RETHINKING NON-ISOMETRIC POSITIONAL TRANSFORMATIONS THROUGH RANK AND STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Positional encoding is increasingly formulated through structured transformations of token interactions, where preserving near-isometric geometry is often preferred to avoid contraction or degeneracy. Inspired by state-transition design in linear attention, we revisit whether non-isometric deformation should instead be treated as a controllable source of positional capacity. We introduce **positional update rank**, $r_{\text{pos}}(P) = \text{rank}(P^\top P - I)$, and propose **PRISM**, which separates **expansion order** from **expansion structure** through independent, mixed, and product expansions. Our analysis shows that equal expansion order can yield substantially different operators depending on structure. Experiments across model scales, long-context tasks, and hybrid architectures reveal a consistent moderate-rank regime: expanding beyond rank one is often beneficial, whereas further expansion is not uniformly helpful. These results suggest that non-isometric positional deformation is not merely distortion, but a structured capacity whose utility depends on both its extent and organization.

1 INTRODUCTION

Transformers rely on positional mechanisms to control interactions across sequence positions (Vaswani et al., 2017; Shaw et al., 2018; Su et al., 2024). As positional encoding has evolved from additive signals toward structured transformations (Vaswani et al., 2017; Shaw et al., 2018; Su et al., 2024; Sun et al., 2023; Movahedi et al., 2026; Zhang et al., 2026b; Zhang, 2026; Yang et al., 2025b), orthogonal or near-isometric operators have remained an influential design choice (Su et al., 2024; Movahedi et al., 2026; Zhang et al., 2026b), since departures from isometry may introduce contraction, amplification, or near-singular behavior. Recent work therefore increasingly studies positional encoding through the geometry of its induced transformations (Su et al., 2024; Movahedi et al., 2026; Zhang et al., 2026b; Zhang, 2026; Gu et al., 2026). We argue that this view is incomplete: non-isometric deformation changes not only how information is preserved, but also what transformations are available for mediating interactions between positions. As illustrated in Fig. 1, such deformation acts along selected representation directions rather than merely introducing undifferentiated distortion.

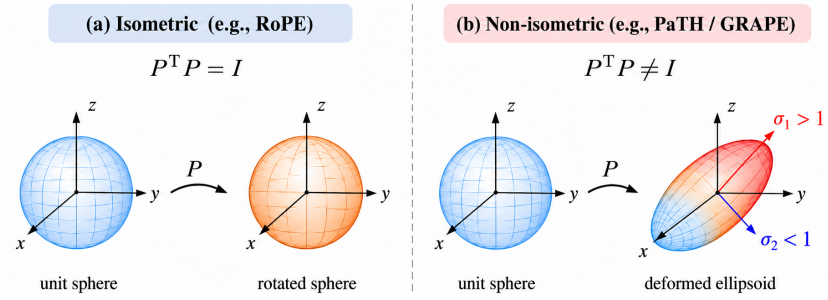


Figure 1: Isometric vs. non-isometric positional transformations.

Linear attention (Yang et al., 2023; 2024; Siems et al., 2025; Zhang et al., 2026a) provides a useful analogy: the rank and structure of state-transition matrices are increasingly treated as sources of

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.

capacity, rather than only distortion. This motivates our central question: **can non-isometric positional deformation likewise provide useful capacity, and how should that capacity be structured?**

For a positional transition $P_t \in \mathbb{R}^{D\times D}$, we define the **positional update rank**

$$r_{\mathrm{pos}}(P_t) = \mathrm{rank}(P_t^\top P_t - I),$$

which counts the singular modes that depart from isometry. RoPE (Su et al., 2024) has $r_{\mathrm{pos}} = 0$, while PaTH (Yang et al., 2025b) has $r_{\mathrm{pos}} = 1$. We treat this local deformation dimensionality as an explicit design variable and study the effect of systematically expanding it.

Rank alone, however, does not determine how this capacity is realized. Recurrent state transitions exhibit the same distinction: diagonal operators can modify many coordinates independently (Yang et al., 2023), whereas low-rank non-diagonal updates can couple coordinates through shared directions (Yang et al., 2024). More generally, multiple modes may remain independent, interact through mixing, or compose sequentially (Siems et al., 2025; Zhang et al., 2026a). We therefore distinguish **expansion order**, which controls the number of deformation modes, from **expansion structure**, which controls how they interact.

We introduce **PRISM: Positional Rank-expansion in Structured Modes**. **PRI-I** keeps transition modes independent, **PRI-M** couples them through a learned mixing matrix, and **PRI-P** composes rank-one transitions multiplicatively. These variants separate expansion order from the structure through which the available deformation modes are organized.

Across experiments, a consistent picture emerges. Moderate rank expansion improves information-intensive and long-context performance, while further expansion is not uniformly beneficial. Moreover, equal expansion order can lead to substantially different behavior under different structures, and these gains are not reproduced by simply increasing head capacity. These results suggest a broader perspective on positional encoding: beyond geometric form, positional transformations should also be understood through the dimensionality and organization of the non-isometric deformation they induce.

2 Related Work

2.1 Positional Transformations and Deformation Rank

Positional encoding has progressively moved from explicit positional signals toward structured transformations of token interactions (Vaswani et al., 2017; Shaw et al., 2018; Su et al., 2024). RoPE represents relative position through orthogonal rotations (Su et al., 2024), while subsequent methods extend this operator view through scaling (Sun et al., 2023), input dependence (Movahedi et al., 2026), more general group actions (Zhang et al., 2026b), non-semisimple operators (Zhang, 2026), and token-dependent transformations (Yang et al., 2025b). Representative examples include XPos, Selective RoPE, GRAPE, Jordan-RoPE, and PaTH.

Recent analyses also suggest that the behavior of positional mechanisms cannot be fully characterized by a single decay or spectral property. For RoPE, distance-dependent decay alone does not explain its effectiveness (Barbero et al., 2024), while explicitly removing long-term decay can improve long-context awareness and extrapolation (Chen et al., 2025). Other studies further show that different frequency bands play distinct roles in interpolation and extrapolation (Oka et al., 2026), and that long-context behavior depends on the geometry and spectral organization induced by positional transformations (Wertheimer et al., 2026; Hengyu, 2026). These observations motivate a more structural view of positional operators.

Table 1 summarizes representative mechanisms through the positional update rank defined in Section 1. Existing positional operators (Su et al., 2024; Sun et al., 2023; Zhang et al., 2026b; Zhang, 2026; Yang et al., 2025b) already span different deformation ranks, but rank is typically tied to the underlying operator family. Orthogonal, scaling-based, Jordan, additive, and Householder-like constructions realize deformation through fundamentally different structures. As a result, existing comparisons do not isolate positional update rank as an independent architectural variable.

Importantly, r_{pos} is not a scalar measure of overall positional expressivity. It characterizes how many directions can depart from isometry, but not how those directions are organized or coupled.

Table 1: Representative positional mechanisms viewed through positional update rank. Reported ranks correspond to generic non-degenerate full-head operators; detailed derivations, including the per-block interpretation of Jordan-RoPE, are provided in Appendix A.1.

Method	Original formulation	r_{pos}
RoPE (Su et al., 2024)	$\mathrm{blockdiag}_{j=1}^{D/2} \begin{pmatrix} \cos(n\theta_j) & -\sin(n\theta_j) \\ \sin(n\theta_j) & \cos(n\theta_j) \end{pmatrix}$	0
XPos (Sun et al., 2023)	$\mathrm{blockdiag}_{j=1}^{D/2} e^{\pm \xi_j n} \begin{pmatrix} \cos(n\theta_j) & -\sin(n\theta_j) \\ \sin(n\theta_j) & \cos(n\theta_j) \end{pmatrix}$	D
PaTH (Yang et al., 2025b)	$I - \beta_t w_t w_t^\top$	1
GRAPE-AP (Zhang et al., 2026b)	$I - \psi_h(t,\ell)E$	2
Jordan-RoPE (Zhang, 2026)	$\mathcal{R}\big[\exp\big(n\big((-\gamma + i\omega)I + \eta N_m\big)\big)\big], \quad N_m^m = 0$	D

A high-rank operator may remain strongly constrained by its structure, while a lower-rank operator may introduce richer interactions among representation directions.

PaTH (Yang et al., 2025b) provides a particularly clean setting for separating these factors. Its elementary transition has $r_{\mathrm{pos}} = 1$, while repeated transitions can still induce a rich global positional operator. PaTH therefore provides a controlled testbed for varying deformation order and structure explicitly, while positional update rank itself is defined independently of this particular operator family.

2.2 Transition Rank and Structure in Linear Attention

A related operator perspective appears in linear attention (Yang et al., 2023; 2024). A recurrent state can be written in the generic form

$$S_t = A_t S_{t-1} + k_t v_t^\top, \qquad o_t = q_t S_t,$$

where the transition matrix A_t determines how information stored at earlier positions is transformed before contributing to later positions. Although state transitions and positional transformations arise through different computational mechanisms, both govern how information is transformed between sequence positions.

Linear attention also makes clear that transition extent and transition structure are distinct properties. For example, GLA (Yang et al., 2023) and DeltaNet (Yang et al., 2024) use fundamentally different transition forms:

$$A_t^{\mathrm{GLA}} = \mathrm{Diag}(\alpha_t), \qquad A_t^{\mathrm{Delta}} = I - \beta_t k_t k_t^\top.$$

The GLA transition can modify many state dimensions simultaneously, but remains coordinate-wise separable (Yang et al., 2023). DeltaNet instead uses a rank-one deviation from identity, yet its non-diagonal outer-product structure couples state dimensions through a data-dependent direction (Yang et al., 2024). Thus, the number of affected directions alone does not determine the structure of a transition.

