Deconstructing Attention: Investigating Design Principles for Effective Language Modeling

Huiyin Xue, Nafise Sadat Moosavi and Nikolaos Aletras
School of Computer Science, University of Sheffield, United Kingdom
{hxue12, n.s.moosavi, n.aletras}@sheffield.ac.uk

Abstract

The success of Transformer language models is widely credited to their dot-product attention mechanism, which interweaves a set of key design principles: mixing information across positions (enabling multi-token interactions), sequence-dependent activations (where attention weights adapt to each input), a specific mathematical form (dot-product similarities plus softmax weighting), and coupling of queries and keys to evolving hidden states (grounding attention in the current layer). However, the necessity of each of these principles remains largely untested. In this work, we systematically deconstruct attention by designing controlled variants that selectively relax these principles, applied both uniformly across all layers and in hybrid architectures where only some layers retain standard attention. Our empirical analysis reveals that mechanisms for mixing tokens are indispensable, as their absence collapses models to near-random behavior, while the exact mathematical form and sequence dependency can be substantially relaxed, especially when preserved in just a subset of layers. Surprisingly, even variants that fail in isolation can achieve robust performance when interleaved with standard attention, highlighting a cooperative effect. These findings deepen our understanding of what truly underpins attention's effectiveness and open new avenues for simplifying language models without sacrificing performance.[1]

1 Introduction

The remarkable success of Transformer-based language models (Singh, 2025; Liu et al., 2024; Yang et al., 2024a, LMs) is widely attributed to the dot-product attention mechanism (i.e. standard attention), which enables these models to weight the significance of each token in a sequence by computing pairwise similarities of their contextual representations (Vaswani et al., 2017). However, this powerful mechanism comes at a substantial computational cost with respect to the input sequence length (L). This has led to a diverse landscape of proposed mechanisms, including processing longer context (Tay et al., 2022), token-mixing via pooling and multi-layer perceptron MLP-Mixer (Tolstikhin et al., 2021), non-parametric transformations (Yu et al., 2022; Lee-Thorp et al., 2022), optimized kernel functions (Aksenov et al., 2024; Arora et al., 2024; Qin et al., 2022; Peng et al., 2021; Kasai et al., 2021; Choromanski et al., 2021; Katharopoulos et al., 2020), and linear recurrent neural network (RNN) architectures (Siems et al., 2025; Peng et al., 2025; Dao and Gu, 2024; Yang et al., 2024b; Qin et al., 2024; Peng et al., 2024; Poli et al., 2023; Peng et al., 2023; Orvieto et al., 2023).

Despite this rich body of work, most of these approaches implicitly preserve several underlying design principles inherited from standard attention. Broadly, these principles include: (1) incorporating mechanisms for mixing information across tokens (`Token Mixing`), enabling multi-token interactions, (2) emulating the original mathematical form of standard attention (`Mathematical Form`), i.e. dot-product similarities followed by softmax weighting, (3) enforcing strict sequence-dependency in activation maps (`Sequence-Dependency`), where attention weights depend on the specific input sequence, and (4) deriving queries and keys from the current layer's hidden states (`Current QK`), as opposed to other input types such as uncontextualized representations. However, the importance of each of these principles remains largely untested. *Are all of these truly essential, or could relaxing some of them suffice if applied selectively?*

Motivated by this foundational question and guided by Occam's Razor (Baker, 2022), we take a diagnostic approach: we systematically relax these principles through controlled attention variants, evaluated in two settings: (1) uniform replace-

[1] Code is available at https://github.com/HUIYINXUE/DeconAttn.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 708–727
December 20-24, 2025 ©2025 Association for Computational Linguistics

ment across all layers, and (2) hybrid configurations that interleave standard and simplified modules. Through extensive empirical analysis across multiple model sizes, attention variants, and layer configurations, while carefully matching parameter counts of variants, we uncover a set of insights that refine our understanding of key attention principles.

Under *uniform* replacement, mechanisms enabling token mixing prove indispensable: removing them, e.g. in *MLP* variants, leads to near-random accuracy on challenging natural language understanding (NLU) tasks, though such models still capture superficial statistical patterns, as reflected in improved perplexity over trivial baselines. Retaining the dot-product structure and sequence-dependent weighting contributes to stability, but these elements are not strictly necessary in every layer, provided token interactions remain strong.

Notably, in hybrid configurations that interleave simpler attention mechanisms with standard layers, we uncover a striking pattern: attention variants that fail in isolation can nonetheless contribute meaningfully when paired with standard attention, achieving robust performance that often matches or exceeds fully standard models. This suggests standard layers may stabilize activations, mitigate distributional drift, and foster cooperative dynamics across the network, as reflected in both predictive outcomes and structural diagnostics such as attention entropy, head diversity, and sink behaviors.

While hybrid attention schemes have been explored in prior work, such as taking advantages of state space models (Glorioso et al., 2024) or augmenting feed-forward modules via mixture-of-experts routing (Lenz et al., 2025), these are typically driven by performance or efficiency goals. *By contrast, our hybrid designs serve as deliberate probes to isolate and examine the causal roles of specific attention properties.* Taken together, our findings challenge the assumption that attention mechanisms must adhere rigidly to their original formulation. By identifying which components are essential and which can be simplified, we outline a path toward new LM architectures that can be structurally leaner and adaptable.

2 Related Work

Prior research attributes the success of Transformer models to their efficient token mixing mechanisms. Consequently, numerous studies explore replacing the standard dot-product attention with simpler architectural components that enable parallel training. For instance, Yu et al. (2022) demonstrate the effectiveness of pooling, MLPs, and convolution as alternatives within vision Transformers. Similarly, Lee-Thorp et al. (2022) highlight the efficiency of token mixers based on Fourier transformation and random projection in the BERT model (Devlin et al., 2019). However, these investigations focus on encoder-only Transformer architectures and may not readily adapt to causal language modeling. While Tolstikhin et al. (2021) to introduce a learnable linear layer for token mixing by employing position-wise projection vectors, similar to Linformer (Wang et al., 2020), this approach encounters scalability challenges with long sequences due to its parameter count growing linearly with L. Concurrent research largely retains the standard dot-product attention mechanism as a foundational principle. Efforts to reduce the computational cost of this mechanism primarily follow two strategies: weight sharing (Rajabzadeh et al., 2024; Ainslie et al., 2023; Xue and Aletras, 2023; Yan et al., 2021; Kitaev et al., 2020; Shazeer, 2019; Xiao et al., 2019) or input length shrinkage (Nawrot et al., 2023; Clark et al., 2022; Xue and Aletras, 2022).

Recent work revisits linear RNNs to handle inputs of varying length (Gu and Dao, 2024; Poli et al., 2023; Peng et al., 2023; Orvieto et al., 2023; Gu et al., 2022). Follow-up research further improves performance by designing more sophisticated gating mechanisms and update rules (He et al., 2025; Lin et al., 2025; Siems et al., 2025; Peng et al., 2025; Dao and Gu, 2024; Yang et al., 2024b; Qin et al., 2024; Peng et al., 2024), with the goal of mimicking human memory, drawing inspiration from the work of Schlag et al. (2021) on fast weight programmers. Notably, such replacements can also be selectively applied to a subset of attention layers or heads (Lenz et al., 2025; Ren et al., 2025; Team et al., 2024; Glorioso et al., 2024; Peng and Cao, 2024; Dong et al., 2025; Tay et al., 2019). Additionally, this work operates on the contextual representations encoded by deep networks to generate activation maps dynamically.

Another line of research approximates the dot-product computation to achieve linear complexity. These methods rely on various kernel functions that emulate the exponential function using its Taylor expansion (Aksenov et al., 2024; Arora et al., 2024; Qin et al., 2022; Peng et al., 2021; Kasai et al., 2021; Choromanski et al., 2021; Katharopoulos et al., 2020). This allows for prioritization of the key-value dot product through feature mapping. However, this line of work does not examine the necessity of the other key principles of attention mechanism identified in §1.

3 Attention Variants

To operationalize our investigation of the four key design principles identified in §1, we design targeted variations of attention that selectively relax each property. This allows us to probe their necessity in a controlled, principled framework.

3.1 Standard Dot-product Attention

We take standard scaled dot-product attention (Vaswani et al., 2017) as our baseline, where queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$) are computed from the layer hidden states $\mathbf{H} \in \mathbb{R}^{L \times d_m}$:

$$\mathbf{O} = \mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{AV} \tag{1}$$

$$\mathbf{A} = \mathrm{Softmax}\left(\mathbf{QK}^\top / \sqrt{d_h}\right) \tag{2}$$

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{HW}^{Q,K,V} \tag{3}$$

This follows all principles: mixing information across positions via $\mathbf{A}$, using a similarity-softmax form, adapting to each input sequence, and tying $\mathbf{Q}$, $\mathbf{K}$ to the current hidden state $\mathbf{H}$.

3.2 Relaxing Token Mixing

MLP. To directly examine the necessity of cross-token interactions, we replace attention with a gated MLP layer, consisting of three fully-connected (FC) layers ($\mathrm{FC}_{\mathrm{Dn}}, \mathrm{FC}_{\mathrm{Gt}}, \mathrm{FC}_{\mathrm{Up}}$) for down-projection, gating and down-projection respectively. This effectively eliminates token mixing and each token is processed independently, only attending to itself.

$$\begin{aligned} \mathbf{O} &= \mathrm{GatedMLP}(\mathbf{H}) && (4) \\ &= \mathrm{FC}_{\mathrm{Dn}}(\mathrm{SiLU}(\mathrm{FC}_{\mathrm{Gt}}(\mathbf{H})) \cdot \mathrm{FC}_{\mathrm{Up}}(\mathbf{H})) && (5) \end{aligned}$$

We use a SiLU activation (Elfwing et al., 2018) and match the parameter count of standard attention. This variant serves as a minimal baseline to assess how much attention's effectiveness depends on cross-token interaction, beyond what feed-forward paths alone can provide without using any $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$.

3.3 Relaxing the Mathematical Form

We assess whether attention must strictly follow the canonical dot-product plus softmax formulation. To this end, we evaluate two variants that either approximate or break this form.

