

Modern Position Encoding in Transformers

RoPE/Yarn and PaTH

Songlin Yang

MIT CSAIL

 sustcsonglin.github.io/

Why Positional Information?

Self-attention **w/o causal mask** treats input tokens as an unordered set.

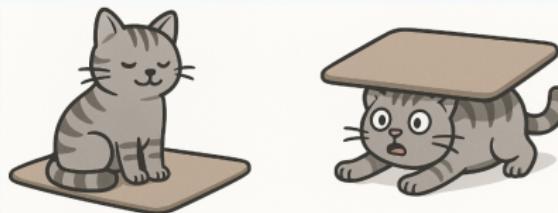


Figure: "The cat sat on the mat" \neq "The mat sat on the cat"

Without positional information, the model cannot distinguish word order.

Solution: Inject position information into token embeddings.

ROPE
○●○○○○○

YARN
oooooooooooo

PATH

PATH-FOX
0000000

REFERENCES

Absolute Positional Encoding

Vaswani et al. 2023 add sinusoidal position encoding to input embeddings:

$$X_{input} = \text{Embedding}(X) + \text{PE}$$

For position pos and dimension i :

$$PE_{(pos,zi)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i}/d_{model}}\right)$$

Rotary Position Embedding (RoPE)

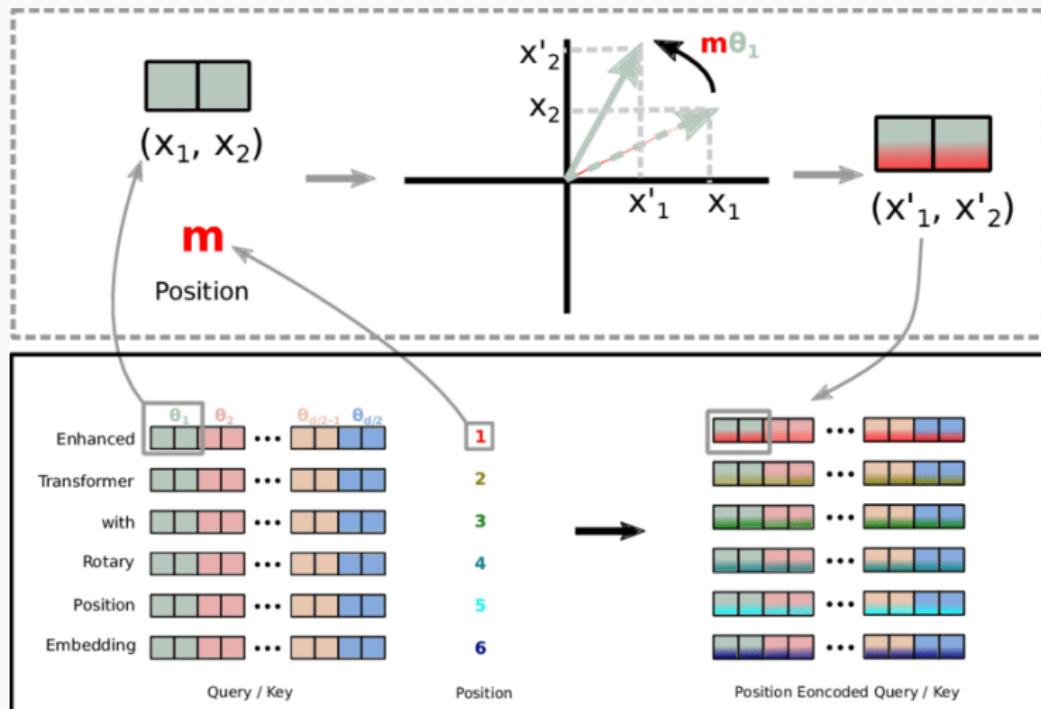


Figure: RoPE overview. The figure is from Su et al. 2023.

Rotation Matrix Fundamentals

Understanding rotation matrices is crucial for RoPE:

2D Rotation Matrix:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Key Properties:

- **Power property:** $R(\theta)^n = R(n\theta)$
- **Inverse:** $R^{-1}(\theta) = R(\theta)^T = R(-\theta)$
- **Composition:** $R(\theta_1) \cdot R(\theta_2) = R(\theta_1 + \theta_2)$

Block-Diagonal Matrix Form

RoPE divides channels into $d/2$ pairs, each with a different rotation frequency $\theta_m = 10000^{-2m/d}$ where m is the pair index.

$$R(\Theta) = \begin{bmatrix} R(\theta_0) & \circ & \circ & \cdots & \circ \\ \circ & R(\theta_1) & \circ & \cdots & \circ \\ \circ & \circ & R(\theta_2) & \cdots & \circ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \circ & \circ & \circ & \cdots & R(\theta_{\frac{d}{2}-1}) \end{bmatrix}$$