Recent work further explores how transition capacity can be expanded. Head-in-Head introduces multiple transition modes together with explicit interactions among them, providing a parallel route to higher-capacity transitions (Mei et al., 2026). DeltaProduct instead composes multiple Householder-like updates within each token step, providing a multiplicative route to expansion (Siems et al., 2025). These two directions motivate the structures studied in PRISM: expansion order controls how many modes are available, while expansion structure controls how those modes interact.

3 Positional Update Rank and PRISM

3.1 What Positional Update Rank Measures

For a positional transition $P \in \mathbb{R}^{D\times D}$, we define the metric-deformation operator $E(P) = P^\top P - I$ and the positional update rank $r_{\mathrm{pos}}(P) = \mathrm{rank}\, E(P)$. Unlike parameter count or transformation

magnitude, r_{pos} measures the number of representation directions whose metric can be modified by the positional operator.

Deformation modes. Let $\sigma_i(P)$ denote the singular values of P. Since the eigenvalues of $E(P)$ are $\sigma_i^2(P) - 1$, we have $r_{\mathrm{pos}}(P) = \#\{i : \sigma_i(P) \neq 1\}$. Therefore, update rank counts the number of non-isometric deformation modes. An operator with $r_{\mathrm{pos}}(P) = k$ can modify the metric on at most a k-dimensional subspace, while the remaining directions preserve their original metric. Importantly, r_{pos} characterizes the support of metric deformation rather than the overall expressivity of the positional transformation.

Rank as positional capacity. The update rank determines how many independent deformation modes can be represented. Given a target transition $P^\star$, let $S^\star = P^{\star\top} P^\star - I$. Any operator with $r_{\mathrm{pos}}(P) \leq r$ can only approximate $S^\star$ with rank at most r. According to the Eckart–Young theorem (Eckart & Young, 1936),

$$\|E(P) - S^\star\|_F \geq \left(\sum_{i>r} \mu_i^2 \right)^{1/2},$$

where μ_i are the singular values of $S^\star$. Thus, when the target deformation contains more than r independent components, a rank-constrained operator cannot represent all of them independently. Increasing r_{pos} therefore enlarges the available deformation capacity.

Why rank expansion may help. Positional transformations may need to represent heterogeneous relations, such as local interactions, long-range dependencies, and structural patterns. With only a limited number of deformation modes, these effects may compete within the same low-dimensional subspace. Rank expansion provides additional independent directions and may therefore reduce such competition.

This capacity argument does not imply that larger rank always improves performance. Additional modes may be redundant or poorly coordinated, and their usefulness depends on how they are organized. We examine this hypothesis first in the controlled MQRAR-N (Arora et al., 2024; Yang et al., 2025b) setting and then in large-scale language models. This motivates studying not only the expansion order but also how expanded modes are structured.

Composition. For sequential positional transitions, the accumulated deformation satisfies $E(AB) = B^\top E(A) B + E(B)$, yielding $r_{\mathrm{pos}}(AB) \leq r_{\mathrm{pos}}(A) + r_{\mathrm{pos}}(B)$. Thus, local low-rank updates can produce a richer global deformation, while the final effective rank depends on the interaction among modes. This observation motivates different PRISM structures for organizing rank-expanded transitions.

3.2 Structured Rank Expansion and Operator Structure

PaTH (Yang et al., 2025b) uses the rank-one positional transition

$$P_t^{\mathrm{PaTH}} = I - \beta_t w_t w_t^\top,$$

with $r_{\mathrm{pos}}(P_t^{\mathrm{PaTH}}) = 1$. PRISM retains the same positional interaction

$$s_{ij} = q_i^\top (P_i \cdots P_{j+1}) k_j,$$

but expands the internal structure of each transition P_t. We denote the configured expansion order by r. It specifies a rank budget, while the realized positional update rank $r_{\mathrm{pos}}(P_t)$ can be smaller depending on the learned operator. We consider three representative structures.

PRI-I: independent expansion. We partition the D-dimensional direction into r disjoint subspaces and collect the corresponding components as

$$W_t = [w_t^{(1)}, \ldots, w_t^{(r)}] \in \mathbb{R}^{D \times r}.$$

PRI-I applies the resulting modes independently:

$$P_t^{\mathrm{PRI-I}} = I - W_t D_t W_t^\top = I - \beta_t \sum_{a=1}^{r} w_t^{(a)} w_t^{(a)\top}.$$

Thus, increasing r introduces multiple independent deformation modes within the same transition.

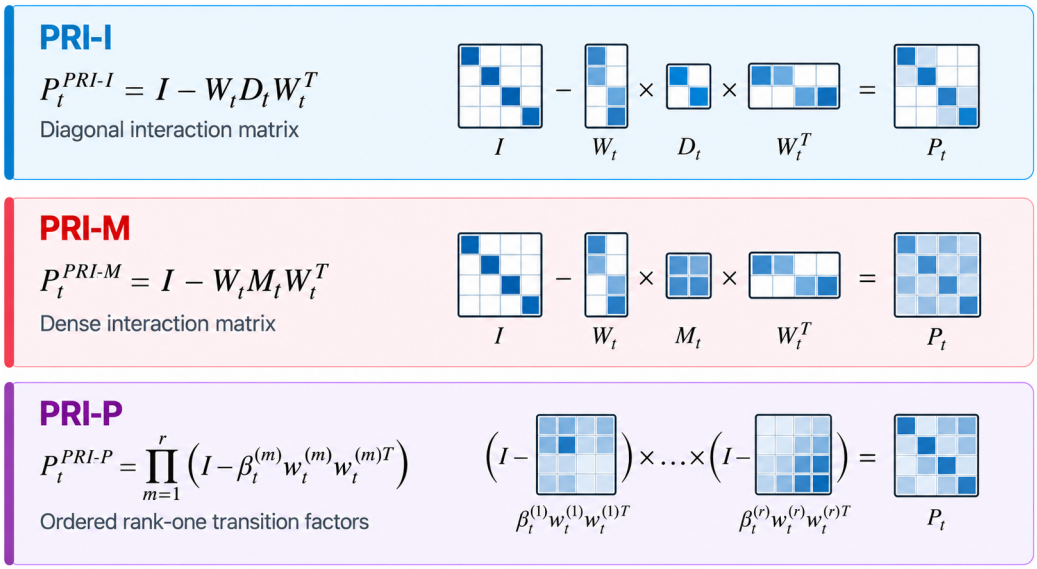


Figure 2: Structured positional rank expansion in PRISM.

PRI-M: mixed expansion. PRI-M uses the same partitioned basis but allows the modes to interact through a learned mixing matrix $M_t \in \mathbb{R}^{r\times r}$:

$$P_t^{\mathrm{PRI-M}} = I - W_t M_t W_t^{\top}.$$

The mixing matrix M_t is generated from the current token representation x_t through a lightweight learned mapping, making the interaction among transition modes input-dependent. The diagonal terms of M_t preserve independent mode updates, while the off-diagonal terms introduce interactions across subspaces. Setting $M_t = \beta_t I_r$ recovers PRI-I, making the I/M comparison a controlled study of mode interaction within the same basis.

PRI-P: product expansion. PRI-P instead composes r independently parameterized full-dimensional rank-one transitions:

$$P_t^{\mathrm{PRI-P}} = \prod_{m=1}^{r} \left(I - \beta_t^{(m)} w_t^{(m)} w_t^{(m)\top}\right).$$

Following PaTH, each $w_t^{(m)}$ is produced by a lightweight low-rank generator. Hence, additional product factors introduce only a marginal parameter overhead rather than full-dimensional projection matrices. Unlike PRI-I and PRI-M, which introduce multiple modes within a single transition, PRI-P expands transition capacity through ordered multiplicative composition.

Computational cost. All variants retain attention over the original L semantic tokens and therefore preserve the leading $O(L^2D)$ attention cost. With transition block size B, the additional transition-processing costs are

$$T_I = O(rLB^2 + LBD), \qquad T_M = O(LB^2r^3 + LBDr^2), \qquad T_P = O(rLB^2 + rLBD).$$

PRI-I shares the feature width across the r modes, PRI-M additionally incurs dense mode interaction, and PRI-P repeats full-dimensional rank-one operations. Thus, equal expansion order does not imply equal parameter count or runtime.

Overall, **PRISM** separates two aspects of positional rank expansion: **expansion order** controls how many transition modes are introduced, while **expansion structure** determines whether they remain independent, interact through mixing, or compose sequentially. Detailed rank and spectral analyses are deferred to Appendix A. Efficient training and decoding operators for the three expansion structures are provided in Appendix B.

4 EXPERIMENTS

We evaluate RoPE (Su et al., 2024), PaTH (Yang et al., 2025b), and PRISM on a synthetic task and a range of standard benchmarks.

4.1 SYNTHETIC TASK

Multi-query Repeated Associative Recall with N-back (MQRAR-N). We evaluate associative memory using the MQRAR-N task (Arora et al., 2024; Yang et al., 2025b). The task requires a model to retrieve the N-th most recent assignment associated with a key from a sequence of repeated key–value bindings. When $N = 1$, the model only needs to recover the latest value, while larger N requires maintaining an ordered history and skipping more recent assignments. For example, given 'A 1 B 2 A 5 A 7', the target for the third occurrence of 'A' with $N = 2$ is '1'. Positions without sufficient history are excluded from both loss and accuracy computation.

All models use two Transformer layers with hidden dimension 256 and two attention heads. Each sequence contains 512 tokens and 32 distinct keys, with repeated key–value assignments sampled throughout the sequence. Results are averaged over five random seeds from 42 to 46.