Approximate. Following Arora et al. (2024), we preserve the mathematical intention of similarity-based weighting, while relaxing the exact form of softmax via a second-order Taylor expansion, yielding a linear-time recurrent form (Appx. I):

$$\mathbf{A} \approx \mathrm{Taylor}\left(\mathbf{QK}^\top / \sqrt{d_h}\right) \tag{6}$$

$\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are computed using Eq. 3.

Non-approximate. To contrast this, we introduce a new variant that discards explicit pairwise similarity altogether. Instead of computing an attention matrix via $\mathbf{QK}^\top$, it uses element-wise self-gating, multiplying $\mathbf{Q}$ and $\mathbf{K}$ derived from the same hidden state, and normalizes the result across time steps with softmax:

$$\mathbf{A} = \mathrm{Softmax}\left((\mathbf{Q} \odot \mathbf{K})\mathbf{1}^\top / \sqrt{d_h}\right) \tag{7}$$

$$\mathbf{Q} = \mathrm{SiLU}\left(\mathbf{HW}^Q\right); \quad \mathbf{K}, \mathbf{V} = \mathbf{HW}^{K,V} \tag{8}$$

This variant follows an entirely different mathematical form to standard attention. We expect that this should make it harder for adjacent context tokens to receive large attention scores, as the denominator in the softmax computation monotonically increases (see recurrent form in Appx. I). Notably, the SiLU activation is applied element-wise and does not introduce additional complexity. We apply SiLU activation on $\mathbf{Q}$ projection to add non-linearity. This does not require additional parameters and allows parallelism during training.

3.4 Relaxing Sequence Dependency

To test whether attention weights must be dynamically adapted to each input sequence (i.e. sequence-dependent), we construct two variants where $\mathbf{Q}$ and $\mathbf{K}$ are fixed across all inputs, inspired by MLP-Mixer (Fusco et al., 2023; Tolstikhin et al., 2021), but making the parameter count in attention blocks independent of the maximum sequence length. Relaxing sequence dependency allows attention scores for all inputs to be pre-computed and cached during inference.

Random-fixed (RndEmbQK). We initialize a set of random embeddings $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ that remain constant across inputs. These are passed through the Transformer stack up to layer l:

$$\mathbf{X} = \mathrm{TransformerBlock}^{(l)}(\epsilon), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}) \tag{9}$$

$$\mathbf{Q}, \mathbf{K} = \mathbf{XW}^{Q,K}; \quad \mathbf{V} = \mathbf{HW}^V \tag{10}$$

Since $\mathbf{Q}$ and $\mathbf{K}$ do not depend on the input, attention maps are fixed and do not adapt to context.

Text-fixed (FixedSeqQK). Instead of random embeddings, we use a randomly selected fixed sequence of natural language tokens $\mathbf{t}^s$ (first 2048 tokens from FineWeb-10BT (Lozhkov et al., 2024)). These are embedded and passed through the Transformer to generate $\mathbf{X}$:

$$\mathbf{X} = \mathrm{TransformerBlock}^{(l)}(\mathrm{Emb}(\mathbf{t}^s)) \tag{11}$$

$$\mathbf{Q}, \mathbf{K} = \mathbf{X}\mathbf{W}^{Q,K}; \quad \mathbf{V} = \mathbf{H}\mathbf{W}^{V} \tag{12}$$

This setup also produces fixed attention maps, but grounded in natural text instead of completely randomly initialized embeddings. Compared to *RndEmbQK*, it may encode weak structural priors, such as grammatical patterns or token co-occurrences. These variants allow us to test whether dynamic, input-conditioned attention maps are necessary, or whether fixed maps, paired with learned value paths, are sufficient.

3.5 Relaxing the Derivation of Q and K

StaticEmbQK. Finally, to test whether tying $\mathbf{Q}, \mathbf{K}$ to current layer hidden states ($\mathbf{H}$ or $\mathbf{X}$ above) is essential, we compute them directly from static input embeddings $\mathbf{e}$ corresponding to the input sequence $\mathbf{t}$:

$$\mathbf{Q}, \mathbf{K} = \mathbf{e}\mathbf{W}^{Q,K}; \quad \mathbf{V} = \mathbf{H}\mathbf{W}^{V}; \quad \mathbf{e} = \mathrm{Emb}(\mathbf{t}) \tag{13}$$

This means that while attention maps are not fixed, they are computed without contextualization from the evolving hidden representations. It further allows attention scores from different layers to be computed in parallel.

4 Experimental Setup

4.1 Data

We use seven zero-shot NLU tasks in English: ARC-E (Clark et al., 2018), BoolQ (Clark et al., 2019), COPA (Roemmele et al., 2011), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), RTE (Wang et al., 2019) and HellaSwag (Zellers et al., 2019). We also experiment with two LM tasks: WikiText (Merity et al., 2017) and Lambada Openai (Radford et al., 2019).

4.2 Implementation Details

Base model. Our models are built upon Qwen2.5 (Yang et al., 2024a). However, we replace its standard attention mechanism with the alternative attention modules detailed in § 3. To ensure a strict parameter count match across all attention variants, we use multi-head attention (Vaswani et al., 2017), deviating from Qwen2.5's default grouped-query attention (Ainslie et al., 2023). For tokenization, we use the 50K English-centric BPE (Sennrich et al., 2016) vocabulary of Pythia (Biderman et al., 2023), offering small memory footprint, and fast training.

Model configurations. We pretrain models with approximately 500M parameters, using two configurations: (1) *Uniform* with simple attention mechanisms across all Transformer layers; (2) *Hybrid* that integrates simple attention mechanisms in odd-numbered layers and standard attention in even-numbered layers. To assess the contribution of the modified attention variants within the *hybrid* configuration, we introduce a configuration where we remove the odd-numbered layers from pre-trained *hybrid* models (*skip*) and evaluate the resulting performance without additional training.

We further test these three configurations by training models of 70 million and 160 million parameters (see Appx. A). We finally explore various alternative *hybrid* configurations such as changing the simple attention replacement ratio, the details of which are presented in § 6. Specific model size details are provided in Appx. L. Meanwhile, we strictly constrain all models with different attention variants to have the same number of parameters to eliminate any effects from differences in size.

Pre-training. All models are pre-trained on the SlimPajama dataset (Soboleva et al., 2023) for up to 15 billion tokens, following Chinchilla scaling laws (Hoffmann et al., 2022). We use a mini-batch size of 500K tokens, aligning with the training budget outlined in Titans (Behrouz et al., 2024). To optimize pre-training efficiency, we use a sequence length of 2048 tokens.[2]

4.3 Predictive Performance Evaluation

We use the LM-evaluation-harness toolkit v0.4.8 (Gao et al., 2024) for evaluation. We report accuracy for all NLU tasks and perplexity (PPL) for LM tasks. For Lambada Openai, we report both.

4.4 Attention Pattern Indicators

Looking at the performance itself may not offer a comprehensive picture of the behavior of the dif-

[2] Details on hyperparameter selection is provided in Appx. J. For both pre-training and evaluation, we use a single AMD Instinct MI300X accelerator.

ferent attention mechanisms we test. To obtain a more granular understanding of their internal workings, we investigate their attention patterns. We compute eight indicators from the attention matrices $\mathbf{A}_j \in \mathbb{R}^{L \times L}$ for each head $j = 1, \dots, n_h$ in a given layer. We specifically focus on *attention sinks*, i.e. over-attending to the initial token in a sequence, and *local patterns* within attention matrices, i.e. prioritizing nearby tokens, following prior work (Xiao et al., 2024; Han et al., 2024).[3]

Entropy (H). Measures the randomness of attention scores. Higher ENTROPY indicates more uniform attention distribution across tokens, similar to mean-pooling: $\mathrm{H} = -\sum_{a \in \mathbf{A}} a \cdot \log(a)$.

Concentration (Conc). Measures the concentration of attention. A higher Frobenius norm $\|\mathbf{A}\|_F$ indicates attention is focused on a limited number of tokens: $\mathrm{Conc} = \|\mathbf{A}\|_F = \sqrt{\sum_{a \in \mathbf{A}} a^2}$.

Head diversity (HeadDiv). Quantifies the variability of attention patterns across different heads. Calculated as the average position-wise standard deviation across heads, higher HEADDIV suggests better use of the multi-head mechanism.

$$\mathrm{HeadDiv} = \frac{2}{L(1+L)} \sum \mathrm{std}(\{\mathbf{A}_1, \dots, \mathbf{A}_{n_h}\})$$

Attention sink (Sink). Detects focus on the first token. It is the average attention score assigned by all queries to the initial token. Higher Sink means a stronger attention sink: $\mathrm{Sink} = \sum \mathbf{A}_{:,1}/L$.

Local Focus (LocFocN). Measures the attention focus on nearby tokens. It is the average attention score for tokens at a fixed relative distance N (here $N \in \{0, 1, 2, 3\}$). Higher LocFocN suggests stronger contribution from local context.

$$\mathrm{LocFocN} = \sum \mathbf{A}_{L-N, L-N} / \, (L - N)$$

5 Results

Tbl. 1 shows the performance of all model variants (§3), employing *uniform*, *hybrid*, and *skip* configurations across NLU and LM tasks. Results illuminate the role each design principle plays in effective language modeling.

Token mixing is crucial. The uniform MLP model, which lacks any cross-token interaction, performs near chance on most NLU tasks, highlighting that token mixing is essential for reasoning and understanding. Despite this, it achieves a much lower perplexity on WikiText (993.5 vs. 300K for *RndEmbQK*), indicating that even without explicit mixing, MLP can memorize or exploit local token statistics, likely unigram or bigram patterns. Introducing token mixing in a hybrid setup substantially improves NLU performance (e.g. 9.2 average accuracy points over uniform MLP), showing that mixing in part of the network can compensate to a degree. Still, the hybrid MLP variant has the highest WikiText perplexity among all hybrids, indicating that token mixing across all layers is important for fully modeling long-range dependencies.