Block-diagonal structure maintains rotation properties:

$$R^m(\Theta) = R(m\Theta)$$

$$R^{-1}(\Theta) = R^{-T}(\Theta) = R(-\Theta)$$

$$R(\Theta_1) \cdot R(\Theta_2) = R(\Theta_1 + \Theta_2)$$

Apply RoPE to query and key:

$$q_i^{\text{RoPE}} = \mathcal{R}^i q_i, \quad k_j^{\text{RoPE}} = \mathcal{R}^j k_j \quad (\text{abbreviate } R^i(\Theta) \text{ as } R^i)$$

Then,

$$\begin{aligned} A_{ij} &\propto (q_i^{\text{RoPE}})^\top k_j^{\text{RoPE}} \\ &= q_i^\top (\mathcal{R}^i)^\top \mathcal{R}^j k_j \\ &= q_i^\top \mathcal{R}^{i-j} k_j \quad (\text{Relative position}) \end{aligned}$$

Takeaway: RoPE attention logit only depends on **relative position $i - j$** between query and key, making RoPE attention score invariant to absolute position.

ROUND AND ROUND WE GO! ⚙ WHAT MAKES ROTARY POSITIONAL ENCODINGS USEFUL?

Federico Barbero*
University of Oxford

Alex Vitvitskyi
Google DeepMind

Christos Perivolaropoulos
Google DeepMind

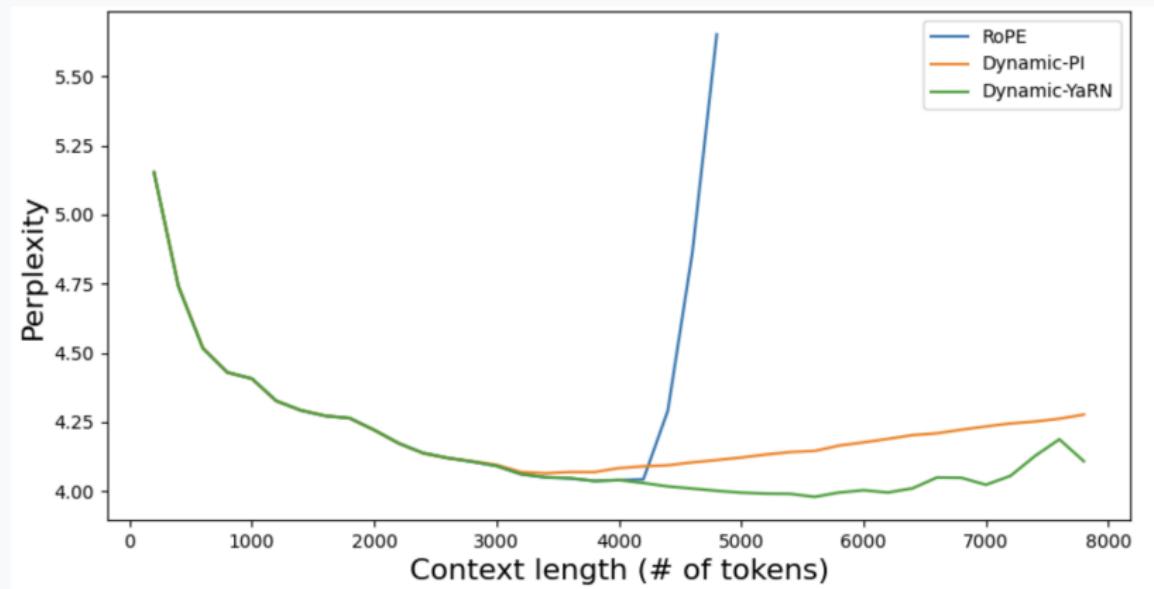
Razvan Pascanu
Google DeepMind

Petar Veličković
Google DeepMind

- **High-frequency channels ($\theta_m \uparrow$, fast rotation):**
 - For positional patterns (e.g., find the k -th nearest word).
 - NoPE cannot construct such patterns.
- **Low-frequency channels ($\theta_m \downarrow$, slow rotation):**
 - For semantic patterns.
 - Rotation has minimal impact on the dot-product.

This division allows RoPE to effectively balance positional and semantic information

Issue: Length Extrapolation



RoPE's PPL grows rapidly when tested on sequences longer than training length.

Position Interpolation (Chen et al. 2023)

$$q_i^{\text{RoPE}} = R^{i/s} q_i, \quad k_j^{\text{RoPE}} = R^{j/s} k_j$$

- L = original context length.
- L' = new context length.
- $s = L'/L$ = scale factor.

Every position i is interpolated to i/s .