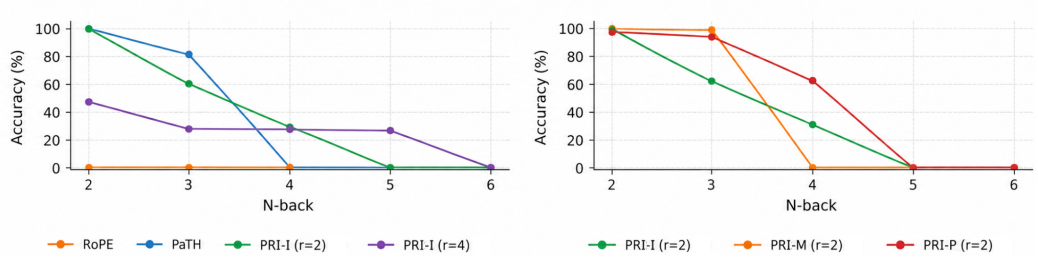


Figure 3: **MQRAR-N results.** Accuracy under increasing recall depth N, comparing expansion order and expansion structure.

As shown in Figure 3, increasing the expansion order changes the recall profile at larger N: $r = 2$ retains non-trivial accuracy at deeper recall depths where PaTH collapses, while further increasing the expansion order does not yield monotonic gains. Moreover, PRISM variants with the same expansion order exhibit distinct behaviors. These results support our hypothesis in a controlled retrieval setting: positional capacity depends on both the available deformation modes and how they are organized.

4.2 LANGUAGE MODELING EXPERIMENTAL SETUP

Setup. For the main positional experiments, we train 340M and 760M LLaMA-style decoder-only models (Touvron et al., 2023) on 10B and 50B FineWeb-Edu tokens (Penedo et al., 2024) with training lengths of 2K and 4K, respectively, comparing RoPE, PaTH, and PRISM under matched backbones and training recipes. Unless otherwise specified, PRISM uses expansion order $r = 2$. We additionally evaluate PRISM in a 600M GatedDeltaNet-H1-derived hybrid architecture (Yang et al., 2025a) trained on 25B FineWeb-Edu tokens (Penedo et al., 2024) with a 4K training length and a 2K sliding-attention window. Detailed model configurations and optimization hyperparameters are provided in Appendix C.

Benchmark. For language modeling and commonsense reasoning, we report perplexity on WikiText (Merity et al., 2016) and LAMBADA (Paperno et al., 2016) together with zero-shot performance on LAMBADA, PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy, and ARC-Challenge (Clark et al., 2018). For long-context evaluation, we use RULER (Hsieh et al., 2024), BABILONG (Kuratov et al., 2024), and LongBench-E (Bai et al., 2024) at multiple context lengths. We evaluate information extraction on SWDE (Lockard et al., 2019), SQuAD-completion (Rajpurkar et al., 2016), and FDA (Arora et al., 2023; Gao et al., 2021). Finally, we measure length extrapolation by evaluating perplexity on Code-

Parrot (Hugging Face, 2021), PG19 (Rae et al., 2019), and NarrativeQA (Kočiskỳ et al., 2018) up to 64K tokens. Detailed task protocols and metrics are provided in the Appendix D.

Experimental scope. All LM experiments use a single fixed random seed; evaluating variance across multiple seeds and larger model scales remains an important direction for future work.

4.3 Main Results

We evaluate PRISM along three axes: expansion structure, expansion order, and rank expansion versus head expansion. Tables 2 and 3 summarize the results on standard, information-intensive, and long-context benchmarks.

Table 2: Results on language modeling, commonsense reasoning, and information-intensive benchmarks.

Model	r	Wiki. $ppl \downarrow$	LMB. $ppl \downarrow$	LMB. $acc \uparrow$	PIQA $acc \uparrow$	Hella. $acc_n \uparrow$	Wino. $acc \uparrow$	ARC-e $acc \uparrow$	ARC-c $acc_n \uparrow$	Avg. $acc \uparrow$	SWDE $acc \uparrow$	SQuAD $acc \uparrow$	FDA $acc \uparrow$	Avg. $acc \uparrow$
340M params / 10B tokens														
RoPE	0	32.33	51.96	29.4	**65.8**	36.2	51.0	**56.0**	**27.5**	44.3	34.65	4.46	6.81	15.31
PaTH	1	31.26	43.38	31.5	65.0	36.4	50.9	54.4	26.0	44.0	44.10	8.58	13.52	22.07
PRI-I	2	30.99	39.89	32.1	64.8	36.8	**52.4**	53.5	26.0	44.3	45.27	**11.66**	12.43	23.12
PRI-I	4	30.91	**37.72**	**33.1**	65.6	**37.1**	51.7	54.3	27.1	**44.8**	**47.25**	8.61	**14.34**	**23.40**
PRI-M	2	**30.86**	37.93	**33.1**	65.1	**37.1**	51.0	55.1	25.1	44.4	36.81	6.03	11.34	18.06
PRI-P	2	30.91	42.02	31.5	64.7	36.6	49.9	54.9	**27.5**	44.2	39.10	3.60	12.00	18.20
760M params / 50B tokens														
RoPE	0	19.98	19.05	41.3	69.7	46.9	53.9	63.1	30.5	50.9	56.70	5.03	44.46	35.40
PaTH	1	19.38	16.89	42.0	69.0	47.6	55.2	63.5	31.1	51.4	60.31	21.88	73.14	51.78
PRI-I	2	**19.25**	16.98	42.1	68.9	48.3	**56.2**	65.4	**33.4**	**52.4**	**61.66**	**45.24**	**78.22**	**61.71**
PRI-I	4	19.31	**16.23**	**43.7**	69.2	48.5	54.9	65.8	32.4	**52.4**	59.14	14.34	63.16	45.55
PRI-M	2	19.27	16.68	42.8	**70.3**	**48.6**	54.0	65.5	31.7	52.2	50.14	38.57	48.00	45.57
PRI-P	2	20.45	21.52	40.1	70.0	**48.6**	55.1	**66.3**	33.3	52.2	58.15	13.32	61.33	44.27

Table 3: Results on long-context benchmarks.

Model	r	RULER 4K	RULER 8K	RULER 16K	BABILONG 0K	BABILONG 4K	BABILONG 8K	BABILONG 16K	LongBench-E 4K	LongBench-E 8K	LongBench-E 16K
760M params / 50B tokens											
RoPE	0	30.04	0	0	43.10	5.34	0.02	0	15.8	5.36	1.5
PaTH	1	39.85	29.5	12.28	41.30	19.8	**8.75**	2.72	15.5	13.8	12.5
PRI-I	2	40.29	34.78	**24.95**	**46.94**	24.70	7.81	2.60	**17.5**	**17.4**	**16.5**
PRI-I	4	42.64	30.99	19.63	**46.94**	21.66	7.45	2.98	14.8	13.4	13.1
PRI-M	2	**49.15**	**41.99**	19.82	37.81	21.04	7.65	**3.20**	13.8	13.3	12.2
PRI-P	2	45.98	36.74	18.53	41.97	**25.02**	8.63	2.76	**17.5**	15.9	14.7

Expansion structure. At expansion order $r = 2$, different expansion structures exhibit distinct strengths. PRI-M is competitive on language modeling and performs strongly on several standard reasoning tasks, whereas PRI-I is consistently stronger on information-intensive benchmarks and LongBench-E. This shows that expansion order alone is insufficient to characterize a positional transformation: even under the same rank budget, different organizations of the deformation modes lead to substantially different empirical behavior. Appendix A.6 provides complementary evidence that the expanded modes remain active after training and exhibit structure-dependent realized spectra.

Expansion order. Moving from the rank-one PaTH baseline ($r = 1$) to PRI-I with expansion order $r = 2$ yields clear gains, especially on information-intensive and long-context tasks. At 760M, the information-intensive average increases from 51.78 to 61.71, while RULER and LongBench-E also

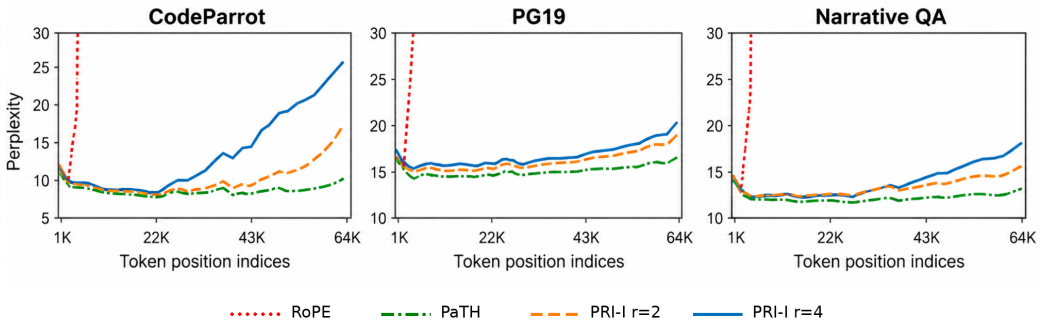


Figure 4: Perplexity under length extrapolation with different expansion orders.

improve substantially. Increasing the expansion order further to $r = 4$ provides no consistent additional benefit. Figure 4 further shows that larger expansion order can reduce length-extrapolation stability. Together, these results suggest a moderate-rank regime within the studied PRISM configurations: expanding the available deformation modes is beneficial, but the realized benefit depends on how the additional modes are structured.

4.4 Rank Expansion vs. Head Expansion

The simplicity of PRI-I raises an important question: if additional positional modes are introduced through parallel rank-one expansions, can the resulting gains be explained by simply increasing the number of heads? We therefore compare PRI-I with a PaTH variant that doubles the number of heads while maintaining a comparable parameter scale.