Standard mathematical form is important in uniform. When applied uniformly, variants that retain the core structure of attention (e.g. *Approximate*, *RndEmbQK*, *FixedSeqQK* and *StaticEmbQK*) restore over 92% of the average NLU accuracy of attention. In contrast, *Non-approximate*, which discards this structure, performs close to random guess (39.3 vs. 39.9 on NLU Avg. accuracy). *Approximate* achieves the strongest results among uniform variants (8.8 higher PPL on WikiText), suggesting that preserving or closely approximating its mathematical form appears critical for maintaining predictive performance.

Sequence-dependency enhances the generalization ability. To assess the role of sequence-dependent attention, we compare variants that retain similar architectures but differ in whether attention scores vary across inputs. *StaticEmbQK*, which preserves Sequence-Dependency, consistently outperforms *RndEmbQK* and *FixedSeqQK*, which use fixed attention patterns, particularly on LAMBADA OPENAI by around 2% higher accuracy. This pattern holds across both uniform and hybrid settings. Additionally, hybrid models that preserve sequence-dependency, such as *Approximate*, *StaticEmbQK*, and *Non-approximate*, tend to perform better on global-context benchmarks. These results suggest that input-specific attention contributes to better generalization, even when other attention properties are simplified.

[3] ENTROPY (H), CONC, and HEADDIV are min-max normalized. SINK and LOCFOCN use absolute values (LOCFOCN is scaled by two for visibility). High ENTROPY and low CONC suggest mean-pooling like behavior. High CONC and low ENTROPY indicate focus on a few tokens. Further examination of SINK and LOCFOCN clarifies if this focus is on the first token or local tokens. Low ENTROPY and high CONC with low scores elsewhere (except HEADDIV) may point to sparse attention on mid-sequence tokens.

		ARC-E	**BoolQ**	**COPA**	**PiQA**	**SciQ**	**RTE**	**HellaSwag**	Avg.	**Wiki**	**LAMBADA**	
		acc↑	acc↑	acc↑	acc↑	acc↑	acc↑	acc↑	acc↑	ppl↓	ppl↓	acc↑
	Rnd. Guess	25.0$_{0.0}$	50.0$_{0.0}$	50.0$_{0.0}$	50.0$_{0.0}$	25.0$_{0.0}$	50.0$_{0.0}$	25.0$_{0.0}$	39.9	3E+5	3E+6	0.0$_{0.0}$
	Majority	25.7$_{0.0}$	62.2$_{0.0}$	56.0$_{0.0}$	50.5$_{0.0}$	25.0$_{0.0}$	52.7$_{0.0}$	25.0$_{0.0}$	39.9	-	-	-
	Standard	41.5$_{1.0}$	56.6$_{0.9}$	63.0$_{4.9}$	60.9$_{1.1}$	60.2$_{1.5}$	53.1$_{3.0}$	28.3$_{0.4}$	51.9	**38.1**	134.1	22.9$_{0.5}$
UNIFORM	MLP	28.5$_{0.9}$	37.8$_{0.8}$	54.0$_{5.0}$	54.8$_{1.2}$	25.9$_{1.4}$	52.7$_{3.0}$	26.1$_{0.4}$	40.0	993.5	1E+5	0.0$_{0.0}$
	Approx.	40.7$_{1.0}$	51.5$_{0.9}$	64.0$_{4.8}$	59.9$_{1.1}$	55.0$_{1.6}$	52.3$_{3.0}$	28.1$_{0.4}$	50.2	47.9	238.6	18.5$_{0.5}$
	Non-apx.	26.8$_{0.9}$	37.8$_{0.8}$	60.0$_{4.9}$	53.2$_{1.2}$	19.3$_{1.2}$	52.3$_{3.0}$	26.0$_{0.4}$	39.3	9E+4	2E+6	0.0$_{0.0}$
	RndEmbQK	39.5$_{1.0}$	55.3$_{0.9}$	57.0$_{5.0}$	59.8$_{1.1}$	46.4$_{1.6}$	50.9$_{3.0}$	27.2$_{0.4}$	48.0	84.8	6402.4	1.3$_{0.2}$
	FixedSeqQK	39.4$_{1.0}$	**59.0**$_{0.9}$	61.0$_{4.9}$	59.4$_{1.1}$	51.2$_{1.6}$	52.7$_{3.0}$	27.5$_{0.4}$	50.0	79.1	19578.1	1.4$_{0.2}$
	StaticEmbQK	39.6$_{1.0}$	52.9$_{0.9}$	63.0$_{4.9}$	59.4$_{1.1}$	49.2$_{1.6}$	54.2$_{3.0}$	27.2$_{0.4}$	49.4	79.9	2287.4	3.3$_{0.2}$
HYBRID	MLP	37.5$_{1.0}$	49.8$_{0.9}$	60.0$_{4.9}$	60.2$_{1.1}$	54.3$_{1.6}$	52.7$_{3.0}$	26.1$_{0.4}$	48.7	45.8	228.7	20.8$_{0.6}$
	Approx.	39.9$_{1.0}$	51.5$_{0.9}$	**67.0**$_{4.7}$	60.4$_{1.1}$	60.5$_{1.5}$	53.4$_{3.0}$	28.4$_{0.4}$	51.6	39.4	140.0	23.7$_{0.6}$
	Non-apx.	**42.3**$_{1.0}$	56.8$_{0.9}$	63.0$_{4.9}$	61.7$_{1.1}$	**63.0**$_{1.5}$	54.9$_{3.0}$	**28.5**$_{0.5}$	**52.9**	39.4	**133.1**	**23.8**$_{0.6}$
	RndEmbQK	40.1$_{1.0}$	48.3$_{0.9}$	61.0$_{4.9}$	61.2$_{1.1}$	60.0$_{1.5}$	50.9$_{3.0}$	27.2$_{0.4}$	49.8	39.3	157.5	22.0$_{0.6}$
	FixedSeqQK	40.5$_{1.0}$	58.5$_{0.9}$	64.0$_{4.8}$	**61.9**$_{1.1}$	62.0$_{1.5}$	52.7$_{3.0}$	28.4$_{0.4}$	52.6	38.5	354.7	20.3$_{0.6}$
	StaticEmbQK	39.2$_{1.0}$	54.7$_{0.9}$	64.0$_{4.8}$	60.9$_{1.1}$	58.4$_{1.6}$	**57.4**$_{3.0}$	28.2$_{0.4}$	51.8	38.7	140.7	**23.8**$_{0.6}$
HYBRID SKIP	MLP	24.4$_{0.9}$	41.8$_{0.9}$	54.0$_{5.0}$	52.8$_{1.2}$	19.0$_{1.2}$	46.9$_{1.7}$	25.6$_{0.4}$	37.8	2E+5	5E+6	0.0$_{0.0}$
	Approx.	26.6$_{0.9}$	46.1$_{0.9}$	59.0$_{4.9}$	52.8$_{1.2}$	20.1$_{1.3}$	48.0$_{3.0}$	26.0$_{0.4}$	39.8	2E+6	1E+7	0.0$_{0.0}$
	Non-apx.	26.6$_{0.9}$	39.2$_{0.9}$	52.0$_{5.0}$	51.4$_{1.2}$	20.4$_{1.3}$	46.9$_{3.0}$	25.8$_{0.4}$	37.5	5E+5	9E+6	0.0$_{0.0}$
	RndEmbQK	27.4$_{0.9}$	37.8$_{0.8}$	58.0$_{5.0}$	53.3$_{1.2}$	21.1$_{1.3}$	52.7$_{3.0}$	26.1$_{0.4}$	39.5	2E+4	3E+6	0.0$_{0.0}$
	FixedSeqQK	27.2$_{0.9}$	39.4$_{0.9}$	59.0$_{4.9}$	52.3$_{1.2}$	22.1$_{1.3}$	48.4$_{3.0}$	25.9$_{0.4}$	39.2	2E+5	5E+6	0.0$_{0.0}$
	StaticEmbQK	25.5$_{0.9}$	43.0$_{0.9}$	57.0$_{5.0}$	53.1$_{1.2}$	22.0$_{1.3}$	51.6$_{3.0}$	25.9$_{0.4}$	39.7	7E+4	5E+6	0.0$_{0.0}$

Table 1: Performance of *uniform*, *hybrid*, *skip* and *standard* models (500M). Purple (MLP), blue (Approx., Non-apx.), green (RndEmbQK, FixedSeqQK) and yellow (StaticEmbQK) denote variants that relax `Token Mixing`, `Mathematical Form`, `Sequence-Dependency` and `Current QK`, respectively.

Current QK is not as essential as expected. *StaticEmbQK* relaxes `Current QK`. Though it does not match the PPL of *standard* across language modeling tasks, it results in PPL of 79.9 twice as high as 38.1 of *standard* under *uniform* configuration on WIKITEXT. It also greatly outperforms *MLP*, reducing PPL tenfold (from 993.5 on WIKITEXT), while its predictive performance is comparable to *standard*. Moreover, under *hybrid* configuration, it achieves predictive performance comparable to *standard* baseline across all tasks. It indicates `Current QK` is not as essential for strong predictive performance as initially believed.

Layer collaboration matters. All *hybrid* models where simple attention variants are used in odd layers and standard attention in even layers achieve predictive performance comparable to *Standard* attention on both NLU and language modeling tasks. Surprisingly, *Non-approximate* attention, the worst performer in the uniform configuration, demonstrates strong performance in this *hybrid* setup, slightly surpassing *Standard* on average NLU accuracy (+1.8%) and LAMBADA OPENAI accuracy (+0.9%), while reducing PPL by 1.0. The *hybrid* configuration also alleviates the relatively higher uncertainty observed with *RndEmbQK* and *FixedSeqQK*, halving their WIKITEXT PPL by incorporating standard layers that aid in grounding attention to individual inputs. These findings suggest that layers exhibiting poor performance in isolation can be effective when combined with stronger layers (i.e. standard attention).

