MODULATED TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Softmax based attention suffer from several challenges like attention sink and attention noise. A well known reason for which is that all attention scores need to sum to one. Hence, a natural remedy would be to explore alternatives to softmax. A recent advancement that replaces softmax is the DIFF attention which relies on correcting a base information carrying softmax attention map by subtracting another sequence dependent softmax attention map instead of per token-position dependent gate as in gated attention. We introduce MODULATED TRANSFORMER that computes a sequence dependent correcting mechanism by modulating the amplitude of a base softmax attention map signal by taking hadamard product with the second correcting attention map. Modulation of a base softmax signal by a suitable modulating function results in sharper and sparser final attention patterns. At 780m scale, Modulated Transformer surpasses softmax based Transformer in language modeling and achieves significant model size and token efficiency as indicated by a lower perplexity score with only 58% of the model size of 1.3B Vanilla Transformer with similar token budget. It also outperforms softmax based Transformer by 3.15 average accuracy points and by 3.24 points after pruning attention weights and is competitive with next strongest baseline DIFF on short-context language modeling tasks. With these results, the Modulated Transformer advances the abilities of existing language models and opens up a new avenue for research on developing new expressive and stable modulating functions.

1 INTRODUCTION

Attention based deep architectures excel at modeling semantic structure of sequences and are successfully applied to both vision and language tasks (Bahdanau et al. (2016); Dosovitskiy et al. (2021); Vaswani et al. (2023)). Attention is the workhorse of modern day Transformer based language models which almost always involve computing softmax attention scores to compute representations of the input sequence via weighing values (information carrying signal) in a sequence dependent way. Several variants of softmax-based attention mechanisms have been proposed, see Section 4. Veličković et al. (2025) point out that the predictive power of softmax-based architectures arises from "circuits" that perform some computation and generalise to diverse inputs. However, softmax-based attention for routing and mixing information may not be optimal and may lead to commonly observed pitfalls like attention sink (Xiao et al. (2024); Barbero et al. (2025); Gu et al. (2025)), attention noise (Ye et al., 2025) and rank collapse (Saada et al., 2025).

We therefore ask if the vanilla softmax-based attention is the most optimal way of utilising the parameter and token budget to learn meaningful representations of the input. We find that re-routing the information processing by modulating the amplitude of the softmax signal by another sequence dependent correcting signal leads to significant gains from the vanilla attention, see Table 2. A notable classification of these attention mechanisms is that of softmax-based and non-softmax-based mechanisms. In this work, we introduce Modulated Transformer with a new attention module - Modulated Attention - an alternative to softmax-based attention, which reweighs the base softmax map of one half of the head dimension by a sequence-dependent sigmoid gate computed from the other half making it sharper and sparser; the headwise RMSNorm on head outputs that follows makes the attention scale-free except when the head's output is small.

Motivation for Modulated Transformer. Compared to RNNs, Transformers are more suitable for stochastic gradient descent based training of deep networks and are parallelisable. We note that a correction mechanism on top of a base computation or representation is often easier to learn

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.

via gradient descent, for instance in Residual networks (He et al. (2015); Vaswani et al. (2023); Srivastava et al. (2015)) learning one monolithic function via neural networks is much harder than to learn $x + \Delta_\theta x$ where x is the input to any given layer of a residual network. Notably, Differential attention (Ye et al., 2025) is a major improvement over Vanilla attention while keeping the rest of the architecture the same that uses an additive correction. In this work, we explore the multiplicative correction instead, in spirit of RMSNorm which normalises and then rescales its input, which we find to be at par with additive correction. A product can zero out an entry (useful for sink removal) while a difference can make it negative, which is harder to interpret.

We evaluate our proposed architecture on several language modeling datasets with perplexity being the first step of analysis. We include Llama style (Touvron et al., 2023) Differential Transformer and Vanilla Transformer as our baselines while noting that Differential Transformer is a very strong baseline. We find that under the same architecture(except normalisation layers), parameter and token budget our method surpasses Vanilla Transformer and is competitive with Differential Transformer, therefore ascertaining that the vanilla softmax attention is not the most optimal series of computation to extract better performance with minimal changes to the experimental setup. However, our method and differential transformer introduce more computation during forward and backward passes hence are slightly slower to train, which is rectified by the fact that both approaches achieve lower perplexity at much lower parameter and token count. We note that due to high variance of attention weights from being unnormalised, normalisation layers help Differential attention and our method to surpass Vanilla attention. The QK-norm ablations, Section 3.5, show that Modulated attention models sequences better than Vanilla attention based on perplexity analysis with and without QK-norm.

