ROTARY ORIENTATION CONTRACTION IN ROPE

Anonymous authors
Paper under double-blind review

ABSTRACT

Rotary Position Embeddings (RoPE) encode relative position by rotating pairs of query and key coordinates according to token position. However, these pairs already have content-dependent orientations before the positional rotation is applied, so content-dependent orientation and positional phase interact within the same angular term of the attention score. We introduce Rotary Orientation Contraction (ROC), a simple modification to RoPE that learns how strongly to contract these pre-existing orientations while preserving the standard RoPE frequencies and positional rotations. Across language-modeling experiments, ROC improves perplexity over matched RoPE baselines. These results identify pre-RoPE content geometry as a useful design dimension for rotary attention and show that it can be modified efficiently, with essentially no measured forward-pass overhead, while preserving the standard paired RoPE structure.

1 INTRODUCTION

Rotary position embeddings (RoPE) encode relative position by rotating pairs of query and key coordinates according to token position (Su et al., 2024). These pairs are not orientation-free before the rotation is applied. A projected query or key pair already has an orientation determined by its projected coordinates, and the difference between query and key orientations enters the same angular interaction as the position-dependent rotation introduced by RoPE. Consequently, the contribution of a rotary pair to the attention score depends jointly on relative position and on the geometry of the projected query and key.

Prior analyses have shown that the geometry of projected queries and keys can materially affect how RoPE uses positional information (Zhu et al., 2024; Chen & Yan, 2024). Most closely related, Gopalakrishnan et al. (2026) introduce Polar Coordinate Positional Embedding (PoPE), which removes content-dependent phase using a different representation, motivated by disentangling content from relative position. We instead ask whether the pre-RoPE content orientation itself can be learned and modified within the standard paired RoPE representation. We introduce Rotary Orientation Contraction (ROC), which learns how strongly to contract these pre-RoPE orientations while preserving the standard paired RoPE structure, frequencies, and positional rotations. ROC recovers standard RoPE when no contraction is applied and removes the pre-RoPE orientation under complete contraction.

ROC improves language-modeling performance over matched RoPE baselines when combined with QK normalization. We additionally compare different geometries for realizing the contraction and find that Cartesian interpolation outperforms direct principal-angle contraction under matched conditions. We also examine how much flexibility is required in the contraction coefficients and find that coarse parameterizations capture most of ROC's improvement, with finer-grained coefficients providing only marginal additional gains. ROC introduces essentially no measured forward-pass overhead, while its modest training overhead decreases with model width. Together, these results suggest that learned orientation contraction can improve rotary attention while preserving the favorable computational structure of RoPE.

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.

2 ROTARY ORIENTATION CONTRACTION

2.1 CONTENT ORIENTATION IN ROPE

For a rotary pair x, let r denote its magnitude and ϕ its pre-RoPE orientation. RoPE rotates pair i at token position m by $m\omega_i$, so its orientation becomes

$$\phi + m\omega_i.$$

For a query at position m and key at position n, rotary pair i contributes

$$r_q r_k \cos\left(\phi_q - \phi_k + (m-n)\omega_i\right). \tag{1}$$

to their inner product. Thus, content-dependent relative orientation $\phi_q - \phi_k$ and relative-position phase $(m-n)\omega_i$ enter the same angular interaction.

2.2 ROC TRANSFORMATION

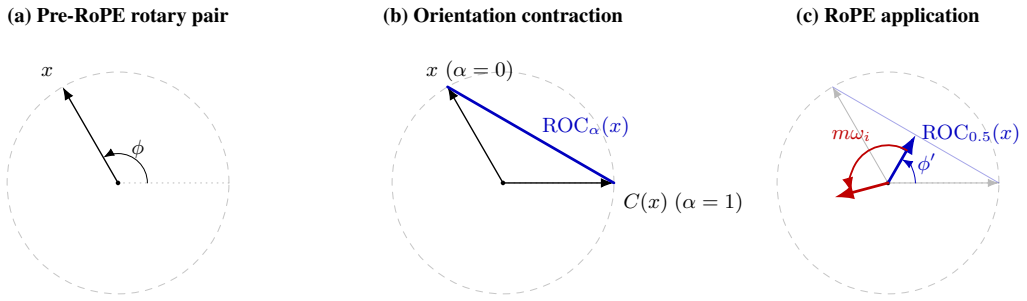

Figure 1: ROC first contracts the orientation of a rotary pair toward its canonical representation $C(x) = \|x\|e_1$, then the usual RoPE rotation is applied.