Considering the residual connections, which facilitate information flow along a shortcut pathway bypassing the simple attention alternatives, we further conduct an ablation study to constrain information flow solely through these residual connections. This involves skipping the non-*Standard* layers when pre-training `hybrid` models (denoted as `SKIP` in Tbl. 1). The results provide further support to the assumption of layer collaboration. All variants in `w/ SKIP` perform even slightly worse than random guessing (i.e. average accuracy lower than 39.9 on NLU) and further result in PPL explosion in language modeling compared to `hybrid` by a margin. This indicates that the non-*Standard* layers, despite their simplicity or poor performance in uniform configurations, contribute positively to the overall predictive performance in hybrid architectures.

6 Analysis and Discussion

Attention variants. *Non-approximate* attention that relaxes standard attention's `Mathematical Form` appears to be the most challenging to train in a *uniform* configuration. Radar plots in Fig. 1 show very low ENTROPY alongside high CONC and HEADDIV, indicating that most heads place

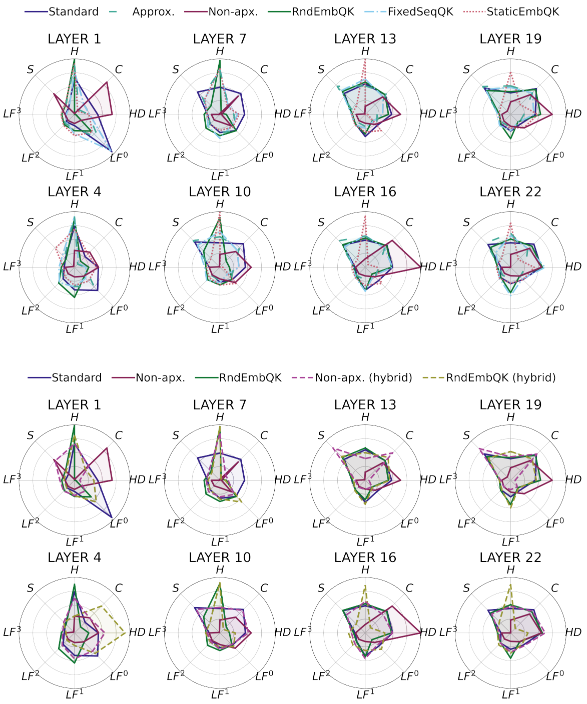


Figure 1: Layer-wise attention indicators for *Approx.*, *Non-approx.*, *RndEmbQK*, *FixedSeqQK* and *StaticEmbQK* in *uniform* (top) and *hybrid* (bottom) configurations, and *Standard* (H: ENTROPY, C: CONC, HD: HEADDIV, LF: LOCFOCN, S: SINK).

almost all probability mass on a narrow set of mid-sequence tokens. This behavior might stem from its monotonically increasing denominators (seeEq. 18). This could make it progressively harder for later tokens in the sequence to attract attention, thereby hindering effective training in uniform configurations. *StaticEmbQK* relaxing `Current QK` coupling, generally presents active token mixing from Layer 7, however, its mid-layers exhibit high similarity. Its reliance on static embeddings for attention computation limits its adaptability to individual layers, further constraining predictive performance. *Approximate* and *FixedSeqQK*, showing attention patterns most similar to *Standard* across all layers. However, the performance of *FixedSeqQK* generally lags behind *Approximate*. This due to *FixedSeqQK*'s derivation of $\mathbf{Q}$ and $\mathbf{K}$ matrices from a fixed, pre-defined text sequence, which remains constant for all inputs. Consequently, the model might become prone to simulating this specific text sequence, thereby compromising its generalization ability. *RndEmbQK* attention faces a similar issue to *FixedSeqQK*, but suffers additional marginal performance drops, perhaps due to its inability to encode syntactic information.

Configurations. To illustrate the impact of different configurations, Fig. 1 shows the attention patterns of *RndEmbQK* and *Non-approximate* variants as representative methods for studying the behavior of different attention variants in *uniform* and *hybrid* configurations (see Fig. 8 for all layers).

With *uniform RndEmbQK* (and *uniform Standard*), the top-most layers (e.g. layer 22) exhibit high concentration (low ENTROPY and high CONC). This indicates a probability mass predominated by few selective tokens. In the hybrid design, those same layers become less selective (higher ENTROPY, lower CONC), leading to a decreased SINK score, suggesting that the hybrid mix alleviates first-token ‘sink’ effects. In the *Non-approximate* hybrid model, odd layers keep the *Non-approximate* heads while even layers revert to *Standard*. A clear division of labor emerges: even (*Standard*) layers mirror the baseline, balancing token mixing and focus, while odd (*Non-approximate*) layers specialize, either acting as attention sinks (high SINK, low ENTROPY) or as mean-poolers (high ENTROPY, low CONC). This complementary interplay compensates for the lower expressiveness of *Non-approximate* heads observed in the uniform setting, explaining why the *hybrid* configuration trains successfully while the *uniform* one does not.

Why hybrid works. We investigate the magnitude of raw activations (logits before softmax) within each *RndEmbQK* and *Non-approximate* layer in the *hybrid* configuration (Fig. 2). Our analysis reveals that activations generally exhibit lower magnitudes compared to the *uniform* configuration for both attention variants. Notably, the *uniform Non-approximate* model shows activation outliers exceeding 10^3 in the final Transformer layers (e.g. Layer 21). In contrast, the *hybrid* configuration maintains activations below 10^1. This suggests that the *Standard* layers in the *hybrid* architecture might serve as a normalization mechanism. This normalization could mitigate over-concentration and the formation of highly sparse attention matrices, which can arise from large magnitude outliers during the numerically stable softmax operation. This normalizing effect appears sufficiently strong to rescue models that are otherwise challenging to train and prone to gradient vanishing (e.g. *Non-approximate* in the *uniform* configuration).

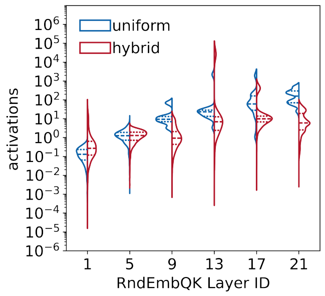


Figure 2: Distribution of pre-softmax activations for *RndEmbQK* (left) and *Non-approximate* (right) across two different configurations. See Fig. 6 for all layers.

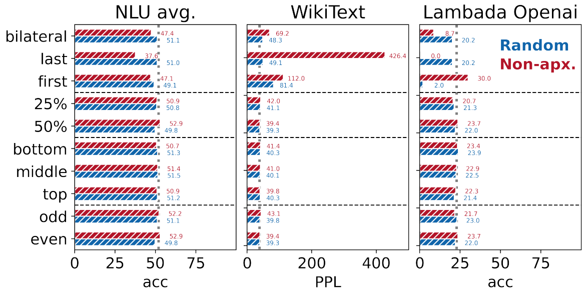


Figure 3: Performance of *RndEmbQK* and *Non-approximate* across nine hybrid configurations. The vertical dotted lines represent the *Standard* baseline.

Theoretical analysis. Li et al. (2024) connects Transformer LMs to spin glass models. They suggest standard attention matrices align with the Gibbs-Boltzmann distribution (Gibbs, 1902), implying an implicit energy minimization process with tokens as spins. Input-independent $\mathbf{Q}$ and $\mathbf{K}$ or form deviations disrupt this. This perspective provides a theoretical basis for the performance variations observed in our uniform replacement experiments. While Zhang et al. (2022) suggests full-rank attention offers maximal flexibility, causal attention can be low-rank due to stable softmax allowing zeros in diagonals with activation outliers. This supports our normalization analysis in *hybrid* configurations, with Neyshabur et al. (2017)'s observation on unbalanced network training difficulty.

Model size. We also evaluate all attention variants across models of 70M, 160M, and 500M parameters. Our main observations remain consistent across these different model sizes. See Appx. A for detailed results.

Hybrid configuration ablation. To investigate the impact of replacing subsets of layers with simpler attention mechanisms, we consider nine different configurations. These focus on different segments of a 24-layer architecture of the 500M model: (1) *even* or *50%* configuration, where even-numbered layers retain standard attention while odd-numbered layers are replaced; (2) *odd* configuration, with the reverse arrangement; (3) *top* configuration, where the upper layers (13-24) employ the simpler attention mechanism; (4) *middle* configuration, targeting the middle layers (7-18); (5) *bottom* configuration, focusing on the initial layers (1-6); (6) *25%*, replacing layers except Layer 4,8,12,16,20,24 with simpler attention; (7) *first*, replacing all layers with simpler attention except the first layer; (8) *last*, replacing all layers with simpler attention except the last layer; (9) *bilateral*, replacing all layers with simpler attention except Layer 1 and 24. See Tbl. 11 in Appx. P for details.

Fig. 3 presents the predictive performance using these nine settings. For both *RndEmbQK* and *Non-approximate* mechanisms, the difference in performance across these hybrid configurations is marginal (e.g. all with a PPL around 40.0 on WikiText). However, this observation does not generalize to extreme settings, such as employing *Standard* attention in only the first or the last layer. For *RndEmbQK* attention, the predictive performance remains comparable to *Standard* if only the last layer (or layers at both ends) uses *Standard.* Nevertheless, its accuracy on Lambada Openai drops to zero in such extreme cases. For *Non-approximate* attention, using *Standard* attention mechanism only in the last layer greatly harms performance, leading to PPL exceeding 400 on WikiText. This indicates that the normalization strength provided by a single *Standard* layer is limited. Therefore, in extreme hybrid settings where we can afford only one or two *Standard* layers, we should choose a substitute that still respects the main design principles presented in the *uniform* setting (i.e. a stronger lightweight attention). Conversely, if the compute budget allows using even a small fraction of *Standard* transformer layers (e.g. 25%), we can safely replace the remainder with a much simpler mechanism and still maintain competitive accuracy.

7 Conclusion

We systematically relax core design principles in a controlled setting, offering the first principled framework for assessing which aspects of attention are truly foundational and which can be safely simplified in language modeling. Our findings reveal that adhering to standard attention design principles varies between *uniform* and *hybrid* architectures. Token mixing and following the mathematical form are crucial for attention alternatives when applied uniformly, but not necessary for *hybrid*. Strategically integrating a few standard attention layers within LMs can greatly improve, even overcome, limitations of less powerful attention mechanisms. This is likely due to the inherent normalization of standard attention, fostering training stability.