Tables 4 and 5 demonstrate that positional rank expansion provides benefits beyond increasing parallel head capacity. Head expansion improves several standard reasoning benchmarks, suggesting that additional model capacity can be useful. However, these improvements do not consistently transfer to information-intensive and long-context tasks. In contrast, PRI-I achieves substantially stronger performance on information retrieval benchmarks and provides more consistent gains on LongBench-E. These results indicate that the gains cannot be explained by head expansion alone and are associated with modifying the positional transition itself.

Table 4: Comparison between positional rank expansion and head expansion on language modeling, reasoning, and information-intensive benchmarks.

Model	r	Wiki. *ppl* ↓	LMB. *ppl* ↓	LMB. *acc* ↑	PIQA *acc* ↑	Hella. acc_n ↑	Wino. *acc* ↑	ARC-e *acc* ↑	ARC-c acc_n ↑	Avg. *acc* ↑	SWDE *acc* ↑	SQuAD *acc* ↑	FDA *acc* ↑	Avg. *acc* ↑
760M params / 25B tokens														
PaTH	1	22.18	25.44	36.9	66.1	42.8	**53.7**	59.8	**30.5**	48.3	51.0	18.4	56.8	42.1
PaTH-head	1	22.11	**21.99**	**39.1**	**67.8**	43.2	53.0	**60.4**	29.4	**48.8**	52.8	13.9	53.7	40.2
PRI-I	2	**22.05**	22.81	38.3	**67.8**	**43.3**	52.1	58.7	30.2	48.4	**54.0**	**37.6**	**58.0**	**49.9**

Table 5: Comparison between positional rank expansion and head expansion on long-context benchmarks.

Model	r	RULER 4K	RULER 8K	RULER 16K	BABILONG 0K	BABILONG 4K	BABILONG 8K	BABILONG 16K	LongBench-E 4K	LongBench-E 8K	LongBench-E 16K
760M params / 25B tokens											
PaTH	1	27.47	14.79	3.84	**43.18**	15.97	5.51	1.98	13.76	11.24	12.14
PaTH-head	1	**33.17**	20.77	**10.84**	41.65	17.27	5.62	**2.20**	14.37	11.94	12.03
PRI-I	2	32.82	**22.49**	6.71	42.15	**20.14**	**5.96**	1.64	**16.51**	**12.25**	**13.60**

Table 6: Results of hybrid models on language modeling, commonsense reasoning, and information-intensive benchmarks.

Model	Wiki. *ppl* ↓	LMB. *ppl* ↓	LMB. *acc* ↑	PIQA *acc* ↑	Hella. acc_n ↑	Wino. *acc* ↑	ARC-e *acc* ↑	ARC-c acc_n ↑	Avg. *acc* ↑	SWDE *acc* ↑	SQuAD *acc* ↑	FDA *acc* ↑	Avg. *acc* ↑
600M params / 25B tokens													
Transformer	24.20	29.57	36.0	67.1	41.0	50.6	59.1	28.1	47.0	50.3	8.6	37.0	32.0
GDN-SWA	23.16	25.61	36.6	67.5	43.1	50.0	**61.4**	29.7	48.0	49.9	41.1	21.8	37.6
GDN-SWPaTH	22.80	**22.81**	**39.2**	66.9	**43.3**	50.7	60.3	28.9	48.2	57.4	42.2	23.6	41.1
GDN-SWPRI-I	22.95	25.04	36.1	67.9	42.5	50.8	61.2	28.9	47.9	**59.5**	40.5	28.1	42.7
GDN-ATTN	22.75	26.01	36.1	67.8	42.7	**51.5**	60.1	29.4	47.9	54.1	41.7	42.0	45.9
GDN-PaTH	23.00	24.72	37.9	67.7	42.6	49.8	61.0	29.5	48.1	59.0	41.5	28.3	43.0
GDN-PRI-I	**22.49**	23.67	37.6	**69.0**	42.9	51.1	60.1	**30.5**	**48.5**	57.4	**43.7**	**56.1**	**52.4**

4.5 Hybrid Models

Table 6 shows that PRISM also improves hybrid architectures. Replacing PaTH with PRISM increases the information-intensive average from 43.0 to 52.4, while preserving competitive performance on standard benchmarks; similar gains are also observed in the sliding-window variants. These results suggest that the benefits of positional-rank expansion are not limited to attention-only models, but extend to hybrid recurrent–attention architectures.

4.6 Training and Inference Efficiency

We compare PaTH, PRI-I, PRI-M, and PRI-P under a controlled single-GPU setting, with all PRISM variants using expansion order $r = 2$. All models are evaluated in BF16 on a single NVIDIA A800 GPU. For training, we measure throughput under four sequence-length and batch-size configurations: 2K×8, 4K×4, 8K×2, and 16K×1, keeping the number of tokens per update fixed at 16K.

For autoregressive decoding, we evaluate prompt lengths of 4K, 8K, and 16K with batch size 1, generating 128 tokens for each prompt. Each configuration uses five warm-up iterations followed by five measured repetitions.

The efficiency of rank expansion depends strongly on its structure. PRI-I introduces relatively limited training overhead over PaTH, whereas more complex interaction patterns incur additional computational costs. For autoregressive decoding, exploiting the structure of PRISM transitions substantially improves throughput at long contexts.

These results show that expansion structure governs not only modeling behavior but also the efficiency–performance trade-off of positional rank expansion.

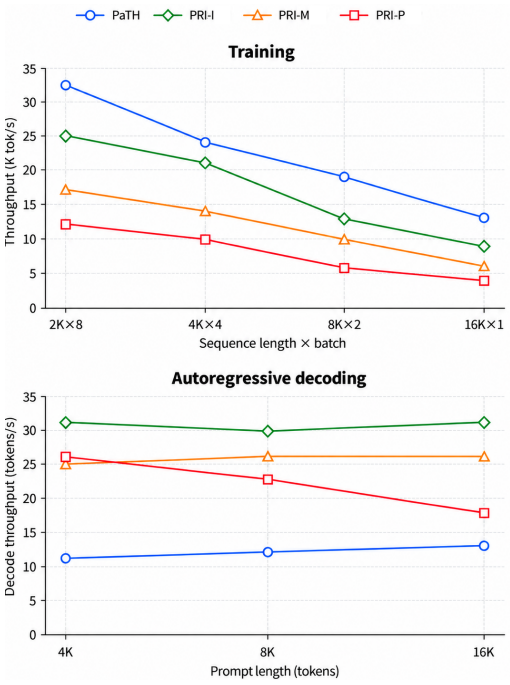


Figure 5: Training and decoding efficiency.

5 Conclusion

Our results suggest that non-isometric positional deformation should not be viewed solely as distortion: when properly controlled, it can provide useful positional capacity, with its benefit depending on how the deformation modes are organized.

REPRODUCIBILITY STATEMENT

We provide model architectures, training configurations, evaluation protocols, and additional rank and spectral analyses in the appendix. All controlled comparisons use matched backbones, training data, and optimization settings unless otherwise specified. An anonymized implementation of PRI-I, PRI-M, and PRI-P, together with the corresponding implementation details, is provided in the supplementary material.

AI USE STATEMENT

Generative AI tools were used to assist with conceptual framing and hypothesis refinement, provide feedback on experimental design and result interpretation, draft and revise portions of the manuscript, and improve writing and presentation. All AI-assisted suggestions and content were reviewed and verified by the authors. The authors take full responsibility for the methodology, experiments, claims, and final content of this work.

REFERENCES

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433*, 2023.

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, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Yuhan Chen, Ang Lv, Jian Luan, Bin Wang, and Wei Liu. Hope: A novel positional encoding without long-term decay for enhanced context awareness and extrapolation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23044–23056, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. *Zenodo*, 2021.

Zihan Gu, Ruoyu Chen, Han Zhang, Hua Zhang, and Yue Hu. Deconstructing positional information: From attention logits to training biases. In *International Conference on Learning Representations*, volume 2026, pp. 136369–136393, 2026.

Li Hengyu. Fingerprint, not blueprint: How positional schemes set the default spectral algebra of attention. *arXiv preprint arXiv:2607.06621*, 2026.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Hugging Face. Codeparrot: Clean python code dataset. `https://huggingface.co/datasets/codeparrot/codeparrot-clean`, 2021.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. Openceres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3047–3056, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shijie Mei, Man Yao, Jiabo Tong, Bo XU, and Guoqi Li. Head-in-head in linear attention. In *Forty-third International Conference on Machine Learning*, 2026.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Sajad Movahedi, Timur Carstensen, Arshia Afzal, Frank Hutter, Antonio Orvieto, and Volkan Cevher. Selective rotary position embedding. In *International Conference on Learning Representations*, volume 2026, pp. 10219–10247, 2026.

Yui Oka, Itsumi Saito, Kyosuke Nishida, and Kuniko Saito. Frequency bands in rope: Base frequency and context length shape the interpolation–extrapolation trade-off. In *The Fourteenth International Conference on Learning Representations*, 2026.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

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.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *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, 2018.

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

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Improving state-tracking in linear rnns via householder products. In *Advances in Neural Information Processing Systems*, volume 38, pp. 153738–153782, 2025.

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

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14590–14604, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arxiv 2023. *arXiv preprint arXiv:2302.13971*, 10, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Davis Wertheimer, Aozhong Zhang, Derrick Liu, Penghang Yin, and Naigang Wang. Frayed rope and long inputs: A geometric perspective. *arXiv preprint arXiv:2603.18017*, 2026.