ROC gives the model explicit control over the pre-RoPE orientation of each rotary pair. For $x \in \mathbb{R}^2$, define its same-magnitude canonical representation by

$$C(x) = \|x\|e_1, \qquad e_1 = (1, 0).$$

For nonzero x, $C(x)$ has the same magnitude as x and zero orientation.

ROC interpolates linearly in Cartesian coordinates between the original representation and its canonical counterpart:

$$\mathrm{ROC}_\alpha(x) = (1-\alpha)x + \alpha C(x), \qquad \alpha \in [0, 1]. \tag{2}$$

For $x = (x_1, x_2)$ with $r = \|x\|$, this gives

$$z_1 = (1-\alpha)x_1 + \alpha r, \qquad z_2 = (1-\alpha)x_2. \tag{3}$$

At $\alpha = 0$, ROC is the identity and standard RoPE is recovered. At $\alpha = 1$, every nonzero pair is mapped to $(r, 0)$, completely removing its pre-RoPE orientation while preserving its magnitude at the endpoint.

Because ROC interpolates linearly in Cartesian coordinates, it follows the chord from x to $(r, 0)$ rather than a fixed-radius circular arc. Consequently, ROC generally changes both orientation and magnitude for intermediate values of α. By contrast, directly contracting the principal angle as $\phi \mapsto (1-\alpha)\phi$ preserves the magnitude and follows the corresponding circular arc. We compare these contraction geometries in Section 3.4.

Let r' and ϕ' denote the magnitude and orientation of the ROC-transformed rotary pair. If ROC transforms the query and key pairs to (r'_q, ϕ'_q) and (r'_k, ϕ'_k), respectively, then after the standard RoPE rotation, rotary pair i contributes

$$r'_q r'_k \cos\left(\phi'_q - \phi'_k + (m-n)\omega_i\right) \tag{4}$$

to their inner product. Thus, ROC modifies the content-dependent geometry presented to RoPE while leaving the relative-position phase $(m-n)\omega_i$, rotary frequencies, and positional rotations unchanged.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We pretrain Transformer language models (Vaswani et al., 2017) from scratch on FineWeb-Edu (Penedo et al., 2024) using a GPT-2 tokenizer (Radford et al., 2019) with a vocabulary of 50,257 tokens and a context length of 2048 tokens. We evaluate two Transformer scales corresponding to approximately 124M and 355M parameters, excluding the input token embedding matrix, trained for 4B and 10B tokens, respectively.

Our models use a pre-norm Transformer architecture. For ROC models, projected queries and keys are first transformed by ROC, then QK-normalized (Henry et al., 2020), with a separate learned gain for each query head, and finally rotated by RoPE using a positional frequency base of $10{,}000$. We additionally use gated attention (Qiu et al., 2025), with a per-head sigmoid gate applied to the attention output. The corresponding components are used consistently across our RoPE, ROC, and PoPE models; their interaction with ROC is examined separately in Section 3.5.

We use an effective batch size of 524,288 tokens per optimization step. Hidden matrix parameters are optimized with Muon (Jordan et al., 2024) using a peak learning rate of 1×10^{-2}, while all remaining parameters are optimized with AdamW (Loshchilov & Hutter, 2019) using a peak learning rate of 1×10^{-3}. ROC contraction parameters use a separate learning rate of 5×10^{-3}. We use a 2% linear warmup followed by cosine learning-rate decay to 1% of the peak learning rate and a weight decay of 0.01.

Unless otherwise specified, ROC learns one contraction coefficient for each rotary pair independently in every attention head and Transformer layer. Query and key representations within a pair share the same coefficient. The contraction coefficients are sigmoid-parameterized and initialized to $\alpha = 0.5$. Throughout, "ROC" denotes applying orientation contraction to the query and key representations before the standard RoPE positional rotation. Full architecture and training details are provided in Appendix A.

3.2 LANGUAGE MODELING PERFORMANCE

We first evaluate whether ROC and PoPE improve standard language modeling performance relative to RoPE. Table 1 reports validation perplexity on FineWeb-Edu at two model scales.

Table 1: Validation perplexity on FineWeb-Edu. Lower is better. Results for the 124M model are reported as mean $\pm$ standard deviation over three matched seeds.

Method	124M	355M
RoPE	19.30 ± 0.02	14.13
ROC	19.07 ± 0.02	14.00
PoPE	19.20 ± 0.04	14.05

Both ROC and PoPE improve over RoPE across the evaluated scales, suggesting that explicitly modifying the interaction between pre-RoPE content geometry and positional phase can improve rotary attention. In our setting, ROC achieves stronger language-modeling performance than PoPE.