Limitations

We performed experiments using a maximum model size of 500M parameters and a pretraining budget of 15B tokens, using a monolingual tokenizer and vocabulary, similar to Allal et al. (2025); Poli et al. (2023). While experimenting with larger models and different model families presents interesting avenues for future work, we believe that the current scope sufficiently supports our conclusions regarding the relative effectiveness of different attention designs.

Acknowledgments

This projected made use of time on UK Tier-2 HPC facility JADE@ARC, funded by EPSRC (EP/T022205/1). We would like to thank Miles Williams and Atsuki Yamaguchi for their invaluable feedback.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Yaroslav Aksenov, Nikita Balagansky, Sofia Lo Cicero Vaina, Boris Shaposhnikov, Alexey Gorbatovski, and Daniil Gavrilov. 2024. Linear transformers with learnable kernel functions are better in-context models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9584–9597, Bangkok, Thailand. Association for Computational Linguistics.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlícek, Agustín Piqueres Lajarín, Vaibhav Srivastav, and 1 others. 2025. Smollm2: When smol goes big-data-centric training of a small language model. *CoRR*.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Re. 2024. Simple linear attention language models balance the recall-throughput tradeoff. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 1763–1840. PMLR.

Alan Baker. 2022. Simplicity. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Summer 2022 edition. Metaphysics Research Lab, Stanford University.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. 2024. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try arc, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the*

41st International Conference on Machine Learning, pages 10041–10071.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, ZIJIA CHEN, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. 2025. Hymba: A hybrid-head architecture for small language models. In *The Thirteenth International Conference on Learning Representations*.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.

Francesco Fusco, Damian Pascual, Peter Staar, and Diego Antognini. 2023. pNLP-mixer: an efficient all-MLP architecture for language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 53–60, Toronto, Canada. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Josiah Willard Gibbs. 1902. *Elementary principles in statistical mechanics: Developed with especial reference to the rational foundations of thermodynamics*. C. Scribner's sons.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7B SSM hybrid model. *arXiv preprint arXiv:2405.16712*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

Zhihao He, Hang Yu, Zi Gong, Shizhan Liu, Jianguo Li, and Weiyao Lin. 2025. Rodimus*: Breaking the accuracy-efficiency trade-off with efficient attentions. In *The Thirteenth International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021. Finetuning pretrained transformers into RNNs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10630–10643, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Barak Lenz, Opher Lieber, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen

Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M. Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, and 42 others. 2025. Jamba: Hybrid Transformer-Mamba language models. In *The Thirteenth International Conference on Learning Representations*.

Yuhao Li, Ruoran Bai, and Haiping Huang. 2024. Spin glass model of in-context learning. *arXiv preprint arXiv:2408.02288*.

Zhixuan Lin, Evgenii Nikishin, Xu He, and Aaron Courville. 2025. Forgetting transformer: Softmax attention with a forget gate. In *The Thirteenth International Conference on Learning Representations*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. FineWeb-Edu: The finest collection of educational content .

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, and 1 others. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–15.

Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. 2023. Efficient transformers with dynamic token pooling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6403–6417, Toronto, Canada. Association for Computational Linguistics.

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. 2017. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, and 13 others. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore. Association for Computational Linguistics.

Bo Peng, Daniel Goldstein, Quentin Gregory Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Teddy Ferdinan, Kranthi Kiran GV, Haowen Hou, Satyapriya Krishna, Ronald McClelland Jr., Niklas Muennighoff, Fares Obeid, Atsushi Saito, Guangyu Song, Haoqin Tu, Ruichong Zhang, Bingchen Zhao, and 3 others. 2024. Eagle and Finch: RWKV with matrix-valued states and dynamic recurrence. In *First Conference on Language Modeling*.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Haowen Hou, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, and 1 others. 2025. RWKV-7" Goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*.

Dazhi Peng and Hangrui Cao. 2024. E-Tamba: Efficient Transformer-Mamba layer transplantation. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2021. Random feature attention. In *International Conference on Learning Representations*.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. 2022. cosFormer: Rethinking Softmax In Attention. In *International Conference on Learning Representations*.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. 2024. HGRN2: Gated linear RNNs with state expansion. In *First Conference on Language Modeling*.

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

Hossein Rajabzadeh, Aref Jafari, Aman Sharma, Benyamin Jami, Hyock Ju Hj Kwon, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2024. Echoatt: Attend, copy, then adjust for more efficient large language models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*, pages 259–269. PMLR.

Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. 2025. Vamba: Understanding hour-long videos with hybrid mamba-transformers. *arXiv preprint arXiv:2503.11579*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pages 9355–9366. PMLR.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. 2025. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. *arXiv preprint arXiv:2502.10297*.

Ajit Singh. 2025. Meta Llama 4: The future of multimodal AI. *Available at SSRN 5208228*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, and 1 others. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28.

Yi Tay, Aston Zhang, Anh Tuan Luu, Jinfeng Rao, Shuai Zhang, Shuohang Wang, Jie Fu, and Siu Cheung Hui. 2019. Lightweight and efficient neural natural language processing with quaternion networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1494–1503, Florence, Italy. Association for Computational Linguistics.

Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, and 1 others. 2024. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, and 1 others. 2021. MLP-Mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Tong Xiao, Yinqiao Li, Jingbo Zhu, Zhengtao Yu, and Tongran Liu. 2019. Sharing attention weights for fast transformer. *arXiv preprint arXiv:1906.11024*.

Huiyin Xue and Nikolaos Aletras. 2022. HashFormers: Towards vocabulary-independent pre-trained transformers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7862–7874, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Huiyin Xue and Nikolaos Aletras. 2023. Pit one against many: Leveraging attention-head embeddings for parameter-efficient multi-head attention. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10355–10373, Singapore. Association for Computational Linguistics.

Yu Yan, Jiusheng Chen, Weizhen Qi, Nikhil Bhendawade, Yeyun Gong, Nan Duan, and Ruofei Zhang. 2021. El-attention: Memory efficient lossless attention for generation. In *International Conference on Machine Learning*, pages 11648–11658. PMLR.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others.

2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024b. Parallelizing linear transformers with the delta rule over sequence length. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. 2022. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2022. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

A Experiments with Different Model Sizes

To assess the impact of model size, we evaluate all attention mechanisms across models with approximately 70M, 160M, and 500M parameters. Fig. 4 illustrates the predictive performance of these models on the WIKITEXT, ARC-E, and SCIQ datasets. Our results indicate that the predictive performance of LMs with a *hybrid* configuration consistently improves with increasing model size. For instance, the accuracy of the *Non-approximate* method on ARC-E improves from 34.3 to 42.3 when increasing the model size from 70M to 500M. Furthermore, all attention mechanisms incorporating token mixing achieve predictive performance comparable to a same-sized model employing *standard* attention (indicated by the vertical dotted lines in Fig. 4). For *RndEmbQK*, such performance gap on WIKITEXT PPL is even within 1.2 across all sizes. This trend suggests that our observations may generalize to larger models.

To further investigate the immediate generalizability of our findings, we further pretrain a larger model (Yang et al., 2025, Qwen3-1.7b-Base) from scratch on 45 billion tokens with *Standard* and our proposed *RndEmbQK* and *Non-approximate* variants in both uniform and hybrid configurations. Tbl. 2 presents their performance on NLU and LM tasks. We find both *RndEmbQK* and *Non-apx.* under *hybrid* configuration, achieve performance comparable to *Standard* across all downstream tasks, which is consistent to our observation on models with modest scales. However, different to the model with 500M parameters, *Non-approximate* under *uniform* configuration successfully converges. This is because Qwen3 incorporates RSMNorm above the queries and key in its attention module. This normalization helps to alleviate the potential for pre-softmax attention activations to explode, but it is less effective than using several standard layers, as it restricts the length of query and key vectors, narrowing the adaptable range for raw pre-softmax activations.

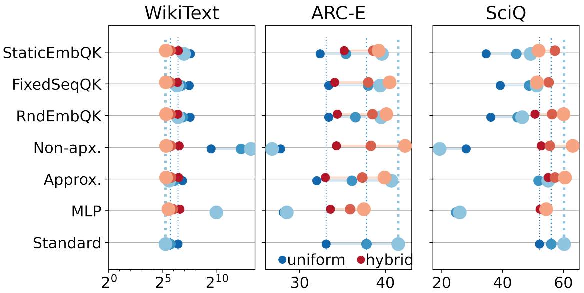


Figure 4: Predictive performance of 70M parameters (small dots), 160M parameters (medium dots), and 500M parameters (large dots) models with different attention mechanisms and configurations on WIKITEXT, ARC-E, and SCIQ.

B Grouped-query Attention Ablation

To confirm the generality of our main investigations, we also trained 500M parameter versions of the *Standard*, *Non-approximate*, and *RndEmbQK* models using the grouped-query attention configuration. These models are trained on the same 15 billion tokens, with precisely matched parameter counts. We observe that the results on downstream tasks remain consistent across both the multi-head attention and grouped-query attention configurations. Their performance on both NLU and LM tasks is detailed in Tbl. 3.