Songlin Yang and Yu Zhang. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism, January 2024. URL `https://github.com/fla-org/flash-linear-attention`.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025a.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. Path attention: Position encoding via accumulating householder transformations. In *Proceedings of NeurIPS*, 2025b.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Jiefu Zhang, Zhen Qin, Jiabo Tong, Shijie Mei, JiaKui Hu, Yuqi Pan, Anjie Hu, Man Yao, Bo XU, and Guoqi Li. Householder-diagonalized linear attention (HDLA): Utilizing enhanced decay mechanism for efficient sequence modeling. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL `https://openreview.net/forum?id=HVFjzaQeig`.

Yaobo Zhang. Jordan-rope: Non-semisimple relative positional encoding via complex jordan blocks. *arXiv preprint arXiv:2605.04217*, 2026.

Yifan Zhang, Zixiang Chen, Yifeng Liu, Qin Zhen, Rina Hughes, Kangping Xu, Yang Yuan, Quanquan Gu, and Andrew Yao. Group representational position encoding. In *International Conference on Learning Representations*, volume 2026, pp. 56496–56519, 2026b.

Yu Zhang and Songlin Yang. Flame: Flash language modeling made easy, January 2025. URL `https://github.com/fla-org/flame`.

A Rank and Spectral Analysis of PRISM

This section provides the rank and spectral analyses omitted from the main text. For a positional transition $P \in \mathbb{R}^{D\times D}$, recall

$$E(P) = P^\top P - I, \qquad r_{\mathrm{pos}}(P) = \operatorname{rank} E(P).$$

Since the eigenvalues of $P^\top P - I$ are $\sigma_i^2(P) - 1$, $r_{\mathrm{pos}}(P)$ counts the singular modes whose singular values differ from one.

A.1 Update Rank of Representative Positional Operators

Recall that for a positional transition $P \in \mathbb{R}^{D\times D}$,

$$E(P) = P^\top P - I, \qquad r_{\mathrm{pos}}(P) = \operatorname{rank} E(P).$$

We derive below the update ranks reported in Table 1. Unless otherwise specified, the stated ranks refer to the generic non-degenerate case.

RoPE. RoPE applies independent planar rotations,

$$P = \operatorname{blockdiag}_{j=1}^{D/2} R_j(n), \qquad R_j(n) = \begin{pmatrix} \cos(n\theta_j) & -\sin(n\theta_j) \\ \sin(n\theta_j) & \cos(n\theta_j) \end{pmatrix}.$$

Each block is orthogonal,

$$R_j(n)^\top R_j(n) = I_2,$$

and therefore

$$P^\top P = I_D.$$

Hence

$$E(P) = 0, \qquad \boxed{r_{\mathrm{pos}}(P) = 0}.$$

XPos. Each XPos block can be written as a scaled planar rotation,

$$P_j = s_j(n) R_j(n), \qquad s_j(n) = e^{\pm \xi_j n}.$$

Since $R_j(n)$ is orthogonal,

$$P_j^\top P_j - I_2 = \left(s_j(n)^2 - 1\right) I_2.$$

Thus each block with nontrivial scaling, $s_j(n)^2 \neq 1$, contributes rank two. Therefore,

$$r_{\mathrm{pos}}(P) = 2 \,\#\left\{j : s_j(n)^2 \neq 1\right\}.$$

When all $D/2$ blocks have nontrivial scaling,

$$\boxed{r_{\mathrm{pos}}(P) = D}.$$

PaTH. PaTH uses the rank-one symmetric transition

$$P = I - \beta w w^\top, \qquad \|w\|_2 = 1.$$

Since P is symmetric,

$$\begin{aligned} E(P) &= (I - \beta w w^\top)^2 - I \\ &= -2\beta w w^\top + \beta^2 w w^\top w w^\top \\ &= (\beta^2 - 2\beta) w w^\top. \end{aligned}$$

Because $ww^\top$ has rank one, for $\beta \notin \{0, 2\}$,

$$\boxed{r_{\mathrm{pos}}(P) = 1}.$$

GRAPE-AP. GRAPE-AP uses the additive unipotent transition

$$P = I - \psi E,$$

where $E = e_p e_q^\top$ with $p \neq q$ is rank one and satisfies $E^2 = 0$. Its metric-deformation operator is

$$E(P) = -\psi(E + E^\top) + \psi^2 E^\top E.$$

The deformation is supported on $\operatorname{span}\{e_p, e_q\}$. Restricted to this two-dimensional subspace,

$$E(P) = \begin{pmatrix} 0 & -\psi \\ -\psi & \psi^2 \end{pmatrix}.$$

Its determinant is

$$-\psi^2,$$

which is nonzero whenever $\psi \neq 0$. Hence the metric deformation has rank two:

$$\boxed{r_{\mathrm{pos}}(P) = 2}.$$

Notably, although $\operatorname{rank}(P - I) = 1$, the induced metric deformation has rank two.

Jordan-RoPE. Consider one order-m complex Jordan block,

$$J_m = (-\gamma + i\omega)I_m + \eta N_m, \qquad N_m^m = 0,$$

with positional operator

$$P = \mathcal{R}[\exp(nJ_m)]\,,$$

where $\mathcal{R}[\cdot]$ denotes realification. Since the scalar and nilpotent terms commute,

$$\exp(nJ_m) = e^{-\gamma n}e^{i\omega n}A_m(n), \qquad A_m(n) = \exp(\eta n N_m).$$

The phase factor $e^{i\omega n}$ becomes an orthogonal rotation under realification and therefore does not contribute to metric deformation. Using the realification identity

$$\mathcal{R}(A)^\top \mathcal{R}(A) = \mathcal{R}(A^*A),$$

we obtain

$$E(P) = \mathcal{R}\left[e^{-2\gamma n}A_m(n)^*A_m(n) - I_m\right].$$

Since realification doubles the rank of a complex Hermitian matrix,

$$r_{\mathrm{pos}}(P) = 2\,\mathrm{rank}_{\mathbb{C}}\left(e^{-2\gamma n}A_m(n)^*A_m(n) - I_m\right).$$

Consequently,

$$\boxed{0 \le r_{\mathrm{pos}}(P) \le 2m}.$$

The lower endpoint is attained by the purely isometric case $\gamma = \eta = 0$, which reduces to a rotation, while a generic non-isometric order-m Jordan block has

$$r_{\mathrm{pos}}(P) = 2m.$$

For a full head containing multiple Jordan blocks, the update ranks add across blocks, and the total rank can therefore reach D.

A.2 PRI-I: Independent Block Deformation

PRI-I partitions the normalized direction w_t into r disjoint coordinate blocks,

$$W_t = [w_t^{(1)}, \ldots, w_t^{(r)}] \in \mathbb{R}^{D\times r},$$

and uses

$$P_t^{\mathrm{I}} = I - \beta_t W_t W_t^\top.$$

Because the blocks have disjoint support,

$$G_t = W_t^\top W_t = \mathrm{Diag}(g_{t,1}, \ldots, g_{t,r}), \qquad g_{t,a} = \|w_t^{(a)}\|_2^2.$$

Assuming the active blocks are nonzero, define

$$U_t = W_t G_t^{-1/2}, \qquad U_t^\top U_t = I.$$

Then

$$P_t^{\mathrm{I}} = I - U_t(\beta_t G_t)U_t^\top.$$

The singular values associated with the active block directions are

$$\sigma_a(P_t^{\mathrm{I}}) = |1 - \beta_t g_{t,a}|.$$

For $\beta_t \in (0, 2)$ and $g_{t,a} > 0$, these values differ from one, giving

$$r_{\mathrm{pos}}(P_t^{\mathrm{I}}) = \#\{a : g_{t,a} > 0\} \le r.$$

In particular, if all blocks are active,

$$r_{\mathrm{pos}}(P_t^{\mathrm{I}}) = r.$$

A.3 PRI-M: MIXED LOW-RANK DEFORMATION

PRI-M uses the same basis W_t as PRI-I, but allows the expanded modes to interact through a symmetric mixing matrix $M_t \in \mathbb{R}^{r\times r}$:

$$P_t^{\mathrm{M}} = I - W_t M_t W_t^\top.$$

Here, M_t jointly controls the strength and interaction of the deformation modes. In particular, setting $M_t = \beta_t I_r$ recovers PRI-I.

Let

$$G_t = W_t^\top W_t, \qquad U_t = W_t G_t^{-1/2},$$

and define the active spectral core

$$C_t = G_t^{1/2} M_t G_t^{1/2}.$$

Then

$$P_t^{\mathrm{M}} = I - U_t C_t U_t^\top.$$

Since M_t is symmetric in our construction, so is C_t, and the metric-deformation operator becomes

$$E(P_t^{\mathrm{M}}) = U_t(C_t^2 - 2C_t)U_t^\top.$$

Therefore,

$$r_{\mathrm{pos}}(P_t^{\mathrm{M}}) = \#\left\{a : \lambda_a(C_t) \notin \{0,2\}\right\}.$$

Under our parameterization,

$$M_t = \beta_t R_t R_t^\top,$$

with $\beta_t \in (0,2)$, so that $C_t \succeq 0$. Except for the degenerate case in which an active eigenvalue of C_t equals 2,

$$r_{\mathrm{pos}}(P_t^{\mathrm{M}}) = \mathrm{rank}(C_t) = \mathrm{rank}(M_t) \le r.$$

This highlights the distinction between expansion order and expansion structure. In PRI-I, the deformation modes remain tied to their individual block directions, whereas PRI-M can redistribute the same r-dimensional rank budget across linear combinations of these directions through M_t. If the learned modes remain independent, the full rank budget can be realized; if they become correlated, the deformation can concentrate into fewer spectral directions. Thus, equal expansion order does not necessarily imply equal realized rank or equal spectral organization.