3.3 CONTRACTION PARAMETERIZATION

We next examine how much flexibility is required in the learned contraction coefficients. We compare increasingly structured parameterizations that allow α to vary across Transformer layers, attention heads, and rotary pairs, as well as an untied variant with separate query and key coefficients. The resulting perplexities are reported in Table 2.

A single learned contraction coefficient per Transformer layer captures nearly all of ROC's improvement over RoPE. Finer variation across rotary pairs and attention heads provides only marginal additional gains, and allowing separate query and key contraction coefficients does not improve performance. Thus, ROC does not rely on a highly expressive contraction parameterization.

Table 2: Effect of contraction-coefficient parameterization at the 124M scale. The parameterizations vary in whether contraction coefficients are shared across attention heads, rotary pairs, and query–key representations within each Transformer layer. Lower perplexity is better.

Parameterization	Perplexity
No contraction (RoPE)	19.317
Per layer	19.090
Per layer + head	19.098
Per layer + rotary pair	19.088
Per layer + head + rotary pair	19.080
Per layer + head + rotary pair, separate Q/K	19.085

In the per-layer parameterization, contraction is generally stronger in the earlier and final layers, with learned coefficients ranging from 0.082 to 0.479 and a mean of 0.248.

Although finer-grained parameterizations provide little additional performance, their learned coefficients still exhibit systematic structure. Under our primary per-layer, per-head, per-pair parameterization, higher-frequency rotary pairs generally receive stronger contraction than lower-frequency pairs, with additional substantial variation across attention heads. We provide these learned-coefficient analyses in Appendix B.

3.4 Contraction Geometry

The primary ROC formulation contracts each rotary pair by interpolating in the Cartesian rotary plane. We next ask whether the observed benefit comes primarily from giving the model learnable control over pre-RoPE orientation, or whether the geometry used to realize that contraction also matters.

We compare ROC against a principal-angle contraction that instead operates directly on the polar orientation,

$$(r, \phi) \mapsto (r, (1 - \alpha)\phi)\,. \tag{5}$$

This transformation preserves pair magnitude and contracts the principal angle linearly toward zero. In contrast, ROC performs Cartesian interpolation between the pre-RoPE rotary vector and its same-magnitude canonical counterpart.

$$x \mapsto (1 - \alpha)x + \alpha C(x), \tag{6}$$

allowing both orientation and magnitude to change along the interpolating path. Figure 2 illustrates this geometric difference, including the contrasting behavior near the principal-angle branch cut.

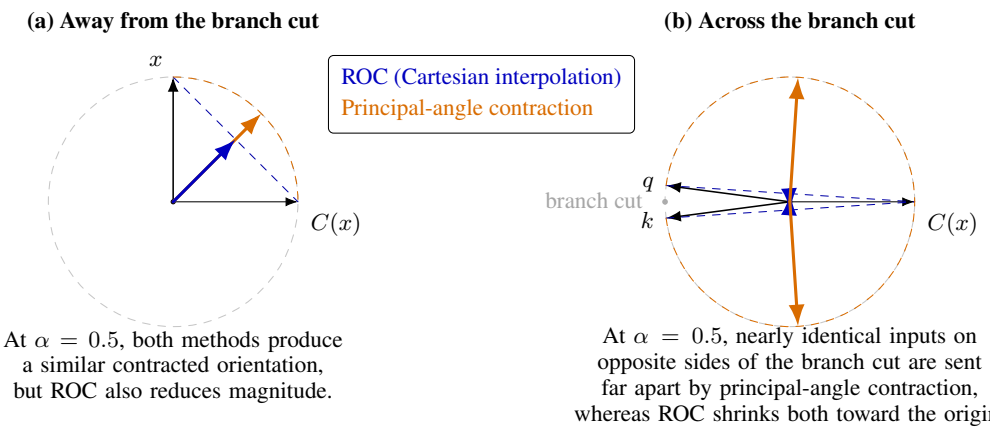

Figure 2: Comparison of ROC Cartesian interpolation and principal-angle contraction at $\alpha = 0.5$, away from and across the principal-angle branch cut.

Both variants learn contraction coefficients with the same parameterization, granularity, initialization, and shared query–key structure; they differ only in how a given contraction coefficient transforms the rotary pair. We compare both learned contraction formulations against a no-contraction RoPE baseline at the 124M and 355M scales in Table 3. The 124M results are averaged over three matched seeds.

Table 3: Comparison of contraction geometry across model scales. The 124M results are mean $\pm$ standard deviation over three matched seeds. Lower perplexity is better.