NTK-aware Interpolation

NTK (Neural Tangent Kernel) theory suggests that deep neural networks struggle to learn high-frequency information when:

- The input dimension is low (e.g., position is 1-dimensional)
- The corresponding embeddings lack high-frequency components

NTK-aware Interpolation

For frequency scaling factor $\alpha(m)$, we need:

- High frequencies ($m \approx 0$): Keep original

$$f(\theta_d, L, L') = \theta_d \quad (\text{extrapolate})$$

- Low frequencies ($m \approx d/2$): Linear scaling

$$f(\theta_d, L, L') = \theta_d \cdot \frac{L}{L'} \quad (\text{interpolate})$$

ROPE
oooooooo

YARN
oooo●oooo

PATH
oooooooooooooooooooo

PATH-FOX
oooooooo

REFERENCES

NTK-aware Interpolation

To smoothly transition between these points $(0, 1)$ and $(d/2 - 1, L/L')$, NTK uses an exponential function to fit:

$$\alpha(m) = s^{2m/(d-2)}, \quad s = \frac{L'}{L}$$

High-Frequency Interpolation hurts short-range accuracy

- High-frequency encodes **fine-grained local order** (n-grams, local syntax).
- Scaling distorts these components \Rightarrow harms short-range accuracy.
- Keep high-frequency unchanged \Rightarrow preserve local discrimination.

ROPE
oooooooo

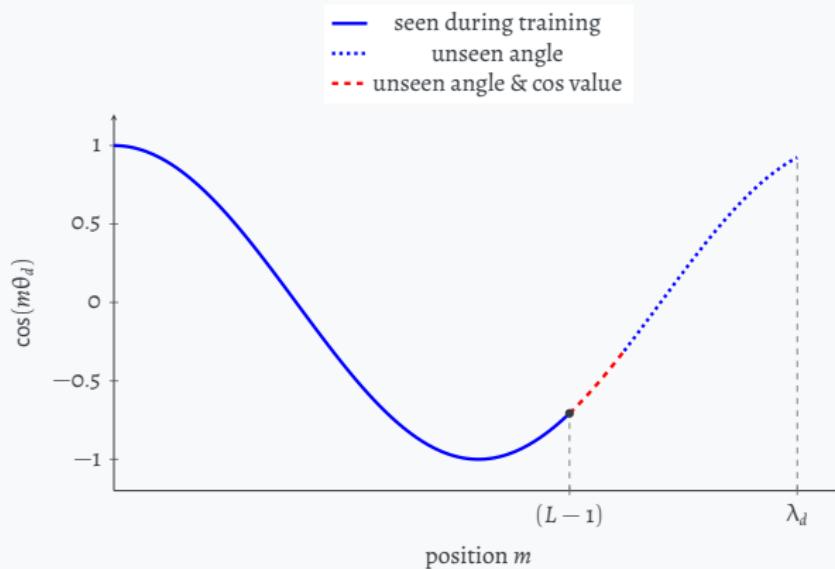
YARN
oooooooo●oo

PATH
oooooooooooooooooooo

PATH-FOX
oooooooo

REFERENCES

Low-Frequency extrapolation leads to OOD issues



NTK-by-parts

Define $r(d) = \frac{L}{\lambda_d}$ and the ramp function:

$$\gamma(r) = \begin{cases} 0, & r < \alpha, \\ 1, & r > \beta, \\ \frac{r-\alpha}{\beta-\alpha}, & \text{otherwise.} \end{cases}$$

Then the NTK-by-parts interpolation is

$$h(\theta_d) = (1 - \gamma(r(d))) \frac{\theta_d}{s} + \gamma(r(d)) \theta_d,$$

- $r(d) < \alpha$: linearly interpolate (like PI).
- $r(d) > \beta$: keep frequency unchanged.
- $\alpha \leq r(d) \leq \beta$: smooth transition.

YaRN (Peng et al. 2023)

YaRN combines NTK-by-parts with temperature scaling t :

$$\sqrt{1/t} = 0.1 \ln(s) + 1, \quad s = L'/L$$

where L' is the target length and L is the training length. The attention scores are computed as:

$$A = \text{softmax} \left(\frac{QK^T}{t\sqrt{d}} \right)$$

For more insights on entropy-invariant attention, see Jianlin Su's blog: <https://spaces.ac.cn/archives/8823>

ROPE
oooooooo

YARN
oooooooooo

PATH
●oooooooooooooooooooo

PATH-FOX
oooooooo

REFERENCES

Apart from the length extrapolation problem, RoPE Transformers have a fundamental expressivity limitation.