A.4 PRI-P: COMPOSITIONAL RANK EXPANSION

PRI-P composes r full-dimensional rank-one transitions,

$$P_t^{\mathrm{P}} = \prod_{m=1}^{r} P_t^{(m)}, \qquad P_t^{(m)} = I - \beta_t^{(m)} w_t^{(m)} w_t^{(m)\top}.$$

Each factor satisfies

$$r_{\mathrm{pos}}(P_t^{(m)}) = 1$$

under the non-degenerate setting $\beta_t^{(m)} \in (0,2)$ and $\|w_t^{(m)}\|_2 = 1$.

For two operators A and B,

$$E(AB) = B^\top E(A) B + E(B).$$

Using rank subadditivity,

$$r_{\mathrm{pos}}(AB) \le r_{\mathrm{pos}}(A) + r_{\mathrm{pos}}(B).$$

Applying this relation recursively gives

$$r_{\mathrm{pos}}(P_t^{\mathrm{P}}) \le \sum_{m=1}^{r} r_{\mathrm{pos}}(P_t^{(m)}) \le r.$$

Thus, PRI-P increases the available positional update rank through composition rather than through an r-dimensional interaction core.

The product structure also introduces order-dependent interactions. For two factors,

$$\begin{aligned} P^{(2)}P^{(1)} &= \left(I - \beta_2 w_2 w_2^\top\right)\left(I - \beta_1 w_1 w_1^\top\right) \\ &= I - \beta_1 w_1 w_1^\top - \beta_2 w_2 w_2^\top + \beta_1\beta_2 (w_2^\top w_1) w_2 w_1^\top. \end{aligned}$$

The final cross term depends on both the alignment and the ordering of the two directions. In general,

$$P^{(2)}P^{(1)} \neq P^{(1)}P^{(2)},$$

so the resulting transition need not remain symmetric even though each individual factor is symmetric. PRI-P therefore changes not only the number of available deformation directions, but also their ordered directional organization.

A.5 Comparison of the Three Structures

The three PRISM variants share the same goal of increasing local positional transition capacity, but realize it through different operator structures:

$$\begin{aligned} \text{PRI-I}: &\quad I - \beta_t W_t W_t^\top, \\ \text{PRI-M}: &\quad I - W_t M_t W_t^\top, \\ \text{PRI-P}: &\quad \prod_{m=1}^{r} \left(I - \beta_t^{(m)} w_t^{(m)} w_t^{(m)\top}\right). \end{aligned}$$

PRI-I distributes deformation across independent partitioned directions; PRI-M couples those directions through a learned spectral core; and PRI-P composes full-dimensional rank-one directions sequentially. Accordingly, the configured expansion order r should be interpreted as a rank budget rather than a complete characterization of the resulting operator. Realized rank, singular-value structure, and directional interactions can differ across the three constructions even at the same r.

A.6 Realized Deformation Modes

The expansion order r specifies a rank budget, but the learned operator need not use all available modes equally. We therefore inspect the realized metric-deformation operator

$$E(P_t) = P_t^\top P_t - I$$

of the trained 340M models on a shared held-out sequence.

For each token-level transition, let $s_i(E(P_t))$ denote the singular values of $E(P_t)$ and define

$$p_i = \frac{s_i(E(P_t))}{\sum_j s_j(E(P_t))}.$$

We measure the realized spectral dimensionality using the entropy-based effective rank

$$r_{\text{eff}} = \exp\left(-\sum_i p_i \log p_i\right).$$

The statistic is averaged over tokens and attention heads at each layer.

Figure 6 provides qualitative evidence that the configured expansion modes remain active after training. PaTH stays at effective rank 1, while PRI-I with $r = 2$ remains close to 2 across layers, with an average effective rank of 1.92. Increasing the expansion order to $r = 4$ raises the average effective rank to 3.33, indicating that the additional modes are used rather than collapsing to a lower-rank transition. PRI-P with $r = 2$ similarly retains two active modes, with an average effective rank of 1.83.

Importantly, equal expansion order does not imply equal realized spectral organization. Although PRI-M also has $r = 2$, its average effective rank is 1.51, indicating a more concentrated singular spectrum. This differs from both PRI-I and PRI-P despite the same configured rank budget.

Figure 7 provides a token-wise view at a representative middle layer. PaTH exhibits a single active deformation mode, whereas the rank-expanded variants retain multiple active modes across tokens. Their mode-wise spectral profiles differ across structures, consistent with the distinction between expansion order and expansion structure developed in Section 3.2.

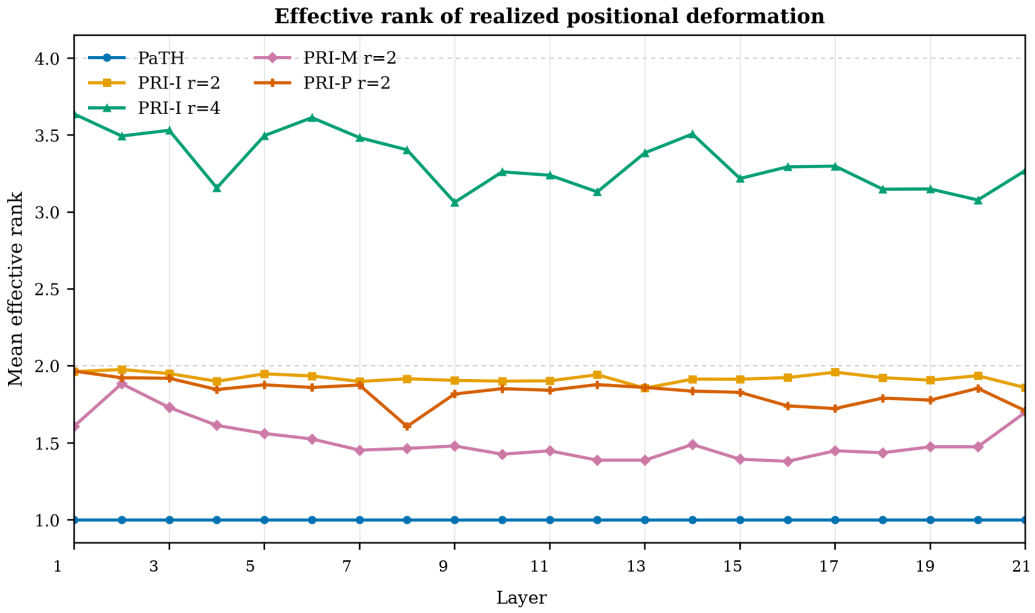


Figure 6: **Effective rank of realized positional deformation.** Entropy-based effective rank of the singular spectrum of $E(P_t) = P_t^\top P_t - I$ across layers of the 340M models, averaged over tokens and attention heads on a shared held-out sequence. Rank-expanded operators retain multiple active deformation modes, while different structures with the same expansion order exhibit distinct realized spectral organizations.

B Efficient Algorithms for PRI

We describe the efficient training and decoding implementations of the three PRI structures. All variants reuse the compact UT formulation of PaTH and avoid explicitly materializing products of $D \times D$ transition matrices. Their computational differences arise only from how the expanded transition modes are organized: PRI-I decomposes the transition into independent compact systems, PRI-M lifts the scalar UT system to a block-coupled one, and PRI-P extends the transition axis from N to rN ordered rank-one factors.

B.1 PaTH Compact UT Form

We first review the compact representation used by PaTH. For $j \le i$, the attention score is

$$s_{ij} = \frac{1}{\sqrt{D}} q_i^\top \left(P_i P_{i-1} \cdots P_{j+1}\right) k_j,$$

where each token-dependent transition takes the rank-one form

$$P_t = I_D - \beta_t w_t w_t^\top.$$

Consider a chunk containing N token transitions. We stack the transition directions as

$$W = \begin{bmatrix} w_1^\top \\ \vdots \\ w_N^\top \end{bmatrix} \in \mathbb{R}^{N \times D}, \qquad D_\beta = \mathrm{diag}(\beta_1, \ldots, \beta_N).$$

The sequential product of all transitions in the chunk admits the compact UT representation