Contraction	124M	355M
No contraction (RoPE)	19.30 ± 0.02	14.13
Principal-angle contraction	19.17 ± 0.04	14.06
ROC (Cartesian interpolation)	19.07 ± 0.02	14.00

Principal-angle contraction improves over RoPE at both model scales, further supporting the view that learning to contract pre-RoPE orientation can improve rotary attention even when each rotary pair remains at fixed magnitude. ROC further improves over principal-angle contraction, showing that the way the contraction is realized also matters.

Principal-angle contraction follows a fixed-radius circular arc and is discontinuous at the principal-angle branch cut. ROC instead interpolates through the interior of the rotary plane, allowing orientation and magnitude to vary continuously along the contraction path. This attenuates representations near the antipodal ambiguity rather than mapping nearby inputs on opposite sides of the branch cut to widely separated outputs. We hypothesize that this continuous handling of the antipodal region contributes to ROC's additional improvement, although other effects of the associated magnitude variation may also play a role.

3.5 Interactions with the Attention Mechanism

ROC changes the geometry presented to attention and can therefore interact with other components of the attention mechanism. We examine this by progressively adding QK normalization and gated attention to an otherwise fixed architecture; the resulting perplexities are reported in Table 4.

Table 4: Interaction between ROC and attention components. Lower perplexity is better.

Configuration	RoPE	ROC	ROC $-$ RoPE
Base attention	19.62	19.65	+0.03
+ QK normalization	19.46	19.28	-0.18
+ QK normalization + gating	19.32	19.08	-0.24

QK normalization is an important component of the ROC configuration used throughout our main experiments. In our setting, orientation contraction provides no benefit without normalization, whereas combining it with QK normalization yields a clear improvement. One possible explanation is that reducing variation in the pre-RoPE angular component changes the balance between angular and magnitude-dependent contributions to the attention score, making control of the overall query and key norms more important.

This interaction is not unique to Cartesian ROC: we observe a similar dependence for principal-angle contraction, which preserves the magnitude of each rotary pair (Appendix A.2). This suggests that the benefit of QK normalization is not solely a consequence of ROC's induced pair-magnitude changes, but may instead reflect a more general synergy between orientation contraction and normalized query–key representations. Understanding the source of this interaction is an interesting direction for future work.

Gated attention provides an additional improvement. Unlike QK normalization, the gate acts after the attention distribution has been formed, controlling how strongly the resulting attention output enters the residual stream. Prior work has shown that such output gating can substantially reduce attention-sink behavior (Qiu et al., 2025) by allowing the model to suppress unhelpful retrieved

information. Its additional benefit under ROC may therefore indicate an interaction between orientation contraction and attention-sink behavior, although we do not investigate this connection further.

The ROC improvement is also comparable in magnitude to the gains from the other attention modifications considered here. For RoPE, adding QK normalization improves perplexity by 0.16, and subsequently adding gated attention improves it by a further 0.14. Replacing RoPE with ROC in the full QK-normalized and gated architecture yields an additional 0.24 improvement. Thus, under this training setup, ROC provides a gain of similar scale to these other attention modifications while remaining complementary to them: QK normalization and gated attention improve the RoPE baseline, and ROC provides a further improvement when used together with both. This is relevant given the increasing use of QK normalization and gated attention in modern Transformer architectures (Yang et al., 2025; Singh et al., 2026).

3.6 Efficiency

ROC preserves the standard query–key dimensionality and adds only a lightweight per-pair transformation before attention. In our implementation, `torch.compile` fuses ROC, QK normalization, and the subsequent RoPE rotation into the same compiled kernel, avoiding an additional intermediate materialization.

To measure practical overhead, we benchmark a representative Transformer block containing both attention and the MLP on an NVIDIA H200 GPU using a batch size of 16 and a sequence length of 2048 across increasing hidden dimensions. Table 5 reports normalized throughput relative to an otherwise identical RoPE block, with RoPE normalized to $1.00\times$ independently for each configuration. Values above $1.00\times$ indicate higher throughput than RoPE, while values below $1.00\times$ indicate lower throughput.

Table 5: Normalized throughput of ROC relative to RoPE on an NVIDIA H200 GPU across hidden dimensions. RoPE is normalized to $1.00\times$ independently for each configuration. Higher is better.