C Robustness to Context Length

Tbl. 4 illustrates the perplexity scores of the UNIFORM, HYBRID and *tandard* models on WIKITEXT dataset. These models were evaluated across various contextual lengths (128, 256, 512, 1024, and

		ARC-E acc↑	BoolQ acc↑	COPA acc↑	PiQA acc↑	SciQ acc↑	RTE acc↑	HellaSwag acc↑	Avg. acc↑	Wiki ppl↓	LAMBADA ppl↓	 acc↑
	Rnd. Guess	$25.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$25.0_{0.0}$	$50.0_{0.0}$	$25.0_{0.0}$	39.9	3E+5	3E+6	$0.0_{0.0}$
	Majority	$25.7_{0.0}$	$62.2_{0.0}$	$56.0_{0.0}$	$50.5_{0.0}$	$25.0_{0.0}$	$52.7_{0.0}$	$25.0_{0.0}$	39.9	-	-	-
	Standard	$44.6_{1.0}$	$56.4_{0.9}$	$64.0_{4.8}$	$64.0_{1.1}$	$67.3_{1.5}$	$52.7_{3.0}$	$30.5_{0.5}$	54.2	27.6	60.0	$28.9_{0.6}$
UNI.	Non-apx.	$41.0_{1.0}$	$61.9_{0.9}$	$58.0_{5.0}$	$59.5_{1.2}$	$56.9_{1.6}$	$52.4_{3.0}$	$27.8_{0.5}$	51.1	67.3	619.6	$8.1_{0.4}$
	RndEmbQK	$44.4_{1.0}$	$50.2_{0.9}$	$60.0_{4.9}$	$62.4_{1.1}$	$56.3_{1.6}$	$54.9_{3.0}$	$28.6_{0.5}$	51.0	54.9	1872.8	$3.8_{0.3}$
HYB.	Non-apx.	$\mathbf{45.0}_{1.0}$	$58.1_{0.9}$	$65.0_{4.8}$	$63.4_{1.1}$	$\mathbf{66.2}_{1.5}$	$53.1_{3.0}$	$\mathbf{30.2}_{0.5}$	54.4	29.9	77.4	$\mathbf{26.8}_{0.6}$
	RndEmbQK	$45.4_{1.0}$	$57.0_{0.9}$	$67.0_{4.7}$	$64.5_{1.1}$	$65.5_{1.5}$	$55.2_{3.0}$	$30.4_{0.5}$	55.0	28.0	61.6	$29.8_{0.6}$

Table 2: Performance of *uniform*, *hybrid* and *standard* models (1.7B). Blue (Non-apx.) and green (RndEmbQK) denote variants that relax `Mathematical Form`, and `Sequence-Dependency`, respectively.

		ARC-E acc↑	BoolQ acc↑	COPA acc↑	PiQA acc↑	SciQ acc↑	RTE acc↑	HellaSwag acc↑	Avg. acc↑	Wiki ppl↓	LAMBADA ppl↓	 acc↑
	Rnd. Guess	$25.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$50.0_{0.0}$	$25.0_{0.0}$	$50.0_{0.0}$	$25.0_{0.0}$	39.9	3E+5	3E+6	$0.0_{0.0}$
	Majority	$25.7_{0.0}$	$62.2_{0.0}$	$56.0_{0.0}$	$50.5_{0.0}$	$25.0_{0.0}$	$52.7_{0.0}$	$25.0_{0.0}$	39.9	-	-	-
	Standard	$39.4_{1.0}$	$49.6_{0.9}$	$60.0_{5.0}$	$62.2_{1.1}$	$59.3_{1.6}$	$51.6_{3.0}$	$28.1_{0.5}$	50.0	38.6	154.0	$22.9_{0.6}$
UNI.	Non-apx.	$26.8_{0.9}$	$37.8_{0.8}$	$52.0_{5.0}$	$52.0_{1.2}$	$20.3_{1.3}$	$52.7_{3.0}$	$25.9_{0.4}$	38.2	5466.8	2E+6	$0.0_{0.0}$
	RndEmbQK	$37.9_{1.0}$	$53.2_{0.9}$	$56.0_{5.0}$	$58.3_{1.2}$	$46.7_{1.6}$	$52.7_{3.0}$	$27.2_{0.4}$	47.4	84.6	6462.7	$12.4_{0.2}$
HYB.	Non-apx.	$40.7_{1.0}$	$44.6_{0.9}$	$67.0_{5.0}$	$61.3_{1.1}$	$61.5_{1.5}$	$52.4_{3.0}$	$28.3_{0.5}$	50.8	38.1	133.1	$23.4_{0.6}$
	RndEmbQK	$40.1_{1.0}$	$45.8_{0.9}$	$63.0_{4.9}$	$61.3_{1.1}$	$61.8_{1.5}$	$52.7_{3.0}$	$28.3_{0.5}$	50.4	39.3	138.6	$23.6_{0.6}$

Table 3: Performance of *uniform*, *hybrid* and *standard* models (500m) using grouped-query attention. Blue (Non-apx.) and green (RndEmbQK) denote variants that relax `Mathematical Form`, and `Sequence-Dependency`, respectively.

PPL	**length**	**128**	**256**	**512**	**1024**	**2048**
	Standard	69.9	56.2	47.8	42.0	38.1
UNIFORM	MLP	993.5	993.5	993.5	993.5	993.5
	Approx.	81.7	66.4	57.2	51.2	47.9
	Non-apx.	10023.7	9476.8	9173.5	9064.8	9025.9
	RndEmbQK	107.1	95.7	89.6	86.3	84.8
	FixedSeqQK	100.5	89.6	83.6	80.6	79.1
	StaticEmbQK	104.8	92.2	85.5	81.7	79.9
HYBRID	MLP	81.3	66.4	57.0	50.3	45.8
	Approx.	71.8	57.8	49.3	43.4	39.4
	Non-apx.	69.0	56.0	48.0	42.5	39.4
	RndEmbQK	72.4	58.1	49.4	43.4	39.3
	FixedSeqQK	69.0	56.0	48.0	42.3	38.5
	StaticEmbQK	70.1	56.6	48.4	42.6	38.7

Table 4: Perplexities of *uniform*, *hybrid* and *standard* models (500M) on WIKITEXT across different context lengths.

2048 tokens), all while being trained on a maximum sequence length of 2048 tokens. The results clearly show that models incorporating token mixing achieve lower perplexity scores with longer contexts. This indicates their ability to capture more contextual information for predicting the next token. Furthermore, under the *hybrid* configuration, the perplexity scores for the *RndEmbQK*, *FixedSeqQK*, *StaticEmbQK*, *Approximate* and *Non-approximate* attention mechanisms consistently match those of the *standard* model on WIKITEXT, regardless of contextual length.

D Characteristics of Different Simpler Attentions

Unlike previous work that primarily focused on reducing computational time complexity to subquadratic with respect to contextual sequence length, we define "simpler attention" more broadly. This encompasses mechanisms that reduce time complexity concerning any factor: inference batch size, sequence length, or hidden dimension. Below, we systematically summarize the characteristics of the different simpler attention mechanisms we investigated.

RndEmbQK and FixedSeqQK. These mechanisms create global static attention graphs during inference. This approach reduces the computational time complexity and cache size within attention while enabling batched decoding (see Appx. E and H).

StaticEmbQK. Inspired by cross-layer attention sharing (Rajabzadeh et al., 2024; Xiao et al., 2019), this mechanism primarily captures semantic similarities between input tokens without contextualization. It establishes an upper bound for broadcasting attention matrices from initial layers to all subsequent layers by aligning its parameter count with standard attention. While *StaticEmbQK* attention does not explicitly reduce computational time complexity, it allows for system optimization by computing attention scores asynchronously. This enables scores to be prefetched before sequentially retrieving output hidden states from each layer.

Approximate and Non-approximate. These attention mechanisms result in time complexities linear to sequence length. Their recurrent forms are detailed in Appx. I. *Non-approximate* can further reduce the activation memory, cache size, and floating-point operations per iteration (FLOPs/it) required for large LMs during the decode stage, offering advantages over *Approximate*. The details for these reductions are provided in Appendices G, H, and F, respectively.

E Time Complexities in Attention Computation

Tbl. 5 details the computational time complexity for a single forward pass, explicitly excluding any caching mechanisms. For *RndEmbQK* and *FixedSeqQK* attention, which employ global attention scores, the floating-operations could be further reduced to through pre-computation and subsequent caching of these scores (see Appx. F). This optimization would free up computational resources, enabling further software-level enhancements such as coordinating CPUs and GPUs to pre-fetch the pre-calculated attention scores. While *StaticEmbQK* does not inherently offer a lower computational time complexity, it provides an upper bound for pre-computing attention scores on static embeddings by aligning the number of parameters. If attention scores on static embeddings are pre-computed, the computational time complexity would be reduced by $O\left((l-1)\cdot(BL^2d+BLd^2)\right)$ in total, where l represents the total number of Transformer layers. Furthermore, an attention mechanism that supports pre-computation offers the potential to proactively evict values, which could lead to further reductions in computation, particularly if the attention matrices exhibit sparsity.

Attention	**Complexity** $\mathcal{O}(.)$
Standard	$BL^2d + BLd^2$
MLP	BLd^2
Approx.	BLd^2
Non-apx.	BLd^2
RndEmbQK	$BL^2d + BLd^2$
FixedSeqQK	$BL^2d + BLd^2$
StaticEmbQK	$BL^2d + BLd^2$

Table 5: Details of time complexities for each attention across all attention variants, where h denotes the number of attention heads, B denotes the batch size, L denotes the input sequence length, d denotes the hidden dimension. We assume $d = h \times d_h$, where h is the number of attention heads and d_h is the dimension of each attention head. We also ignore those low-order terms for element-wise activations and scaling factors with a $\mathcal{O}(BLd)$ complexity.

F Floating-point Operations per Token

Tbl. 6 details the floating-point operations per iteration (FLOP/it) for inference with the cache enabled. We focus solely on General Matrix Multiplications (GEMMs) (Narayanan et al., 2021, GEMMs), as they are the dominant contributors to the total floating-point operations.

Attention	**Prefill**	**Decode**
Standard	$4BL^2d + 6BLd^2$	$6Bd^2 + 4BLd$
MLP	$6BLd^2$	$6Bd^2$
Approx.	$14BLd^2$	$10Bd^2$
Non-apx.	$6BLd^2$	$6Bd^2$
RndEmbQK	$2L^2d + 2BL^2d + 6BLd^2$	$2Ld + 2BLd + 6Bd^2$
FixedSeqQK	$2L^2d + 2BL^2d + 6BLd^2$	$2Ld + 2BLd + 6Bd^2$
StaticEmbQK	$4BL^2d + 6BLd^2$	$6Bd^2 + 4BLd$

Table 6: Details of floating-point operations per iteration for each attention across all attention variants, where h denotes the number of attention heads, B denotes the batch size, L denotes the input sequence length, d denotes the hidden dimension. We assume $d = h \times d_h$, where h is the number of attention heads and d_h is the dimension of each attention head.