2 Modulated Transformer

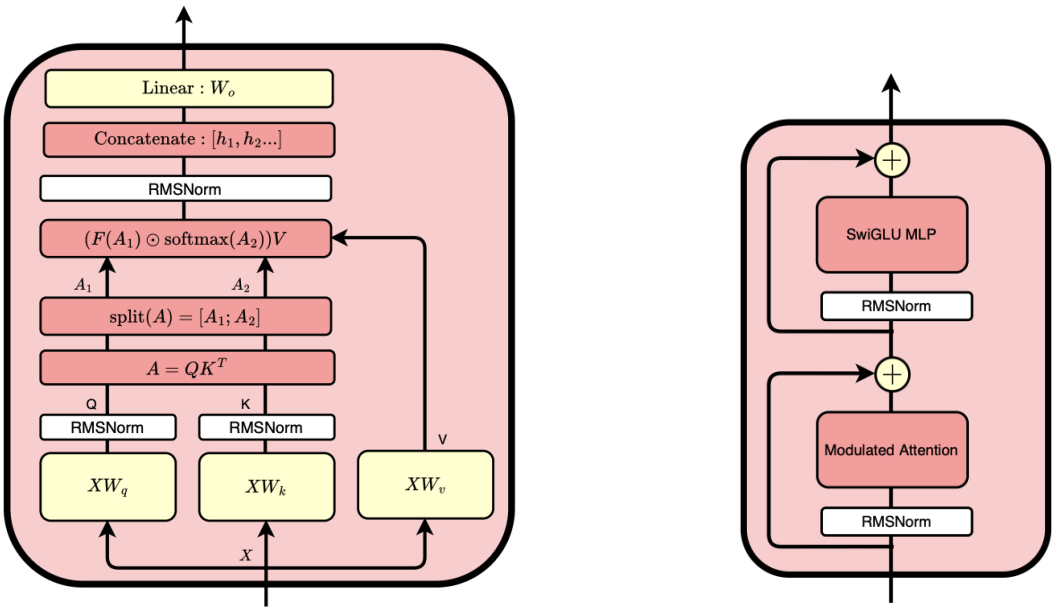

Figure 1: (**Left**) Illustration of Modulated Attention block and (**Right**) Overall architecture. Input $X \in \mathbb{R}^{N \times d_{model}}$ is projected into the $Q, K$and V. A_1 and A_2 are two half splits of the score tensor A across the head dimension which are fed into F and softmax respectively. Where F is the modulating function modulating the amplitude of softmax(A_2) by taking hadamard product with $F(A_1)$. After the attention block the output is fed into SwiGLU MLP.

We follow most architectural choices as in Differential Transformer(Ye et al., 2025), but at a smaller scale, with an additional QK-norm (Dehghani et al. (2023);Wortsman et al. (2023)) and *modulated attention* instead of *diff attention*. We call the architecture that uses Vanilla multi-head attention as Vanilla Transformer. We adopt improvements commonly used with Vanilla Transformer like RMSNorm (Zhang & Sennrich, 2019), SwiGLU (Shazeer, 2020). As in Differential Transformer, we

stay close to Vanilla attention, in order to achieve a sequence dependent correcting mechanism both Modulated and Differential Transformer split the pre-softmax attention logit tensor A into $[A_1; A_2]$ and then process them separately. Differential attention computes difference of two softmax operations $\text{softmax}(A_1) - \lambda\text{softmax}(A_2)$, while Modulated attention computes the $F(A_1) \odot \text{softmax}(A_2)$, see Figure 1. Details about the choice of F is given below.