Hidden dimension	Forward	Forward + Backward
1024	$1.008\times$	$0.946\times$
2048	$1.000\times$	$0.954\times$
4096	$0.999\times$	$0.970\times$
8192	$0.999\times$	$0.986\times$

ROC introduces essentially no forward-pass overhead across the configurations tested. Forward-and-backward computation incurs a modest throughput reduction, but this overhead decreases steadily as model width increases, consistent with ROC's additional computation being increasingly amortized by the surrounding matrix operations.

We do not report a direct runtime comparison against PoPE because realized performance depends strongly on the attention implementation, and available implementations do not necessarily reflect the specialized fused implementation described by the PoPE authors. The methods nevertheless differ in their computational structure; we discuss these considerations in Section 4.

4 Related Work

Positional encoding. Because self-attention does not by itself encode sequence order, Transformers commonly incorporate an explicit mechanism for representing position. The original Transformer adds fixed sinusoidal encodings to token representations (Vaswani et al., 2017), while subsequent work has developed a variety of relative positional mechanisms (Shaw et al., 2018; Dai et al., 2019; Raffel et al., 2020). Rather than modifying token representations directly, ALiBi introduces fixed, head-specific linear distance biases into the attention scores (Press et al., 2022).

Rotary Position Embedding (RoPE) instead applies position-dependent rotations to query and key representations, causing their inner product to depend on relative displacement (Su et al., 2024).

ROC instead leaves the positional rotations unchanged and modifies the content-dependent geometry presented to RoPE before those rotations are applied.

Content–position interaction in RoPE. Recent work has examined how the learned geometry of queries and keys interacts with the fixed rotations imposed by RoPE. Zhu et al. (2024) identify undesirable behavior arising from the initial angular relationship between queries and keys and propose Collinear Constrained Attention (CoCA), which enforces a collinearity constraint between them to improve long-context extrapolation. Chen & Yan (2024) analyze how the geometry of paired query and key projections affects RoPE attention, showing that the angle between paired weight vectors influences the input sensitivity and relative-position behavior of the resulting rotary components. Barbero et al. (2025) study how different RoPE frequencies are used in a trained language model, finding that high-frequency components support robust positional attention patterns while lower-frequency components are used more heavily and may carry semantic information. These works show that the learned geometry and frequency structure on which RoPE acts can materially affect attention behavior.

Most closely related to ROC, Polar Coordinate Positional Embedding (PoPE) (Gopalakrishnan et al., 2026) identifies the content-dependent phase difference between queries and keys as a confound between content ("what") and relative position ("where"), and removes this interaction entirely. PoPE maps each scalar query or key feature to a nonnegative magnitude and assigns a content-independent phase determined by position, together with a learned componentwise phase bias. Consequently, PoPE constructs d complex components from a d-dimensional representation, rather than the $d/2$ paired components of standard RoPE.

ROC shares PoPE's focus on the interaction between content orientation and positional phase, but makes a different representational and computational tradeoff. ROC retains the standard paired RoPE structure and learns how strongly to contract the pre-RoPE orientation rather than removing it entirely. Complete contraction in ROC is therefore not equivalent to PoPE: at $\alpha = 1$, a conventional RoPE pair is mapped to $(r, 0)$, collapsing its two input coordinates to a single content-dependent magnitude, whereas PoPE avoids this collapse by assigning a separate magnitude and positional phase to each scalar feature.

This reparameterization increases the number of rotary components from $d/2$ to d, doubling the arithmetic cost of the query–key dot product. If the complex query and key representations are materialized in Cartesian form, this also doubles the corresponding representation storage and memory bandwidth. Avoiding explicit materialization of that expansion requires reconstructing the rotated representation within the attention computation instead. ROC retains the standard RoPE dimensionality and query–key dot product, adding only a lightweight transformation over the existing rotary pairs before the usual positional rotation.

ROC and PoPE can therefore be viewed as occupying different points in the tradeoff between content–position separation and computational efficiency: PoPE performs full decoupling through a more substantial reparameterization, while ROC modifies the content–position interaction within the standard computational structure of rotary attention.

5 LIMITATIONS

Our experiments are limited to two model scales and a single pretraining domain. Although ROC improves over RoPE at both scales we evaluate, it remains unclear whether these gains persist in substantially larger models, at larger pretraining budgets, or across other data domains.

ROC also interacts with surrounding architectural choices. In our experiments, its strongest performance is obtained with QK normalization and gated attention. QK normalization is comparatively straightforward to incorporate and is part of our default ROC implementation, where it is fused with both ROC and RoPE. Gated attention provides a further improvement, but is a more substantial architectural modification and is not yet a universal component of standard Transformer implementations. As a result, reproducing ROC's best observed performance may require adopting this additional attention component.