$$G = P_N \cdots P_1 = I_D - W^\top \Gamma W,$$

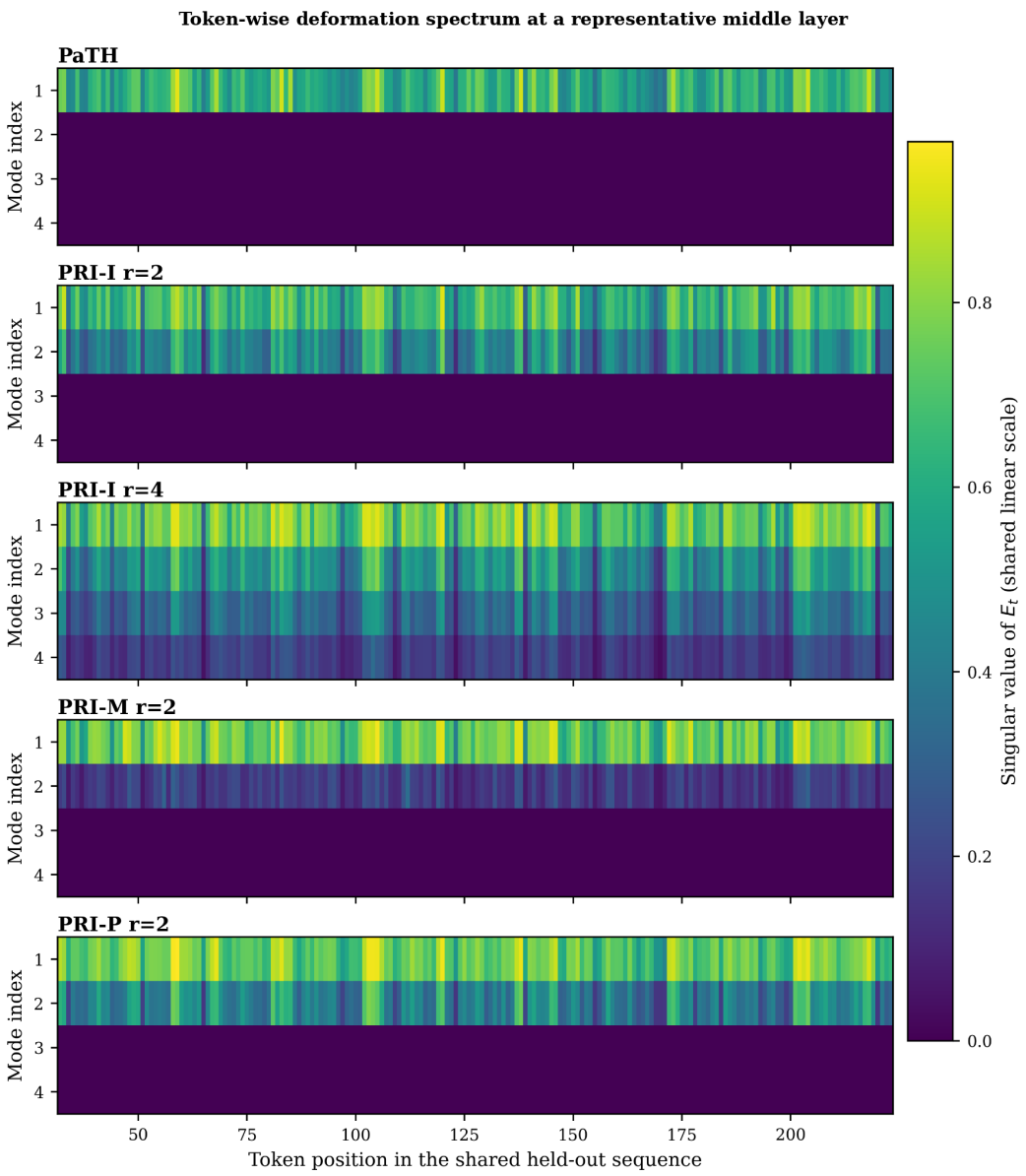


Figure 7: **Token-wise deformation spectrum at a representative middle layer.** Singular values of $E(P_t) = P_t^\top P_t - I$ across a shared held-out sequence, averaged over attention heads and visualized with a shared linear scale. PaTH contains one active deformation mode, whereas rank-expanded variants retain multiple modes with structure-dependent spectral profiles.

where

$$C = D_\beta W W^\top, \qquad \Gamma = (I + \mathrm{sLow}(C))^{-1} D_\beta.$$

Here $\mathrm{sLow}(\cdot)$ denotes the strictly lower-triangular part.

Thus, the sequential product of N dense $D \times D$ transition matrices is represented only through (W, Γ). The resulting compact coefficient is used to form boundary-adjusted queries and keys for intra-chunk causal attention, while cross-chunk contributions are accumulated by propagating the query-side state through $G^\top$.

B.2 PRI-I: INDEPENDENT TRANSITION MODES

PRI-I divides each head of dimension D into r independent transition modes, each of dimension $d = D/r$. Its token-level transition is block diagonal:

$$P_t^{\mathrm{I}} = \mathrm{diag}\left(P_t^{(1)}, \ldots, P_t^{(r)}\right),$$

with

$$P_t^{(a)} = I_d - \beta_t w_{t,a} w_{t,a}^\top, \qquad a = 1, \ldots, r.$$

The gate β_t is shared across modes, while each mode has an independent transition direction $w_{t,a}$.

For mode a, stack its transition directions inside a chunk as

$$W^{(a)} = \begin{bmatrix} w_{1,a}^\top \\ \vdots \\ w_{N,a}^\top \end{bmatrix} \in \mathbb{R}^{N \times d}.$$

The corresponding compact coefficient is

$$C^{(a)} = D_\beta W^{(a)} W^{(a)\top}, \qquad \Gamma^{(a)} = \left[I + \mathrm{sLow}\left(C^{(a)}\right)\right]^{-1} D_\beta,$$

and the accumulated transition is

$$G^{(a)} = I_d - W^{(a)\top} \Gamma^{(a)} W^{(a)}.$$

Therefore, PRI-I reduces to r independent PaTH compact-UT systems. The local and cross-chunk attention contributions can be evaluated independently for each mode and summed at the logit level:

$$A_{ij}^{\mathrm{I}} = \sum_{a=1}^{r} A_{ij}^{(a)}.$$

From an implementation perspective, the head tensor is reshaped as

$$[B, H, T, D] \to [B, H, T, r, d],$$

after which each mode follows the original PaTH computation. PRI-I therefore increases the number of independent transition modes without introducing cross-mode dependencies in the triangular solve.

B.3 PRI-M: MIXED TRANSITION MODES

PRI-M allows the expanded transition modes to interact through the mixing matrix $M_t \in \mathbb{R}^{r \times r}$. Following the notation in the main text, M_t already includes the scalar deformation gate β_t. Let

$$W_t = [\bar{w}_{t,1}, \ldots, \bar{w}_{t,r}] \in \mathbb{R}^{D \times r}$$

collect the mode-wise transition directions in the full head space. The token-level transition is

$$P_t^{\mathrm{M}} = I_D - W_t M_t W_t^\top.$$

Thus, relative to PaTH, the single transition direction w_t is replaced by the basis W_t, while the scalar coefficient β_t is generalized to the small matrix M_t.

Within a chunk, the token-level bases are stacked as

$$W = \begin{bmatrix} W_1^\top \\ W_2^\top \\ \vdots \\ W_N^\top \end{bmatrix} \in \mathbb{R}^{Nr \times D},$$

and their mixing coefficients are collected in the block-diagonal matrix

$$\mathcal{M} = \mathrm{blockdiag}\left(M_1, \ldots, M_N\right) \in \mathbb{R}^{Nr \times Nr}.$$

The PaTH compact system is then lifted from a scalar triangular system to a block-triangular one:

$$C^{\mathrm{M}} = \mathcal{M}WW^{\top}, \qquad \Gamma^{\mathrm{M}} = \left[I + \mathrm{bsLow}\left(C^{\mathrm{M}}\right)\right]^{-1}\mathcal{M}.$$

Here $\mathrm{bsLow}(\cdot)$ denotes the block strictly lower-triangular part with respect to token order. In particular, if C^{M} is viewed as an $N \times N$ matrix of $r \times r$ blocks,

$$C^{\mathrm{M}} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1N} \\ C_{21} & C_{22} & \cdots & C_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{N2} & \cdots & C_{NN} \end{bmatrix},$$

then

$$\mathrm{bsLow}\left(C^{\mathrm{M}}\right) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ C_{N1} & \cdots & C_{N,N-1} & 0 \end{bmatrix}.$$

The entire diagonal $r \times r$ blocks are removed; no additional elementwise triangular masking is applied inside each block.

The accumulated chunk transition is

$$G^{\mathrm{M}} = I_D - W^{\top}\Gamma^{\mathrm{M}}W.$$

Once Γ^{M} and G^{M} are obtained, the remaining boundary correction, intra-chunk causal attention, and cross-chunk scan follow the same blockwise template as PaTH.

Hence, PRI-M changes the compact coefficient structure rather than the attention traversal itself:

$$1 \times 1 \quad \longrightarrow \quad r \times r$$

at each token.

B.4 PRI-P: Expanded Transition Sequence

PRI-P organizes the r transition modes through ordered composition. At token t,

$$P_t^{\mathrm{P}} = P_{t,r} \cdots P_{t,2}P_{t,1},$$

where each factor is a rank-one transition,

$$P_{t,a} = I_D - \beta_{t,a}w_{t,a}w_{t,a}^{\top}, \qquad a = 1, \ldots, r.$$

Consequently, a sequence containing N token-level transitions can be expanded into an ordered sequence of rN elementary transitions:

$$(P_{1,1}, \ldots, P_{1,r}, P_{2,1}, \ldots, P_{2,r}, \ldots, P_{N,1}, \ldots, P_{N,r}).$$

The transition product is therefore

$$P_N^{\mathrm{P}} \cdots P_1^{\mathrm{P}} = P_{N,r} \cdots P_{N,1} \cdots P_{2,r} \cdots P_{2,1}P_{1,r} \cdots P_{1,1}.$$

This allows PRI-P to directly reuse the original PaTH compact-UT operator. We stack the rN transition directions in their product order:

$$\widetilde{W} = \begin{bmatrix} w_{1,1}^{\top} \\ \vdots \\ w_{1,r}^{\top} \\ w_{2,1}^{\top} \\ \vdots \\ w_{N,r}^{\top} \end{bmatrix} \in \mathbb{R}^{rN \times D},$$

and collect the corresponding gates as

$$\widetilde{D}_\beta = \mathrm{diag}\left(\beta_{1,1}, \ldots, \beta_{1,r}, \ldots, \beta_{N,1}, \ldots, \beta_{N,r}\right).$$

The compact coefficient is then computed exactly as in PaTH:

$$\widetilde{C} = \widetilde{D}_\beta \widetilde{W}\widetilde{W}^\top, \qquad \widetilde{\Gamma} = \left[I + \mathrm{sLow}\left(\widetilde{C}\right)\right]^{-1} \widetilde{D}_\beta,$$

giving

$$G^{\mathrm{P}} = I_D - \widetilde{W}^\top \widetilde{\Gamma} \widetilde{W}.$$

Therefore, PRI-P requires no new compact-UT algebra. It preserves the original PaTH operator and changes only the effective transition length:

$$\boxed{N \longrightarrow rN.}$$

The additional interaction among transition modes arises from their ordered multiplication rather than from an explicit mixing matrix.