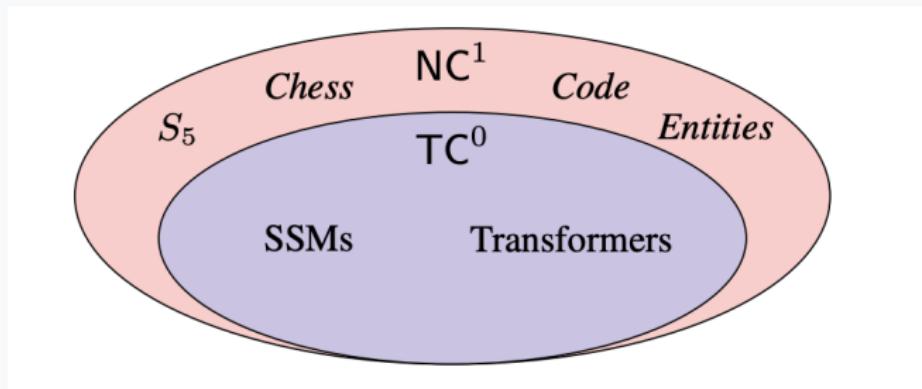


Figure: Figure is from Merrill, Petty, and Sabharwal 2025.

- Transformers with RoPE are within TC⁰ (Chen et al. 2024).
- Not able to solve complicated tasks like coding and entity tracking.

TC⁰ Overview

- **Circuits:** constant depth, polynomial size
- Gates: unbounded fan-in AND/OR/NOT + **threshold gates**
 - Threshold gate outputs 1 if $\sum x_i \geq t$
- **Intuition:** even shallow circuits can perform counting
- **Examples:** addition, comparison, prefix sums, parity, integer multiplication/division

NC¹ Overview

- **Circuits:** logarithmic depth $O(\log n)$, polynomial size
- Gates: bounded fan-in AND/OR/NOT
- **Intuition:** captures parallel divide-and-conquer / formula evaluation
- **Examples:** Boolean formula evaluation, permutation composition

Minimal NC¹-complete Problem: 5-element Permutations

- Consider **five elements** {1, 2, 3, 4, 5}.
- A **permutation** reorders these elements.
- **Swaps (transpositions)** are the building blocks: any permutation can be written as a sequence of swaps.
- **Problem:** given swaps s_1, s_2, \dots, s_m (each exchanging two of the five elements), compute the resulting permutation.
- This is exactly **permutation composition** in S_5 .

Why RoPE Fails (Intuition)

Task Requirements:

- **Data-dependent operations**
(e.g., $\text{swap}(A,B) \neq \text{swap}(C,D)$)
- **Non-commutative behavior:**
 - $\text{swap}(A,B) \rightarrow \text{swap}(B,C) \Rightarrow$
A ends at C
 - $\text{swap}(B,C) \rightarrow \text{swap}(A,B) \Rightarrow$
A ends at B

RoPE Limitations:

- **Position-only rotations**
(independent of token content)
- **Commutative angles**
 $(\theta_m + \theta_n = \theta_n + \theta_m)$

Fundamental Limitation

RoPE's commutative, block-diagonal structure
inherently operates at **TC^o complexity**

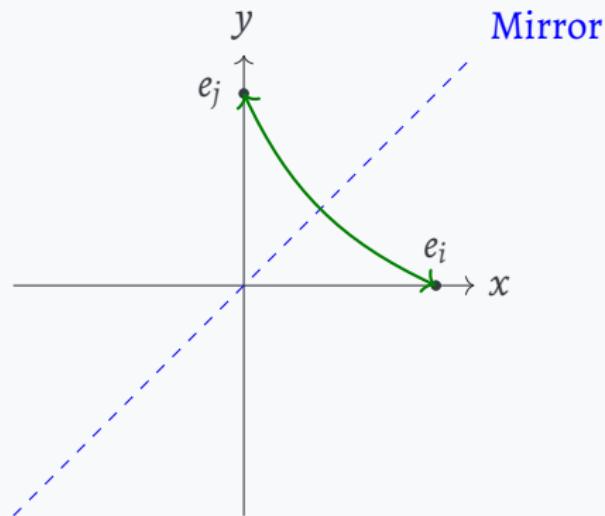
⇒ **Cannot handle non-commutative tasks**

Reflections \Rightarrow Swaps

Intuition: Householder = reflection (mirror). Reflection across proper axis \Rightarrow swap e_i, e_j .

