ and poles $|g| \sim \mathcal{U}[0.8, 0.9]$ with random sign. No activation is applied inside the convolution. Unless stated otherwise the same kernel is used on the attention and MLP branches.

C.2 Data

We train on FineWeb `sample-100BT`, tokenized into contiguous 4096-token blocks with the Mistral-7B tokenizer (seed 42). Training consumes the first 2,441,472 blocks, one epoch, in a fixed

order. The held-out window for validation perplexity is the $1{,}000$ blocks immediately following the last block seen in training, so it is disjoint from training for every run and identical across runs. The position-PCA analysis draws its calibration and evaluation sequences from a separate region of the corpus, disjoint from both.

C.3 Optimisation

Every run uses AdamW with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1 and gradient clipping at global norm 1.0, in bfloat16 with `torch.compile`. The schedule is a linear warmup over the first 10% of steps from 10^{-5} to the peak learning rate, then a cosine decay to 10^{-4}. Each run is $19{,}074$ steps at a global batch of 128 sequences of 4096 tokens, i.e. $524{,}288$ tokens per step and ≈ 10B tokens in total. We train on 16 A100 GPUs with a micro-batch of 2 and 4 gradient-accumulation steps. Checkpoints, including optimiser state, are written every 2048 steps; all reported numbers use the final checkpoint unless a figure names a step.

Note that the random seed is applied after the model is constructed, so runs that share a configuration differ in their initialization as well as their data order.

C.4 The main grid

Most figures draw on one sweep: position encoding $\in \{\text{NoPE}, \text{RoPE}\}$ $\times$ normalisation $\in \{\text{pre}, \text{post}\}$ $\times$ kernel $k \in \{0, 1, 2, 4, \infty\}$ $\times$ peak learning rate $\in \{1, 2, 4, 8, 16\} \times 10^{-3}$, with $k = 0$ meaning no convolution and $k = \infty$ the long convolution. Runs whose loss settles at the ≈ 1730 perplexity collapse value are reported as diverged and drawn as hollow markers at the top of the axis; we stop such runs early. One cell needs a deviation: NoPE $k = 1$ diverges above 2×10^{-3} unless the keys are mean-centred per sequence and head during training, which leaves the forward pass invariant, so those runs use it. RoPE $k = 1$ does not.

C.5 Convolution initialization

Fig. 10 varies the initialization of the convolution filters while holding everything else fixed: the default $\mathcal{N}(0, 0.02^2)$, the identity tap (all weight on the current token, so the model starts as the exact Transformer without convolutions), and $\mathcal{N}(0, \sigma^2)$ with $\sigma \in \{0.5, 1\}$. For the long convolution the identity tap means $b_0 = 1$, $b_1 = 0$. The learning-rate ranges differ between normalisations because the post-norm sweep was run at 1–4×10^{-3} and the pre-norm sweep at 2–8×10^{-3}.

C.6 Evaluation

Validation perplexity is the exponential of the mean next-token loss over the $1{,}000$ held-out blocks, and is what selects the best learning rate (and initialization, where the figure says so) in every figure and in Table 1(Appendix A.

Downstream numbers in Table 1 come from `lm-evaluation-harness` $0.4.8$, zero-shot, in bfloat16 at batch size 16: WikiText word-level perplexity, LAMBADA (OpenAI variant) perplexity and accuracy, PIQA, HellaSwag (normalized), WinoGrande, ARC-easy, ARC-challenge (normalized), and the substring-match recall tasks SWDE, SQuAD-completion and FDA. The recall tasks are sensitive to how sequences are batched and padded, so their absolute values should be compared only within a fixed batch size.

Length extrapolation (Fig. 4) evaluates models trained at 4096 tokens on sequences of $16{,}384$, reporting per-token cross-entropy as a 501-token moving average, on CodeParrot, TriviaQA, OpenThoughts-114k-Math and held-out FineWeb.

C.7 Per-figure measurements

Beyond training and the three evaluations above, each analysis figure needs one extra measurement.

- **Figs. 1, 8 and 10 and Table 1.** Validation perplexity, and for Table 1 the harness, at the final checkpoint. Fig. 8 additionally places the long convolution at the Canon positions, before and after the attention and MLP projections, in four combinations.

- **Fig. 9 (query and key gradients).** One model, the pre-norm NoPE Transformer without convolutions at its best learning rate. At each of three checkpoints we backpropagate the mean next-token loss over 32 held-out sequences in float32 and record the gradient norm of every projection weight.
- **Fig. 2 (rank collapse).** Mean pairwise cosine similarity between the token representations of a layer, averaged over the held-out sequences, computed in float32 at the final checkpoint of the best learning rate of each cell.
- **Fig. 3 (optimiser update).** Six pre-norm NoPE models at a common 8×10^{-3}. From the saved optimiser state at three checkpoints we form the AdamW update direction $\hat{m}/(\sqrt{\hat{v}} + \epsilon)$ with bias correction, sum it in quadrature over each block's tensors, and divide by the norm of that block's parameters. The scheduled learning rate is divided out, and weight decay is excluded. Panel (a) additionally needs validation perplexity at every saved checkpoint.
- **Fig. 5 (filter compression).** The two pre-norm $k = 4$ models, NoPE and RoPE. At inference time each of the four taps is either reassigned to a position of a shorter filter or dropped, with taps sharing a position summed; we enumerate all such patterns for $k' \in \{1, 2, 3\}$, apply them to one component at a time, and record the change in validation perplexity.
- **Fig. 14 (massive activations).** For each layer we take the largest $|h|$ over all channels and tokens of the residual stream and divide it by the median magnitude of ordinary tokens, a per-channel median followed by the median over channels. Both are averaged over 128 held-out blocks of 4096 tokens, and the ratio is averaged over the 24 layers.
- **Figs. 6 and 11 to 13 (position components).** For each model we fit PCA at every block output on the position-mean pattern of 128 calibration sequences: subtract the grand mean from the per-position mean representation and take the eigenvectors of the exact covariance, without normalising activations. On a disjoint evaluation set we then (i) measure the cosine between each token representation and a fitted direction, over 4096 sequences, and (ii) remove the component from the input of one projection — on a private copy of the residual, leaving the residual stream and the other projections untouched — either in every layer up to a depth (cumulative, Figs. 6 and 12) or in one layer only (Fig. 11), and record the next-token loss over 256 sequences. The control removes a random unit direction orthogonal to the first component. Intervals are paired 95% sequence bootstraps. Under RoPE the rotation is still applied after the projection, so these figures measure reliance on position information carried by the residual stream, not on the rotary encoding itself.