B.5 Unified Implementation View

The three structures modify different parts of the same PaTH compact-UT template. PRI-I expands the number of independent systems, PRI-M expands the coefficient associated with each token, and PRI-P expands the transition sequence itself:

PRI-I:	$D \to r \times d$,	r independent UT systems,
PRI-M:	$1 \times 1 \to r \times r$,	block-coupled UT system,
PRI-P:	$N \to rN$,	expanded transition sequence.

Despite these differences, all variants follow the same overall blockwise pipeline:

transition parameters $\to \Gamma \to$ boundary-adjusted $Q, K \to$ intra-chunk attention $\to G \to$ cross-chunk scan.

This separation allows the transition structure to change while keeping the overall PaTH attention traversal unchanged.

B.6 Autoregressive Decoding

The same transition formulation supports efficient autoregressive decoding. During prefill, we use the chunked compact-UT computation described above. During decoding, each newly generated token t produces its transition P_t^χ, where

$$\chi \in \{\mathrm{I}, \mathrm{M}, \mathrm{P}\}.$$

Rather than recomputing the full transition product for every new query, the current transition is applied directly to the transformed historical key cache:

$$\bar{k}_j^{(t)} = P_t^\chi \bar{k}_j^{(t-1)}, \qquad j < t.$$

The current query then attends to the updated cache using ordinary dot products:

$$s_{tj} = \frac{1}{\sqrt{D}} q_t^\top \bar{k}_j^{(t)}.$$

Finally, the current key is appended as the newest cache entry,

$$\bar{k}_t^{(t)} = k_t.$$

For PRI-I, the cache update is performed independently across the r modes. For PRI-M, the update uses the mixed transition P_t^{M}. For PRI-P, the r rank-one factors are applied in their prescribed product order:

$$\bar{k}_j^{(t)} = P_{t,r} \cdots P_{t,1} \bar{k}_j^{(t-1)}.$$

Thus, prefill and decoding implement exactly the same transition semantics: prefill compresses multiple transitions into chunk-level compact representations, whereas decoding applies each newly generated transition directly to the persistent key cache.

C Model and Training Details

All models are trained from scratch on FineWeb-Edu (Penedo et al., 2024). Within each controlled comparison, models share the same backbone, training data, optimization budget, and training recipe unless explicitly stated otherwise. Our experiments are implemented with the Flame training framework and the Flash Linear Attention (FLA) library (Zhang & Yang, 2025; Yang & Zhang, 2024). The attention operators follow the blockwise implementation paradigm of FlashAttention-style kernels (Dao et al., 2022; Dao, 2024).

C.1 340M Models

The 340M models use a LLaMA-style decoder-only architecture (Touvron et al., 2023) with 21 layers, hidden size 1024, 16 attention heads, head dimension 64, and a vocabulary size of 32,000. The SwiGLU (Shazeer, 2020) intermediate size is 2880, and the maximum training sequence length is 2048. The PRI-P variant uses an intermediate size of 2840 for parameter matching.

We train RoPE, PaTH, PRI-I with expansion order $r \in \{2, 4\}$, PRI-M with $r = 2$, and PRI-P with $r = 2$. All models are trained on approximately 10B tokens.

Training uses AdamW (Loshchilov & Hutter, 2017) with a peak learning rate of 4×10^{-4}, weight decay 0.01, and 1000 warm-up steps, followed by cosine learning rate decay. The global batch size is 256 sequences. Gradient norm is clipped at 1.0, and training uses BF16 model parameters with FP32 reduction.

C.2 760M Models

The 760M models use 24 decoder layers, hidden size 1536, 24 attention heads, head dimension 64, a SwiGLU intermediate size of 6144, and a vocabulary size of 32,000. The maximum training sequence length is 4096.

We train RoPE, PaTH, and PRISM variants from scratch under a matched 50B-token training budget. Unless otherwise specified, PRISM uses expansion order $r = 2$; the expansion-order comparison additionally evaluates $r = 4$.

The global batch size is 1024 sequences. We use AdamW with a peak learning rate of 10^{-3} and weight decay 0.01. The learning rate is warmed up for 250 steps and then decayed with a cosine schedule.

C.3 Head-Expansion Control

To distinguish positional-rank expansion from increased parallel head capacity, we additionally train matched 760M models for 25B tokens. We compare PaTH, PRI-I with expansion order $r = 2$, and a PaTH variant with twice the number of positional heads. All models use the same backbone, sequence length, training data, and optimization recipe; only the positional transition or head multiplicity is changed.

C.4 600M Hybrid Models

The hybrid experiments use a 600M-scale backbone derived from GatedDeltaNet-H1 (Yang et al., 2025a). The model contains 14 layers, alternating seven Gated DeltaNet layers and seven sliding-window attention layers. The Gated DeltaNet layers use 9 query/key heads with head dimension 128 and 9 value heads with value dimension 256. The sliding-window attention layers use 12 heads with head dimension 128 and a window size of 2048. The model uses SwiGLU feed-forward layers with intermediate size 6144, RMSNorm (Zhang & Sennrich, 2019), a vocabulary size of 32,000, and a training sequence length of 4096.

We compare the original hybrid architecture with controlled variants incorporating attention, PaTH, or PRISM while keeping the surrounding backbone fixed. All hybrid models are trained on approximately 25B FineWeb-Edu tokens with a global batch size of 1024 sequences.

Training uses AdamW with a peak learning rate of 10^{-3} and weight decay 0.01. The learning rate is warmed up for 250 steps and then decayed with a cosine schedule. Gradient norm is clipped at 1.0.

D Evaluation Details

We use a common evaluation suite across the expansion-structure, expansion-order, and head-expansion experiments. The hybrid experiments reuse the language-modeling, commonsense-reasoning, and information- extraction subsets of this suite.

D.1 Language Modeling and Commonsense Reasoning

We evaluate language-modeling quality using perplexity on WikiText (Merity et al., 2016) and LAMBADA (Paperno et al., 2016). Lower perplexity indicates better language modeling.

We additionally evaluate zero-shot downstream performance on LAMBADA, PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy, and ARC-Challenge (Clark et al., 2018). We report standard accuracy for LAMBADA, PIQA, WinoGrande, and ARC-Easy, and normalized accuracy for HellaSwag and ARC-Challenge. The average downstream score reported in the main tables is computed over these six accuracy-based benchmarks and excludes perplexity.

Together, these tasks cover next-token prediction, physical commonsense, commonsense completion, coreference-style reasoning, and elementary science question answering.

D.2 Long-Context Evaluation

We evaluate long-context performance using RULER (Hsieh et al., 2024), BABILONG (Kuratov et al., 2024), and LongBench-E (Bai et al., 2024).

RULER. We report the zero-shot average score over the RULER subtasks at 4K, 8K, and 16K context lengths. RULER contains controlled synthetic tasks covering needle-in-a-haystack retrieval, multi-hop retrieval, aggregation, and variable tracking. It therefore provides a direct measure of how effectively a model preserves and accesses information as context length increases.

BABILONG. We follow the standard BABILONG evaluation protocol and report the average few-shot accuracy over QA1–QA5. These tasks require reasoning over facts embedded in increasingly long narrative contexts. We evaluate the original setting without added background text (denoted 0K) and settings with approximately 4K, 8K, and 16K contexts.

LongBench-E. We use the English subset of LongBench and group examples according to input length. We report the average score over three length ranges: 0–4K, 4–8K, and 8–16K. The evaluated tasks cover question answering, summarization, few-shot learning, and code understanding. Task-level results are provided separately when space permits.

D.3 Information Extraction

We evaluate information preservation and retrieval using SWDE (Lockard et al., 2019), SQuAD-completion (Rajpurkar et al., 2016), and FDA (Arora et al., 2023; Gao et al., 2021).

SWDE evaluates structured information extraction from semi-structured web pages. SQuAD-completion evaluates retrieval and completion of context-dependent answers, while FDA measures extraction of structured key–value information from document-like inputs. We report accuracy for all three datasets and use their arithmetic mean as the information- extraction average.

These tasks are particularly relevant to our study because they require the model not only to encode contextual information, but also to preserve and recover task-relevant content after intervening tokens.

D.4 LENGTH EXTRAPOLATION

We evaluate length extrapolation using perplexity on CodeParrot (Hugging Face, 2021), PG19 (Rae et al., 2019), and NarrativeQA (Kočiskỳ et al., 2018). Following the PaTH-style extrapolation protocol (Yang et al., 2025b), models trained with substantially shorter context windows are evaluated at token positions extending up to 64K.

CodeParrot evaluates long-range language modeling on source code, PG19 evaluates long-form book text, and NarrativeQA evaluates narrative text. We report perplexity as a function of token position rather than only a single aggregate value.

This evaluation is especially useful for studying positional rank expansion. Repeated non-isometric positional transitions can accumulate over long distances, and the resulting perplexity curves reveal whether increasing expansion order improves transition capacity at the cost of weaker long-range information retention.

D.5 HYBRID EVALUATION

The hybrid models are evaluated using the same language-modeling, commonsense-reasoning, and information-extraction protocols described above. We do not mix these results with the 340M or 760M tables, since the hybrid experiments use a different model scale and training budget. Instead, comparisons are made only among models trained under the same 600M/25B setting.