Non-approximate achieves a low FLOP/it, equivalent to that of the simplest *MLP* model, because it leverages vectors instead of the matrices employed by the Approximate method for state tracking. This structural difference significantly reduces the number of GEMMs required.

Furthermore, if *RndEmbQK* and *FixedSeqQK* are allowed to use pre-computed global attention scores, their FLOP/it can be further reduced. During the prefill stage, the operations drop to $2L^2d + 2BLd^2$ and $2BLd + 2d^2$ during prefill and decode stage respectively.

Attention	Activation memory
Standard	$\frac{8BLd+2BL^2h}{t}$
MLP	$\frac{8BLd}{t}$
Approx.	$\frac{11BLd}{t}+\frac{3Bd^2}{ht}$
Non-apx.	$\frac{8BLd+4BLh}{t}$
RndEmbQK	$\frac{4BLd+8Ld+2L^2h}{t}$
FixedSeqQK	$\frac{4BLd+8Ld+2L^2h}{t}$
StaticEmbQK	$\frac{8BLd+2BL^2h}{t}$

Table 7: Details of activation memory for each attention across all attention variants, where h denotes the number of attention heads, B denotes the batch size, L denotes the input sequence length, d denotes the hidden dimension, t denotes the tensor parallel size. We assume $d = h \times d_h$, where h is the number of attention heads and d_h is the dimension of each attention head. We ignore the attention dropout here.

G Activation Memory Required for Attention Computation

We detail the activation memory required for half-precision training in Tbl. 7. Unlike the full recomputation method mentioned in Smith et al. (2022), our approach incorporates sequence parallelism following Korthikanti et al. (2023). We find that *RndEmbQk* and *FixedSqeQK* are effective at reducing activation memory, particularly when using a substantially large batch size. Furthermore, both *Approximate* and *Non-approximate* enhance memory efficiency for long-context processing. *Non-approximate* offers a superior reduction in activation memory compared to *Approximate*, especially for large LMs characterized by a relatively large hidden state dimension.

H Cache Size Required for Inference

Tbl. 8 presents the cache size required for half-precision inference. Both the *Approximate* and *Non-approximate* variants allow the cache size to be independent of the context sequence length. Meanwhile, *RndEmbQk* and *FixedSeqQK* can reduce the cache size by nearly half by sharing the same set of keys within the same batch, provided the batch size is sufficiently large. It is also important to note that *RndEmbQK* and *FixedSeqQk* enable a cache size further optimized to $(2L+\delta)\delta$. This can be achieved by using a dynamic cache and prefetching the attention scores for the next δ steps into a buffer, given that the attention matrices are independent of the inputs.

Attention	Cache Size for Inference
Standard	$4BLd$
MLP	0
Approx.	$6Bd + 4Bd^2/h$
Non-apx.	$2Bd + 4Bh$
RndEmbQK	$2(B+1)Ld$
FixedSeqQK	$2(B+1)Ld$
StaticEmbQK	$4BLd$

Table 8: Details of cache size (in bytes) per layer across all attention variants required during inference, where h denotes the number of attention heads, B denotes the batch size, L denotes the context length, d denotes the hidden dimension. We assume $d = h \times d_h$, where h is the number of attention heads and d_h is the dimension of each attention head.

I Recurrent Form of Linear Attentions

The recurrent form of the *Approximate* attention computation, derived from Eq. 6, is presented in Eq. 17. Similarly, Eq. 18 shows the recurrent form of the *Non-approximate* attention computation, originating from Eq. 7. As detailed in Tbl. 5, the *Approximate* attention mechanism necessitates the computation of recursions for both first-order and second-order terms in the Taylor expansion, resulting in a higher time complexity compared to the *Non-approximate* approach. A key characteristic of $\mathbf{O}_i$ in Eq. 18 is that its denominator strictly increases with the index i. Notably, as i grows along the sequence, the attention score for the i^{th} token, given by $\frac{\mathrm{e}^{q_i k_i^\top} v_i}{\sum_{j=1}^{i-1} \mathrm{e}^{q_j k_j^\top} + \mathrm{e}^{q_i k_i^\top}}$, becomes progressively more challenging to increase.

Hyperparameters in Pretraining	
Maximum train steps	120000
Batch size (in total)	256 instances
Adam *epsilon*	1e-8
Adam β_1	0.9
Adam β_2	0.9999
Sequence length	2048
Peak learning rate	4e-4 (3e-4 for Qwen3-1.7B)
Learning rate schedule	CosineLRScheduler
Number of cycles in scheduler	0.5
Warmup steps	2000 (1B tokens)
Weight decay	0.1
Max gradient norm clip value	1.0

Table 9: Details of hyperparameters used in pre-training.

$$\mathbf{o}_i = \mathbf{o}_{0i} + \mathbf{o}_{1i} + \mathbf{o}_{2i} \tag{14}$$

$$\mathbf{o}_{0i} = \frac{\sum_{j=1}^{i-1} v_j + v_i}{i} \tag{15}$$

$$\mathbf{o}_{1i} = \frac{q_i\left(\sum_{j=1}^{i-1} k_j^\top v_j + k_i^\top v_i\right)}{q_i\left(\sum_{j=1}^{i-1} k_j^\top + k_i^\top\right)} \tag{16}$$

$$\mathbf{o}_{2i} = \frac{\frac{q_i^2}{\sqrt{2}}\left(\sum_{j=1}^{i-1}(\frac{k_j^2}{\sqrt{2}})^\top v_j + (\frac{k_i^2}{\sqrt{2}})^\top v_i\right)}{\frac{q_i^2}{\sqrt{2}}\left(\sum_{j=1}^{i-1}(\frac{k_j^2}{\sqrt{2}})^\top + (\frac{k_i^2}{\sqrt{2}})^\top\right)} \tag{17}$$

$$\mathbf{o}_i = \frac{\sum_{j=1}^{i-1} e^{q_j k_j^\top} v_j + e^{q_i k_i^\top} v_i}{\sum_{j=1}^{i-1} e^{q_j k_j^\top} + e^{q_i k_i^\top}} \tag{18}$$

J Hyperparameters

The hyperparameters used in pre-training are listed in Tbl. 9.

K Training Loss across all Attention Mechanisms

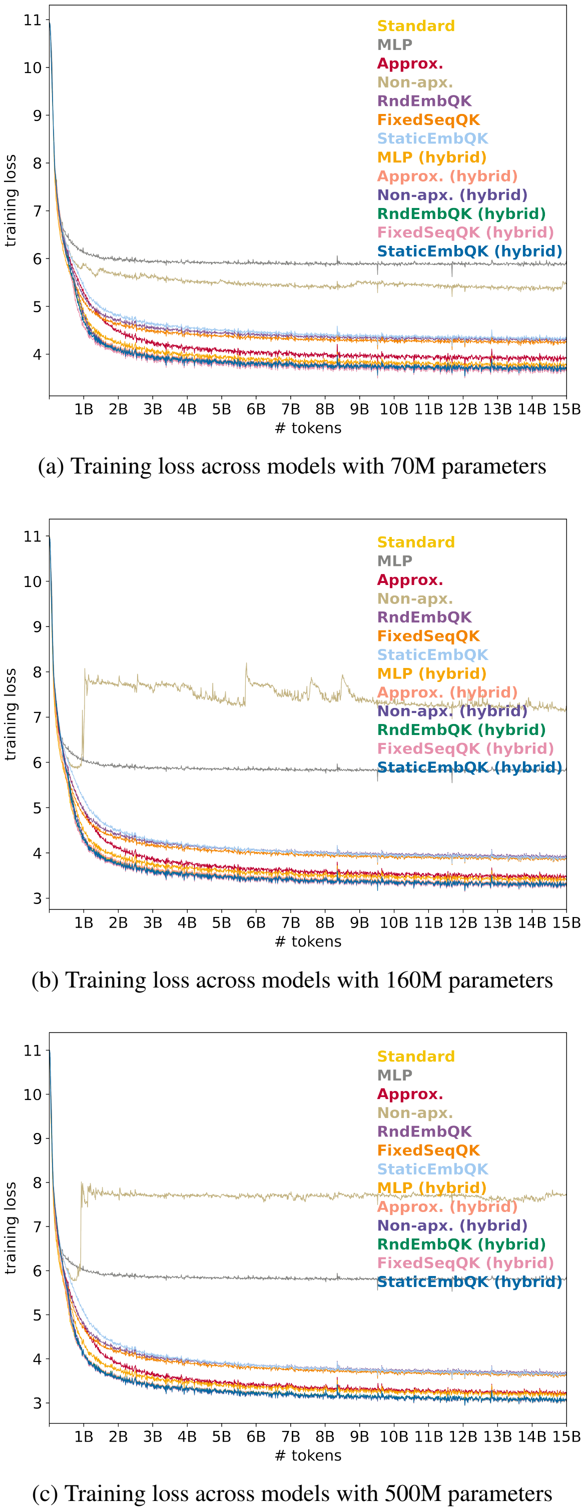
Fig. 5 presents the loss curves across all model variants and sizes, while training for 15B tokens.

L Model Configurations for Different Sizes

Tbl. 10 presents the detailed configurations of models across various sizes (70M, 160M, 500M and 1.7B).

M Distribution of Raw Logits

Fig. 6 (the full version of Fig. 2) exhibits the magnitude of pre-softmax activations within each 24-layer (500M) *RndEmbQK* and *Non-approximate* layer in the *hybrid* configuration.



(a) Training loss across models with 70M parameters

(b) Training loss across models with 160M parameters

(c) Training loss across models with 500M parameters

Figure 5: Training loss across all model variants with three different sizes.