6 Conclusion

We introduced Rotary Orientation Contraction (ROC), a simple modification to RoPE that learns how strongly to contract pre-RoPE orientation before the standard positional rotation. Across the model scales considered in our main QK-normalized experiments, ROC consistently improves language-modeling perplexity over matched RoPE baselines. Direct principal-angle contraction also improves over RoPE, while ROC's Cartesian realization provides a further improvement under matched conditions. These results identify the content geometry presented to rotary attention as a useful design dimension, and show that modifying this geometry can improve language modeling while preserving the standard paired RoPE structure with essentially no measured forward-pass overhead and only modest additional training cost.

AI Use Statement

Generative AI tools were used for manuscript drafting and editing, figure creation, identifying relevant prior work, and assistance with development of the ROC implementation, including support for multiple contraction parameterizations and the custom backward implementation. All AI-assisted material was reviewed and verified by the authors.

Reproducibility Statement

The ROC transformation and its parameterization are specified in Section 2. Section 3 describes the experimental setup, and Appendix A provides model architectures, optimization hyperparameters, and implementation details for ROC, RoPE, and PoPE.

References

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes Rotary Positional Encodings useful? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 92408–92438, 2025.

Yiting Chen and Junchi Yan. What Rotary Position Embedding can tell us: Identifying query and key weights corresponding to basic syntactic or high-level semantic information. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 54507–54528. Curran Associates, Inc., 2024. doi: 10.52202/079017-1727.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285.

Anand Gopalakrishnan, Robert Csordás, Jürgen Schmidhuber, and Michael Curtis Mozer. Decoupling the "what" and "where" with Polar Coordinate Positional Embedding. In *Forty-third International Conference on Machine Learning*, 2026.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.379.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The FineWeb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Ofir Press, Noah Smith, and Mike Lewis. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*, 2022.

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 *Advances in Neural Information Processing Systems*, volume 38, pp. 110931–110957, 2025. doi: 10.52202/085713-3345.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Varun Singh, Lucas Krauss, Sami Jaghouar, Matej Sirovatka, Charles Goddard, Fares Obied, Jack Min Ong, Jannik Straube, Fern, Aria Harley, Conner Stewart, Colin Kealty, Maziyar Panahi, Simon Kirsten, Anushka Deshpande, Anneketh Vij, Arthur Bresnu, Pranav Veldurthi, Raghav Ravishankar, Hardik Bishnoi, DatologyAI Team, Arcee AI Team, Prime Intellect Team, Mark McQuade, Johannes Hagemann, and Lucas Atkins. Arcee trinity large technical report. *arXiv preprint arXiv:2602.17004*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, 2019.

Shiyi Zhu, Jing Ye, Wei Jiang, Siqiao Xue, Qi Zhang, Yifan Wu, and Jianguo Li. CoCA: Fusing position embedding with collinear constrained attention in transformers for long context window extending. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4247–4262, 2024. doi: 10.18653/v1/2024.acl-long.233.

A Additional Experimental Details

A.1 Model Architecture and Training

All experiments use decoder-only pre-norm Transformers with RMSNorm (Zhang & Sennrich, 2019) and SwiGLU (Shazeer, 2020) feed-forward blocks. The SwiGLU intermediate dimension uses an expansion factor of $8/3$ relative to the model dimension. Attention uses multi-head attention with a head dimension of 64. We apply QK normalization (Henry et al., 2020) independently within each attention head, with learned per-head gains, and apply a learned per-head sigmoid gate (Qiu et al., 2025) to the output of each attention head. Input and output token embeddings are untied. We use the GPT-2 tokenizer (Radford et al., 2019) with a vocabulary size of 50,257. RoPE (Su et al., 2024) uses a base frequency of 10,000.

Table 6: Model configurations used in our experiments. Parameter counts exclude the input token embedding matrix.

Model	Layers	d_{model}	Heads	Head dim.	Training tokens
124M	12	768	12	64	4B
355M	24	1024	16	64	10B

Models are trained with a context length of 2048 tokens and an effective batch size of 524,288 tokens per optimization step. Training uses bfloat16 automatic mixed precision and a global gradient-norm clipping threshold of 1.0.

Parameters corresponding to hidden weight matrices are optimized with Muon (Jordan et al., 2024) using a peak learning rate of 1×10^{-2} and 0.95 momentum. All remaining trainable parameters are optimized with AdamW (Loshchilov & Hutter, 2019) using a peak learning rate of 1×10^{-3} and $\beta_1 = 0.9$, $\beta_2 = 0.95$. ROC contraction parameters use AdamW with a separate peak learning rate of 5×10^{-3}, chosen heuristically to compensate for the local compression induced by the sigmoid parameterization near its initialization; we do not separately tune this learning rate. Weight decay is 0.01 and is applied only to parameters with at least two dimensions; ROC contraction parameters are also excluded from weight decay. We use a 2% linear warmup followed by cosine learning-rate decay to 1% of the peak learning rate.