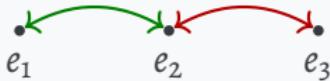
Math:

$$v = e_j - e_i, \quad H = I - 2 \frac{vv^T}{v^Tv}$$



Accumulated Reflections \Rightarrow Composition of Swaps

- One reflection \Rightarrow swap two coordinates
- Sequence of reflections \Rightarrow compose multiple swaps
- Accumulated effect: **permutations** on basis vectors
- Algebra: product of Householder matrices = product of swaps

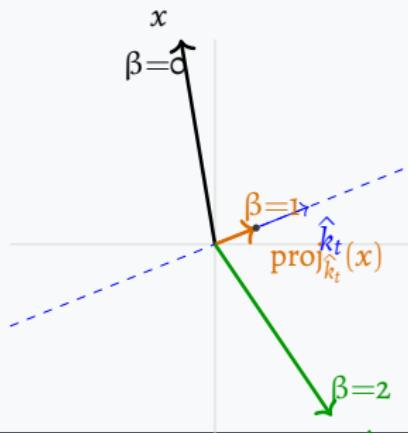


two reflections \Rightarrow composed permutation

Generalized Householder Transformations

$$H_t = I - \beta_t \hat{k}_t \hat{k}_t^\top \quad (\text{use } \hat{k}_t = k_t / \|k_t\|)$$

- $\beta_t = 0$: identity (do nothing)
- $\beta_t = 1$: projection onto $\text{span}(\hat{k}_t)$
- $\beta_t = 2$: reflection across the line spanned by \hat{k}_t



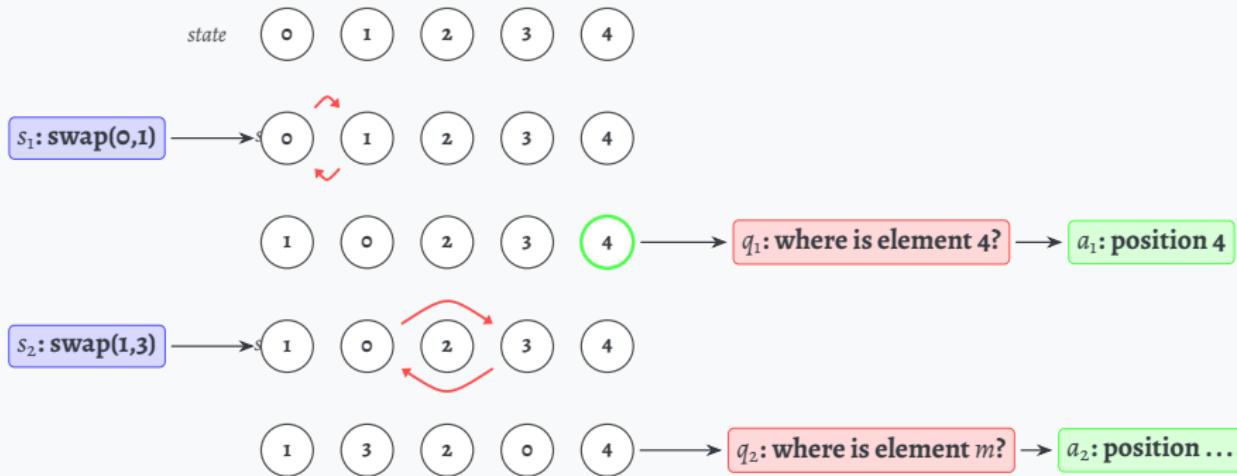
RoPE vs. PaTH

$$A_{ij} \propto \exp \left(\mathbf{k}_j^\top \left[\prod_{m=j+1}^i R(\Theta) \right] \mathbf{q}_i \right) \quad (\text{RoPE})$$

$$A_{ij} \propto \exp \left(\mathbf{k}_j^\top \left[\prod_{m=j+1}^i H_m \right] \mathbf{q}_i \right) \quad (\text{PaTH})$$

- RoPE: Data-independent rotation matrix $R(\Theta)$
- PaTH: Data-dependent generalized Householder matrix H_m
- PaTH is NC¹-complete under AC⁰ reduction (Yang et al. 2025)

Synthetic Task: Multiquery Swap-5



ROPE
oooooooo

YARN
oooooooooo

PATH
oooooooooooo●oooooooooooo

PATH-FOX
oooooooo

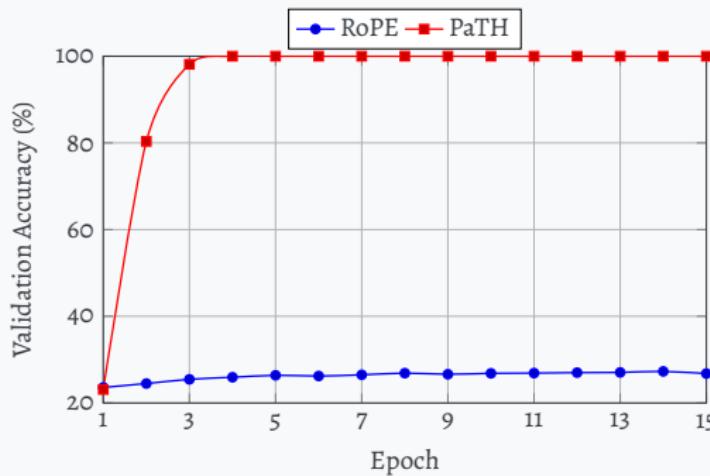
REFERENCES

Synthetic Task: Multiquery Swap-5

LM auto-regressive training based on the following sequence:

$$s_1, q_1, a_1, s_2, q_2, a_2, \dots, s_m, q_m, a_m$$

Results:



ROPE
ooooooooYARN
ooooooooooooPATH
oooooooooooo●ooooooooPATH-FOX
oooooooo

REFERENCES

Empirical Evidence: Flip-Flop Language Modeling (FFLM)

The Task

A diagnostic task for sequential reasoning. The model sees a sequence of actions (write, read, ignore) and must recall the last written bit at a ‘read’ token.