N Attention Characteristics from All Layers across Attention Variants

Fig. 7, the full version of the left subfigure in Fig. 1), exhibits attention characterstics from all 24 layers across *Standard* attention and five attention

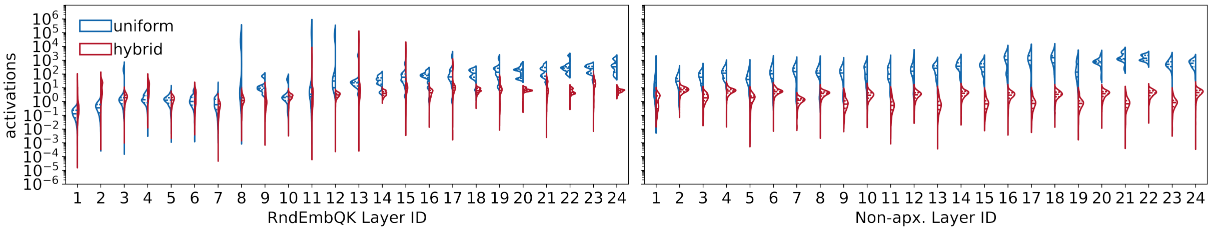


Figure 6: Distribution of raw logits in the pre-softmax activations for *RndEmbQK* (left) and *Non-approximate* (right) attention mechanisms in both uniform and hybrid configurations.

Model Size	70M	160M	500M	1.7B
Hidden Size	512	768	896	2048
Intermediate Size	2048	3072	4864	6144
Num of Hidden Layers	6	12	24	28
Max Window Layers	6	12	24	28
Num of Attention Heads	8	12	14	16
Num of Key Value Heads	8	12	14	16

Table 10: Details of model configurations for different sizes.

Config	Standard Layer IDs
even (50%)	{2,4,6,8,10,12,14,16,18,20,22,24}
odd	{1,3,5,7,9,11,13,15,17,19,21,23}
top	{1,2,3,4,5,6,7,8,9,10,11,12}
middle	{1,2,3,4,5,6,19,20,21,22,23,24}
bottom	{13,14,15,16,17,18,19,20,21,22,23,24}
25%	{4,8,12,16,20,24}
first	{1}
last	{24}
bilteral	{1,24}

Table 11: Details of model configurations for ablation study.

variants - *RndEmbQK*, *FixedSeqQK*, *StaticEmbQK*, *Approximate* and *Non-approximate*.

O Attention Characteristics from All Layers across Configurations

Fig. 8, the full version of the right subfigure in Fig. 1, exhibits attention characterstics from all 24 layers across *Standard* and two representative attention variants - *Approximate* and *Non-approximate* in both *uniform* and *hybrid* configurations.

P Model Configurations for Ablation Study

Tbl. 11 details nine distinct *hybrid* architectures, as discussed in § 6, for 24-layer model variants with approximately 500 million parameters.

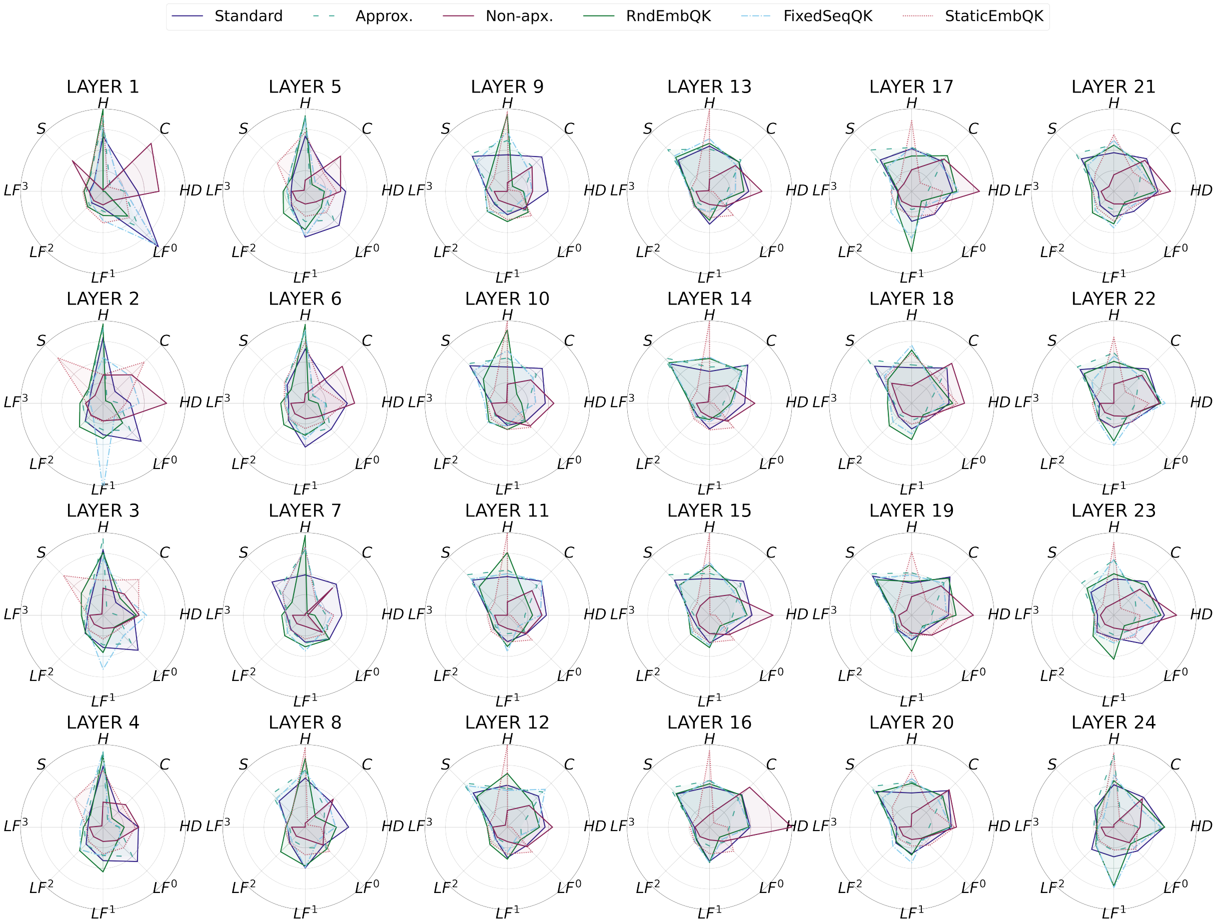


Figure 7: Visualization of attention matrix characteristics across different layers for *Approximate*, *Non-approximate*, *RndEmbQK*, *FixedSeqQK* and *StaticEmbQK*, and their hybrid variants, compared to *Standard* (H: Entropy, C: Conc, HD: HeadDiv, LF: LocFoc$N$, S: Sink).

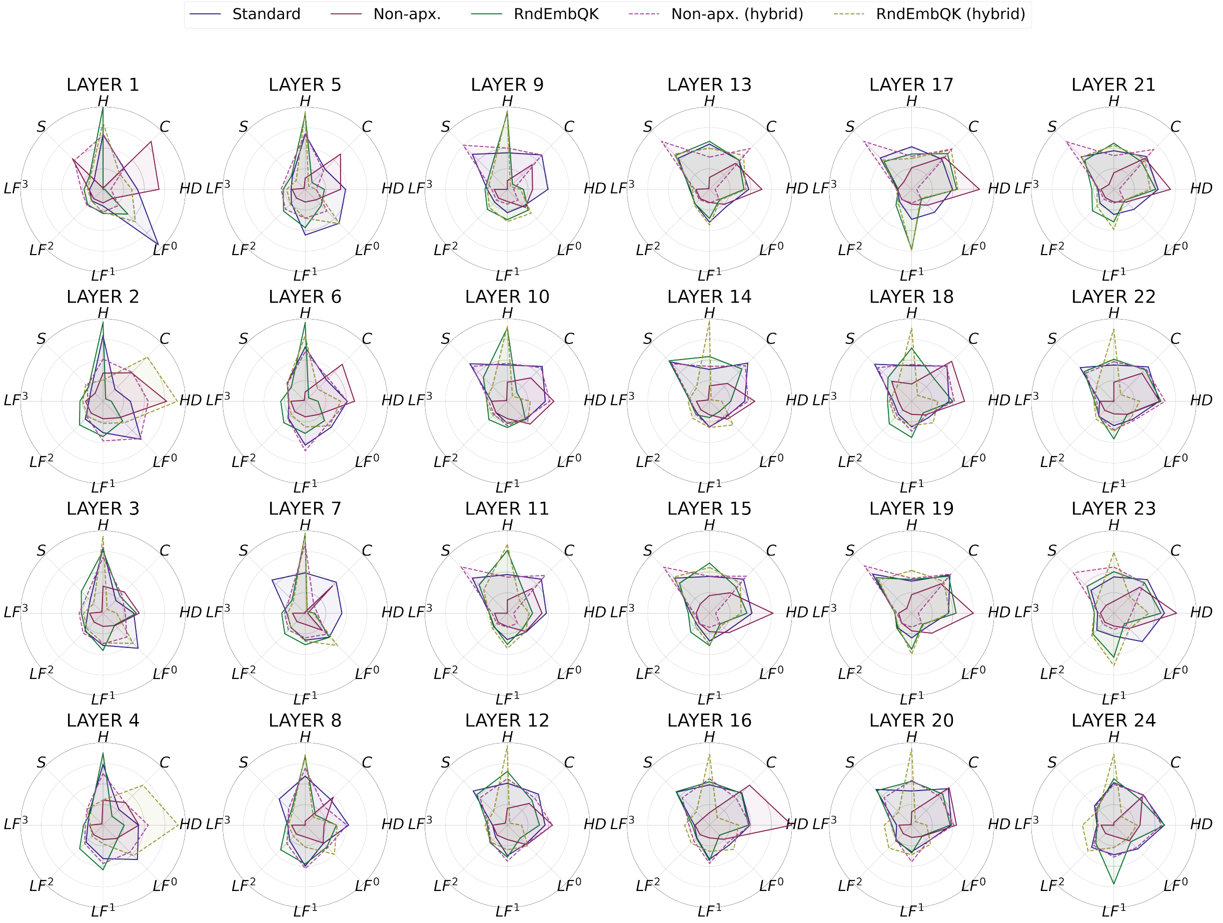


Figure 8: Visualization of attention matrix characteristics across different layers for *Non-approximate* and *RndEmbQK*, and their hybrid variants, compared to *Standard* (H: ENTROPY, C: CONC, HD: HEADDIV, LF: LOCFOCN, S: SINK).