2.1 Modulated Attention

$$
\begin{aligned}
Q = XW_q, K = XW_k, V = XW_v \\
[A_1; A_2] = A = \frac{QK^T}{\sqrt{d}} \\
\overline{\text{heads}} = [\text{head}_1, \text{head}_2 ... \text{head}_h] = (F(A_1) \odot \text{softmax}(A_2))V \\
\text{ModulatedAttn}(X) = \text{Concat}[\text{RMSNorm}(\overline{\text{heads}})]W_O
\end{aligned}
\tag{1}
$$

where, up to appropriate reshaping, $X \in \mathbb{R}^{N \times d_{model}}$; $Q, K \in \mathbb{R}^{2h \times N \times d}$; $V, \overline{\text{heads}} \in \mathbb{R}^{h \times N \times 2d}$, N is the sequence length, h is the number of heads which is half that of Vanilla Attention, $d = \frac{d_{model}}{2h}$, $A \in \mathbb{R}^{2h \times N \times N}$ and $A_1, A_2 \in \mathbb{R}^{h \times N \times N}$. Concat converts $\overline{\text{heads}}$, a $\mathbb{R}^{h \times N \times 2d}$ tensor to $\mathbb{R}^{N \times d_{model}}$. There are two ways to split the QK or A tensors to obtain A_1 and A_2, see 3.5 for Method 1 and Method 2. We use the second split method by default i.e. splitting A into $[A_1; A_2]$ equally across the head dimension. Note that we also additionally employ QK-Norm on Q and K to be competitive with Differential Transformer.

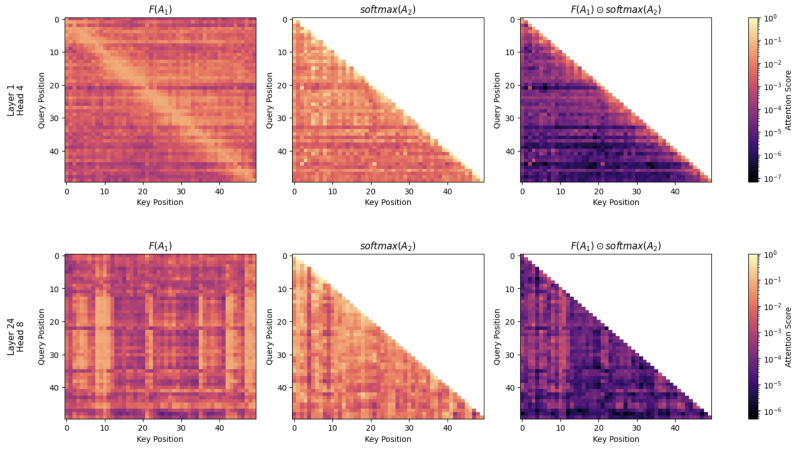

Figure 2: Visualization of modulating function $F(A_1)$, base signal $\text{softmax}(A_2)$ and final learned attention $F(A_1) \odot \text{softmax}(A_2)$ matrices for first and last layer of Modulated Transformer for first 50 tokens of a sample from C4. Overall magnitude of $F(A_1)$ and consequently $F(A_1) \odot \text{softmax}(A_2)$ is small and it leads to sparser and sharper attention scores for weighing values vectors.

A key component of the Modulated Attention is the modulating function given by $F(A_1)$(interchangeably referred as just F). We empirically find the below form to be expressive, stable and cheap:

$$
F(A_1) = \frac{(\exp(\lambda))^2}{\exp(\lambda) + \exp(A_1)} = \exp(\lambda) \cdot \sigma(\lambda - A_1) \tag{2}
$$

where $\lambda = \lambda_q \cdot \lambda_k$ is a scalar computed by taking the dot product of two learned vector parameters λ_q, $\lambda_k \in \mathbb{R}^d$ per attention module for all layers. This form of parameterization is taken from Differential Transformer (Ye et al., 2025) to synchronize training dynamics with other weights.