We reserve 10M tokens of FineWeb-Edu as a held-out validation set and report perplexity on this split.

Unless otherwise specified, ROC learns one contraction coefficient for each rotary pair independently in every attention head and Transformer layer. Query and key representations within a pair share the same coefficient. We parameterize the coefficients through a sigmoid and initialize them to $\alpha = 0.5$. In our attention implementation, queries and keys are first projected, then transformed by ROC, QK-normalized, and finally rotated by RoPE. QK normalization is applied over the full head dimension; the normalized query of each attention head is additionally scaled by a learned scalar gain, initialized to one.

Our implementation uses a custom backward for the combined ROC, QK normalization, and RoPE transformation. The backward computation is evaluated in float32 under bfloat16 training, with the resulting query and key gradients cast back to the input dtype. For numerical stability, terms involving the reciprocal pair magnitude in the ROC backward use $\max(r, 10^{-6})^{-1}$. This stabilization affects only the backward computation and does not modify the forward transformation.

For PoPE, we use the formulation from the released implementation, including the learnable phase bias initialized to zero. We additionally apply QK normalization immediately after the softplus, followed by the same learned per-head query gain used in our ROC and RoPE models. We evaluate the full PoPE transformation outside bfloat16 autocast, as we found this gives slightly better performance in our setting.

A.2 Attention-Component Interactions Across Contraction Geometries

The interaction between orientation contraction and QK normalization is not unique to Cartesian ROC. We observe the same qualitative pattern for principal-angle contraction, which preserves the magnitude of each rotary pair. This suggests that the benefit of QK normalization is not solely a consequence of ROC's induced pair-magnitude changes, but may instead reflect a more general synergy between orientation contraction and normalized query–key representations.

Table 7: Interaction between contraction geometry and attention components. Lower perplexity is better.

Configuration	RoPE	Principal-angle	ROC	Principal-angle − RoPE
Base attention	19.62	19.64	19.65	+0.02
+ QK normalization	19.46	19.33	19.28	-0.13
+ QK normalization + gating	19.32	19.12	19.08	-0.20

B Learned Contraction Structure

Although finer-grained contraction parameterizations provide only modest additional performance over the coarse variants considered in Section 3.3, their learned coefficients exhibit clear non-uniform structure. We summarize this structure here for the primary per-layer, per-head, per-pair parameterization and for the coarser per-layer variant.

Figure 3 shows the mean learned contraction coefficient across attention heads as a function of Transformer layer and rotary frequency for the primary ROC parameterization.