Error Rate (%) on FFLM

Method	ID	OOD	
	Sparse	Dense	
RoPE	6.9%	40.3%	0.01%
SBA	9.6%	38.9%	0%
FoX	8.3%	36.3%	0%
PaTH	0%	0.0001%	0%

1-layer, 2-head, 64-dim models.

We proved that PaTH provably solves FFLM in the paper.

ROPE
ooooooooYARN
ooooooooooooPATH
oooooooooooooooo●ooooooooPATH-FOX
oooooooo

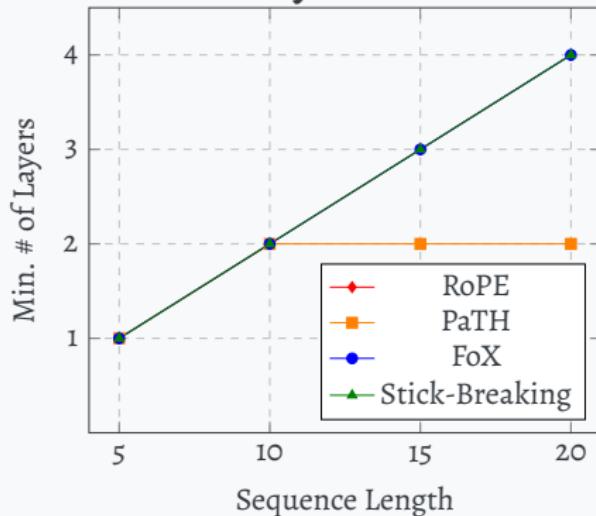
REFERENCES

Empirical Evidence: A_5 Word Problem

The Task

An NC^1 -hard problem testing algebraic reasoning. The model must determine if a sequence of group operations (e.g., $g_1 \cdot g_2 \cdot g_1^{-1}$) evaluates to the identity.

Minimum Layers to Solve



Cheap conversion from RoPE to PaTH

Since PaTH is more powerful than RoPE, it is possible to convert RoPE-based pretrained checkpoints into PaTH.

- Stage1: align per layer output l₂ distance.
- Stage2: minimize output logit KL divergence (i.e., Knowledge Distillation)

Results

100M tokens for stage1 and 3B tokens for stage2.

Task	Teacher (Qwen2.5-7B-Instruct)	Student (Distilled PaTH)
MMLU	74.21	73.28
Hellaswag	85.20	84.83
Winogrande	71.51	68.90
GPQA_Diamond	33.33	34.34
TheoremQA	18.12	21.88
GSM-8K	80.29	80.67
MATH	69.10	65.38
HumanEval	82.32	77.44
MBPP	74.71	75.10

Preliminary results. Stay tuned.

Continual Pretraining Results

Experiment 1: Mixed domain pretraining (12B tokens)

- Code (python-edu): 33%
- Math (MegaMath MathWebPro): 33%
- Text (DCLM): 33%

Model	GSM8K	HumanEval	MBPP+
PaTH	20.09	25.60	51.32
RoPE	19.86	23.10	47.09
FoX	15.54	21.30	48.15
SmolLM-2-1.7B (pre-decay)	8.60	16.40	38.62

Experiment 2: Math-focused pretraining (50B tokens)

- Math (MegaMath MathWebProMax): 100%

Model	GSM8K
-------	-------

The Challenge

In PaTH, the score for any query-key pair (i, j) depends on a cumulative product of matrices:

$$\text{score}(i, j) \propto k_j^\top \left(\prod_{s=j+1}^i H_s \right) q_i$$

A naïve implementation would be computationally intractable.