Bounded Gate. Note that after the hadamard product of $F(A_1)$ and $\text{softmax}(A_2)$ is computed the final attention scores to weigh value vectors are no longer normalised this is by design and can be achieved by choosing $F(A_1) = \exp(A_1)$, where A_1 is added to the exponents of the numerator $\frac{\exp(A_2)}{Z}$ i.e. $\text{softmax}(A_2)$ elementwise. Intuitively, this functions as a correcting mechanism

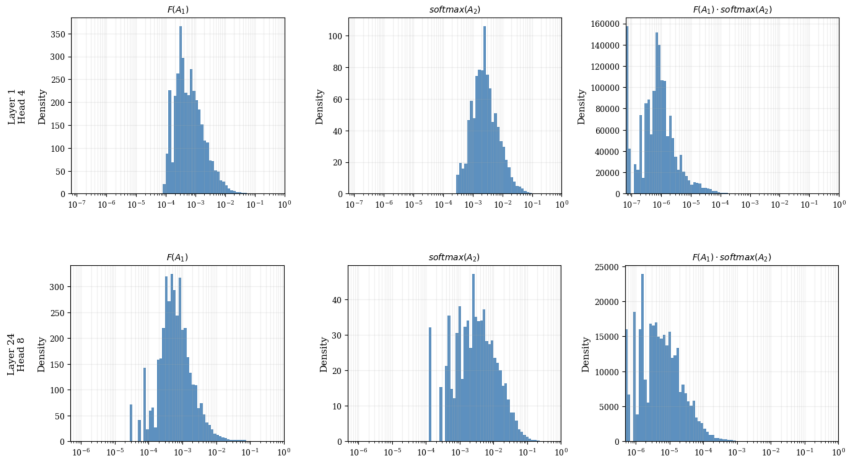

Figure 3: Histogram of amplitude modulating function, $F(A_1)$, softmax(A_2) and modulated signal $F(A_1) \odot$ softmax(A_2) depicting sparsity of resulting attention scores for first and last layer.

in a sequence dependent way by amplifying or attenuating the softmax(A_2) scores. This however is unstable as exponential very quickly become very large for typically values of $\frac{QK^T}{\sqrt{d}}$ scores. The above form of F (eq. 2), essentially bounds the output of $F(A_1)$; for very small values of $\exp(A_1)$, $F(A_1)$ approaches $\exp(\lambda)$ and for large values of $\exp(A_1)$, $F(A_1)$ is close to zero, thus making the output well-behaved. Alternatively, another form of the same modulating function is $F(A) = u \cdot \sigma(-A + \log(u))$, where u is learned. To enforce it to be positive we set u as $\exp(\lambda)$. We empirically found $\exp(\lambda)$ does not grow too large(at most 0.2713 for 780m Modulated Transformer across layers), potentially due to small initialization, thus enabling stable training.

Scale-Invariance and Role of ε. The RMSNorm applied head wise to head outputs after the modulated attention scores and before projection by W_o makes the whole computation invariant to scaling by u and our work then closely resembles Gated attention (Bondarenko et al., 2023) but with sequence-dependent modulation with sigmoid. Due to the scaling symmetry of value headwise RMSNorm any unnormalised attention will in theory be scale-free, effectively attention may or may not be normalised without effecting the output of attention module. The small ε for stability in the denominator of RMSNorm breaks this scaling symmetry, particularly when head output is small and numerically the scaling u contributes to better language modeling. Appendix A.3 examines this.

Modulated attention and Full architecture is shown in Figure 1. An implementation of Modulating attention is given below where the notation of h, d is as above.

```
def F(A, lambda_):
    mod = exp(lambda_) * sigmoid(-A + log(exp(lambda_))
          #exp(lambda_)^2/(exp(lambda_) + exp(A))
    return mod

def ModulatedAttn(X, W_q, W_k, W_v, W_o, lambda_):
    seq_len, d_model = X.size()
    q = W_q(X).view(seq_len, 2 * h, d)
    k = W_k(X).view(seq_len, 2 * h, d)
    v = W_v(X).view(seq_len, h, 2 * d)
    q, k = RMSNorm(q).transpose(0, 1), RMSNorm(k).transpose(0, 1)
    v = v.transpose(0, 1)
    attn_weights = q @ k.transpose(-1, -2)/sqrt(d)
    attn_weights = attn_weights.view(h, 2, seq_len, seq_len)
    A_1, A_2 = attn_weights[:, 1], attn_weights[:, 0]
    attn = Hadamard(F(A_1, lambda_), softmax(A_2, dim=-1)) @ v
    attn = RMSNorm(attn).tranpose(0, 1).reshape(seq_len, h * 2 * d)
    attn = W_o(attn)
    return attn
```