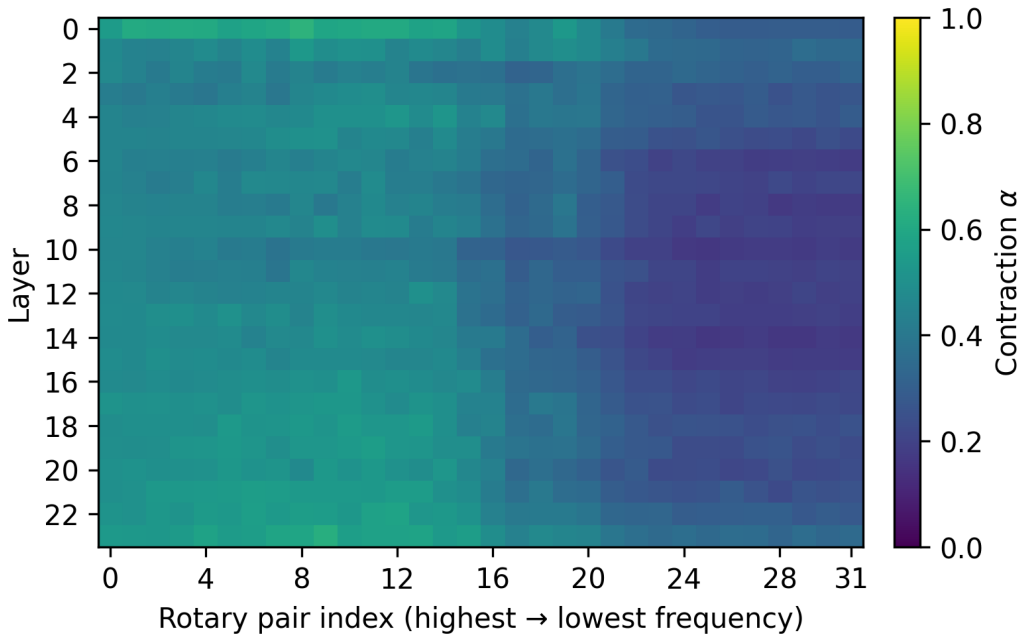

Figure 3: Mean learned contraction coefficient α across attention heads as a function of Transformer layer and rotary frequency for the primary ROC parameterization. Rotary pairs are ordered from highest frequency (left) to lowest frequency (right). Larger values correspond to stronger contraction.

The learned coefficients exhibit a clear frequency-dependent structure. Higher-frequency rotary pairs generally receive stronger contraction than lower-frequency pairs. A weaker depth-dependent pattern is also visible, with somewhat stronger contraction toward the early and final layers.

The same broad depth dependence is also visible under a much coarser parameterization. Figure 4 shows the learned coefficients for the per-layer ROC variant at the 124M scale, where each Transformer layer learns a single contraction coefficient shared across rotary pairs and attention heads.

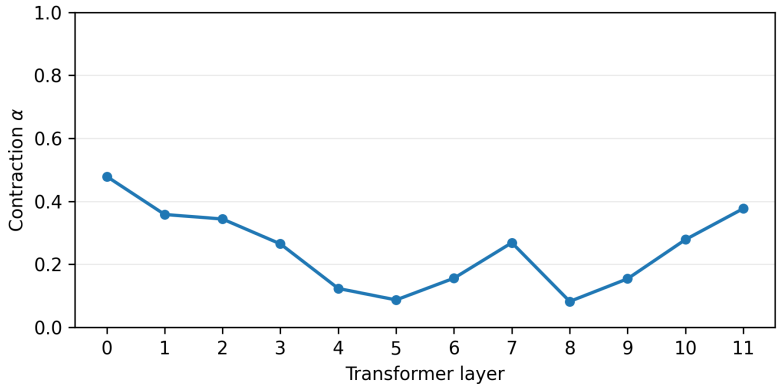

Figure 4: Learned contraction coefficients for the per-layer ROC parameterization at the 124M scale. Each layer learns a single coefficient shared across rotary pairs and attention heads.

The per-layer coefficients vary substantially across depth rather than converging to a common contraction strength. Contraction is generally stronger toward the early and final layers than across much of the middle of the network, although the pattern is not monotonic.

Finally, Figure 5 shows the minimum and maximum learned contraction coefficients across attention heads for each layer and rotary pair in the primary parameterization.

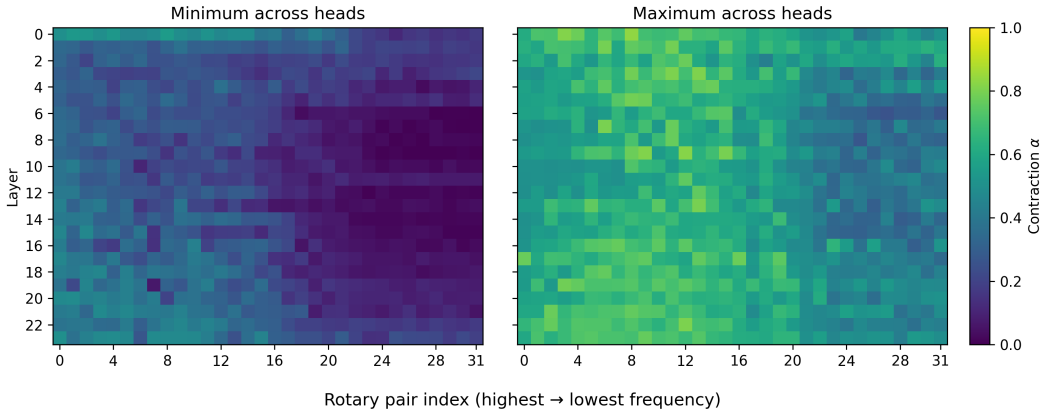

Figure 5: Headwise extrema of the learned ROC contraction coefficients. For each rotary pair and Transformer layer, we plot the minimum and maximum α value across attention heads.

The learned coefficients exhibit substantial head-to-head variation in addition to the broader frequency- and depth-dependent structure. Thus, while most of ROC's performance gain can be captured with much coarser contraction parameterizations, the primary parameterization makes systematic use of its additional flexibility.