The Solution: A FlashAttention-Style Algorithm

We decompose the score into three parts:

$$\text{score}(i, j) \propto k_j^\top \prod_{s=j+1}^{\text{end}(B_j)} H_s \cdot \prod_{m=\text{block}(j)+1}^{\text{block}(i)-1} P_{[m]} \cdot \left(\prod_{s=\text{start}(B_i)}^i H_s \right) q_i$$

- **Key-side Transform:** Within key's block
 $\vec{K}_{[j]} = K_{[j]} - (T_{[j]}^{-1} \text{strictLower}(K_{[j]} W_{[j]}^\top))^\top W_{[j]}$
- **Inter-Block Transform:** Between blocks
 $P_{[m]} = I - W_{[m]}^\top T_{[m]}^{-1} W_{[m]}$
- **Query-side Transform:** Within query's block
 $\overleftarrow{Q}_{[i]} = Q_{[i]} - \text{lower}(Q_{[i]} W_{[i]}^\top) \cdot T_{[i]}^{-1} W_{[i]}$

where $T_{[j]}^{-1} = (I + \text{strictLower}(W_{[j]} W_{[j]}^\top))^{-1}$

FlashAttention-Style Algorithm

- Load $\overleftarrow{\mathbf{Q}}_{[i]}$ into SRAM.
- For key/value blocks $j = i - 1, \dots, 0$ (right-to-left scan):
 - Load $\overrightarrow{\mathbf{K}}_{[j]}, \mathbf{V}_{[j]}$, and $\mathbf{P}_{[j]}$ from HBM into SRAM.
 - Compute logits: $\tilde{\mathbf{A}}_{[i],[j]} = \overleftarrow{\mathbf{Q}}_{[i]} \overrightarrow{\mathbf{K}}_{[j]}^\top$.
 - Update online softmax statistics and accumulate output as in FlashAttention.
 - Update query: $\overleftarrow{\mathbf{Q}}_{[i]} \leftarrow \overleftarrow{\mathbf{Q}}_{[i]} \mathbf{P}_{[j]}^\top$.
- Normalize and store the output to HBM as in FlashAttention.

Efficient Inference with PaTH

- **In-place key update:** Historical keys are updated using the current timestep's transition matrix:

$$\mathbf{k}_i^{(t)} \leftarrow (\mathbf{I} - \beta_t \mathbf{w}_t \mathbf{w}_t^\top) \mathbf{k}_i^{(t-1)} \quad \text{for } i < t$$

eliminating the need to cache or recompute Householder products, where $\mathbf{k}_i^{(i)} = \mathbf{k}_i$.

- **Decoder compatibility:** This yields standard softmax-style decoding—compatible with **FlashDecoding**, **PagedAttention**, etc.

Forgetting Transformer

Forgetting Transformer (FoX) introduces a **data-dependent forget gate** f_s that additively modifies attention logits:

$$A_{ij} \propto \left(\prod_{s=j+1}^i f_s \right) \exp(k_j^\top q_i)$$

This allows the model to **forget stale memory** and improves **length generalization**.

PaTH-FoX

PaTH's state-tracking and FoX's forgetting gate can be naturally combined into a unified attention mechanism:

$$A_{ij} \propto \left(\prod_{s=j+1}^i f_s \right) \exp \left(k_j^\top \left(\prod_{s=j+1}^i H_s \right) q_i \right)$$

This hybrid mechanism is both **expressive** and **robust**, capable of generalizing far beyond training lengths.

PaTH-FoX in Context: An RNN Perspective

Analogy to RNN Families

- **NoPE:** Linear attention with softmax
- **FoX:** Mamba2-like model with softmax
- **PaTH:** DeltaNet with softmax
- **PaTH-FoX:** Gated DeltaNet with softmax

Combining the Best of Both Worlds

- **Softmax:** Enables precise long-range retrieval
- **Householder-like transition:** Enables hardware-friendly parallel state tracking
- **Forget gate:** Selectively decays stale memory, enhancing extrapolation

Experimental Setup

- 760M models, 24 layer, 1536 hidden dim, head dim 64.
- 50B FineWeb edu tokens.
- All models re-trained from scratch using the same experimental setup.