2.2 Full Architecture

The proposed Full Architecture is similar to Differential Transformer with a headwise RMSNorm and QK-Norm being new components from Vanilla Transformer. Each layer of the Modulated Transformer contains a multi-head modulated attention module followed by SwiGLU (Shazeer, 2020) feed-forward network operation. Below is the description of a single layer l.

$$Y_l = \text{ModulatedAttn}\left(\text{RMSNorm}(X_l)\right) + X_l, \tag{3}$$

$$X_{l+1} = \text{FFN}_{\text{SwiGLU}}\left(\text{RMSNorm}(Y_l)\right) + Y_l. \tag{4}$$

where

$$\text{FFN}_{\text{SwiGLU}}(X) = \left(\text{swish}(XW_G) \odot XW_1\right) W_2$$

$$\text{swish}(x) = x\sigma(x)$$

$W_G, W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{\text{model}}}$, and $d_{ff} = \frac{8}{3} d_{\text{model}}$

3 Experiments and Analysis

This section compares our proposed method against normalised vanilla attention. We carry out experiments that assess language modelling capability of Modulated Transformer. We also study the final modulated attention patterns learned by Modulated Transformer, show that it is attention sink free and ablate on role of QK-norm and two realisations of splitting methods.

3.1 Experimental setup

Baselines. Modulated attention introduces unnormalised attention scores and induces sparser and sharper attention patterns, see Figure 2, via a learned sequence dependent correction mechanism. This is done by splitting the computed QK or A tensor appropriately to compute two sets of attention maps which are processed as discussed in Section 2. This approach enables better performance. To assess this performance gain we include a very strong baseline of Differential Transformer (Ye et al., 2025) with Differential attention, which also splits QK or A tensor, and Vanilla Transformer (Vaswani et al. (2023); Touvron et al. (2023)) with Vanilla attention.

Architecture Details We pretrain Llama style architectures (Touvron et al., 2023) on C4 dataset at three model sizes namely small, medium and large corresponding to 130m, 360m and 780m. These model sizes have hidden dimension of 768, 1024 and 1536 with 12, 24 and 24 layers respectively. The value head dimension is 128, 128, 96, respectively for Modulated and Differential Transformer corresponding to 6, 8 and 16 heads and 64, 64, 48 for Vanilla Transformer corresponding to 12, 16, 32 heads. Following Differential attention, the number of heads for our method is half that of number of heads for Vanilla attention to keep the total number of $N \times N$ attention maps constant. As shown in the implementation of our method above, the query and key head dimension d (notation as in Section 2.1) is same as that of Vanilla attention i.e. 64, 64, 48 for the three model sizes.

Pretraining Details. We train all our models on C4 dataset (Raffel et al., 2023). To develop Modulated Transformer we follow Differential Transformer (Ye et al., 2025) and we follow GaLore (Zhao et al., 2024) for the training setup. We train the three model sizes across all baselines on 3B, 7.5B and 12B tokens respectively with context length as 512. All models are trained with AdamW(Loshchilov & Hutter, 2019) with BF16 format. We use cosine learning rate scheduler with warmup. Number of warmup steps is 10% of the total number of iterations which is 20k, 50k, 80k respectively for the three model sizes. We tune peak learning from a set of $\{0.0025, 0.003, 0.0035 \ldots 0.0075\}$ starting from 0.0025 up to 0.0075 at step sizes of 0.0005, picking the best performing learning rate for 130m and 360m model sizes while for 780m we start from the best learning rate of 360m and tune around it for all baselines. We find that Modulated Transformer can tolerate much larger learning rates. See Appendix A

3.2 Language Modeling Tasks

In this section we discuss experiments aimed at assessing and comparing language modelling capability of Vanilla Transformer, Differential Transformer and Modulated Transformer on three model

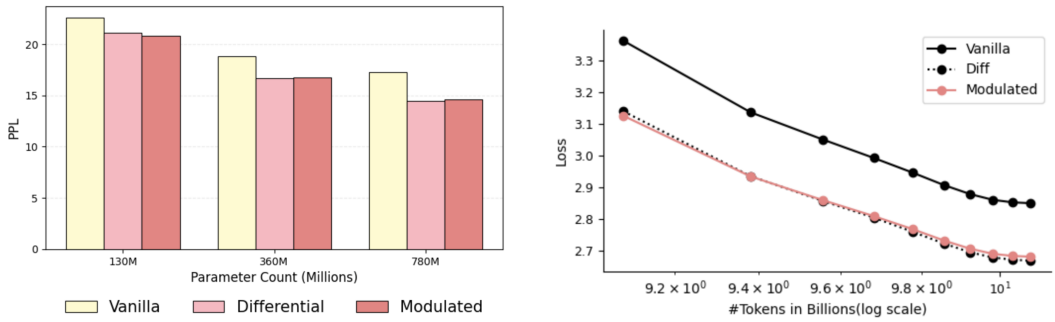

Figure 4: (**Left**) C4 eval. PPL(lower is better) across model sizes for our method(Modulated) and all other baselines. (**Right**) Comparison of eval. loss scaling of 780m scale with the number of C4 training tokens, up to 12B. Notably, the Diff and Modulated curves are very close, implying multiplicative correction is at par with additive correction.

Table 1: Comparison across model scales on seven evaluation metrics using LM eval harness (Gao et al., 2024). "acc" represents zero-shot accuracy and "acc-n" represents length normalised zero-shot accuracy. (Pruned) means final attention weights below a threshold were set to zero at inference. Bold numbers in $(.)$ in the last column indicate performance improvement with respect to Vanilla Transformer. Higher is better for all metrics.

Model Size	**Method**	**ARC-C** acc-n ↑	**ARC-E** acc ↑	**BoolQ** acc ↑	**HellaSwag** acc-n ↑	**OBQA** acc-n ↑	**PIQA** acc ↑	**SciQ** acc ↑	**Avg.**
Small(130M)	Vanilla	22.53	40.74	55.87	29.90	27.00	61.70	69.60	43.91
	Differential	22.61	41.04	62.14	30.91	26.40	62.40	69.80	45.04(**+1.13**)
	Modulated	23.63	41.25	61.35	31.08	25.80	62.51	72.20	45.40(**+1.49**)
Medium(360M)	Vanilla	23.98	44.19	61.59	33.29	28.20	65.29	72.50	47.01
	Differential	23.63	45.83	60.73	37.92	29.80	65.94	74.00	48.26(**+1.25**)
	Modulated	24.49	46.00	57.98	36.74	29.60	66.10	77.60	48.36(**+1.35**)
Large(780M)	Vanilla	24.40	44.49	56.15	35.45	29.60	66.43	73.50	47.15
	Differential	25.85	48.70	60.28	43.16	30.00	68.28	78.00	50.61(**+3.46**)
	Modulated	25.68	47.98	59.82	41.72	28.80	68.50	79.60	50.30(**+3.15**)
	Modulated(Pruned)	25.68	48.06	60.00	41.86	29.00	68.55	79.60	50.39(**+3.24**)

sizes as discussed in the previous subsection, while keeping the overall architecture fixed across baselines and datasets. We evaluate all the pretrained models on 7 language modeling benchmarks.

We use ARC-easy, ARC-challenge (Clark et al., 2018), BoolQA (Clark et al., 2019a), HellaSwag (Zellers et al., 2019), OpenbookQA (Mihaylov et al., 2018), PiQA (Bisk et al., 2019) and SciQA (Auer et al., 2023) for language modeling evaluation. We report length-normalized accuracy for HellaSwag, ARC-challenge, and OpenbookQA, and accuracy for all other tasks. As shown in Table 1, Modulated Transformer achieves better performance than Vanilla Transformer and is competitive with Differential Transformer.

We observe that 778M parameter Modulated Transformer outperforms 1339M parameter Vanilla Transformer with lesser number of tokens based on perplexity analysis. As in Table 2 we only need 58% of the parameters trained on similar number of tokens to get better perplexity score than Vanilla Transformer under similar experimental setup.