ROPE
ooooooooYARN
ooooooooooooPATH
ooooooooooooooooooooPATH-FOX
oooo●oo

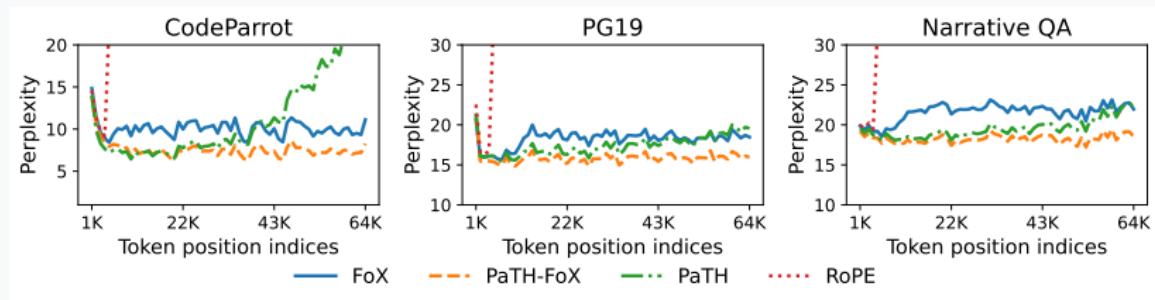
REFERENCES

Commonsense Reasoning and Language Modeling Results

Model	Wiki.	LMB.	LMB.	PIQA	Hella.	Wino.	ARC-e	ARC-c	Avg.
	ppl ↓	ppl ↓	acc ↑	acc ↑	acc _n ↑	acc ↑	acc ↑	acc _n ↑	↑
RoPE	19.01	19.77	40.4	70.2	50.3	54.9	67.2	33.3	52.7
FoX	18.33	18.28	41.7	70.8	50.9	57.1	65.7	32.6	53.1
PaTH	<u>18.03</u>	<u>16.79</u>	<u>44.0</u>	70.5	<u>51.5</u>	56.0	68.9	34.4	54.2
PaTH-FoX	17.35	16.23	44.1	70.8	52.2	57.1	<u>67.3</u>	<u>33.9</u>	54.2

Best in **bold**, second-best underlined. LM: perplexity on Wiki + LAMBADA; others: zero-shot accuracy.

Length Extrapolation



- Evaluated on three domains: **PG-19** (books), **CodeParrot** (code), and **NarrativeQA** (QA).
- **PaTH-FoX** maintains lowest perplexity throughout, especially on code, where state tracking is crucial.
- Highlights the benefit of **data-dependent encoding** and the **forgetting mechanism** for long-context generalization.

Long-context benchmarks

Model	RULER			BABILONG			PhoneBook			LongBench-E			
	4K	8K	16K	oK	4K	8K	16K	2K	4K	8K	4K	8K	16K
RoPE	35.7	1.3	0.0	33.0	13.8	0.0	0.0	32.3	15.6	0.0	18.7	3.7	2.0
FoX	41.6	29.5	4.9	23.8	20.2	8.2	4.4	62.5	38.5	17.7	23.4	16.9	11.7
PaTH	44.6	34.8	18.7	33.8	24.6	16.8	11.6	55.2	20.8	0.0	27.2	22.5	14.4
PaTH-FoX	42.3	34.0	22.6	28.6	25.6	19.2	10.0	89.6	93.8	66.6	23.4	21.8	16.1

Table: Summary of average scores on long-context tasks for 760M models with training length 4096.

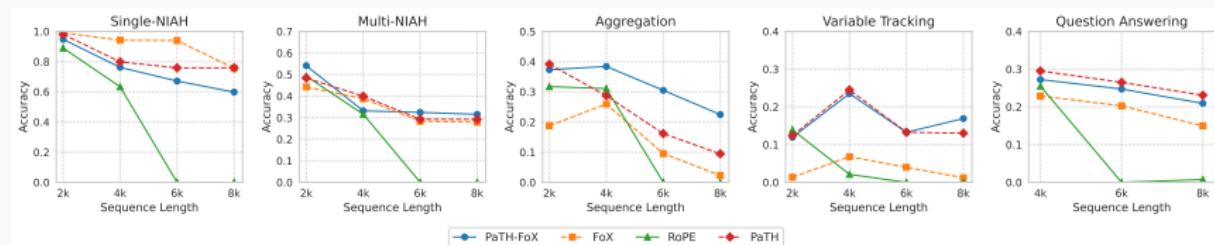


Figure: Task performance decomposition on RULER.

References I

-  Chen, Bo et al. (2024). *Circuit Complexity Bounds for RoPE-based Transformer Architecture*. arXiv: 2411.07602 [cs.LG]. URL: <https://arxiv.org/abs/2411.07602>.
-  Chen, Shouyuan et al. (2023). *Extending Context Window of Large Language Models via Positional Interpolation*. arXiv: 2306.15595 [cs.CL]. URL: <https://arxiv.org/abs/2306.15595>.
-  Merrill, William, Jackson Petty, and Ashish Sabharwal (2025). *The Illusion of State in State-Space Models*. arXiv: 2404.08819 [cs.LG]. URL: <https://arxiv.org/abs/2404.08819>.
-  Peng, Bowen et al. (2023). *YaRN: Efficient Context Window Extension of Large Language Models*. arXiv: 2309.00071 [cs.CL]. URL: <https://arxiv.org/abs/2309.00071>.

References II

-  Su, Jianlin et al. (2023). *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv: 2104.09864 [cs.CL]. URL: <https://arxiv.org/abs/2104.09864>.
-  Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.
-  Yang, Songlin et al. (2025). *PaTH Attention: Position Encoding via Accumulating Householder Transformations*. arXiv: 2505.16381 [cs.CL]. URL: <https://arxiv.org/abs/2505.16381>.