Table 2: PPL(lower is better) comparison of Llama style Vanilla Transformer and Modulated Transformer with roughly half the number of Parameters. Baseline number, under similar experimental setup, taken from GaLore(Zhao et al., 2024).

Model	**Dataset**	**Hidden size**	**# Parameters**	**# Heads**	**# Tokens**	**Perplexity** ↓
Vanilla	C4	2048	1339M	32	13.1B	15.56
Modulated	C4	1536	778M	16	12B	**14.60**

3.3 Sparse Attention Patterns

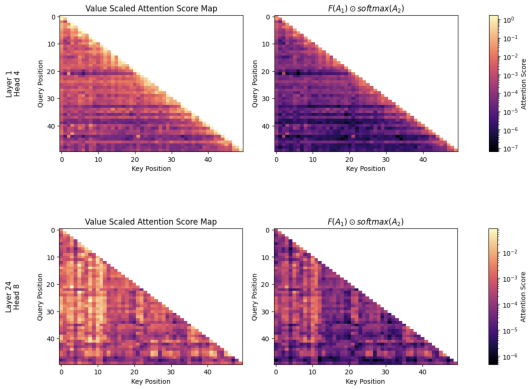

Figure 5: Value norm scaled attention map.

Figure 3 shows that the majority of learned attention scores are below 10^{-4} for this particular histogram. Therefore, we conducted evaluation by pruning the final attention scores of pre-trained Modulated Transformer at 780m scale(Table 4) by setting all scores below a threshold to 0.0 during inference. This improves or maintains the accuracy on language modeling tasks across all datasets with an 0.09 average improvement for a fixed seed to exclude gains due to randomness, indicating some attention weights of all the unnormalised weights are very small noise that can be pruned. Additionally, following Kobayashi et al. (2020), we scaled the final attention maps by L^2 norm of value vectors, i.e $\text{ScaledAttn}_{ij} = w_{ij} \cdot ||V_j||_2$ where $w = F(A_1) \odot \text{softmax}(A_2)$, to gauge if the overall information propagated is effectively attenuated by very small attention scores or is compensated by respective value vectors. We find that all layers capture both high and low signal amplitudes across the value-scaled attention maps and heads, thus even with a skewed histogram the final attention scores rather contribute to superior performance of Modulated attention. A sharper signal with both sparsity and high values is shown in Figure 5. See Appendix B.

3.4 Attention Sink Analysis

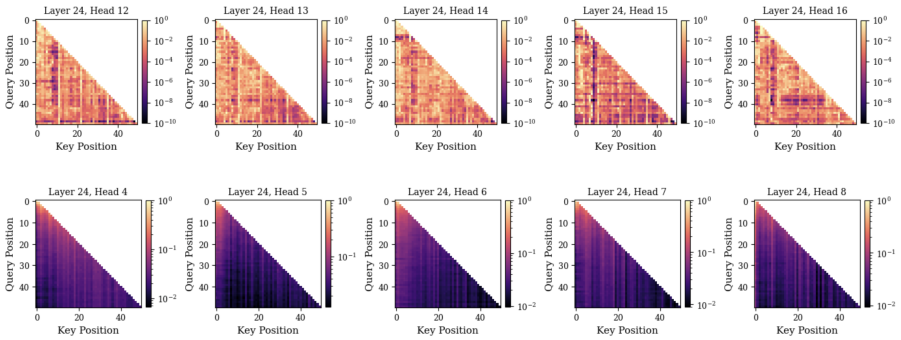

Figure 6: **(Top)** Visualization of attention map of Vanilla Transformer, the attention pattern assigns large weightage to first few tokens. **(Bottom)** Visualization of the final attention map i.e. $F(A_1) \odot \text{softmax}(A_2)$ of Modulated Transformer, the patterns are normalised row-wise. The normalised attention patterns are sparse, lower in magnitude and attention sink free.

The mixing of information is governed by the attention patterns of the model (Barbero et al., 2025), based on how the information is processed in attention heads or layers. A common phenomenon normalised attention patterns exhibit is that of attention sink (Xiao et al. (2024);Gu et al. (2025)) where the attention score assigned to the first few tokens of a sequence is consistently high even when it carries no semantic information. When the attention scores are unnormalised the proportion of attention score in a row to first few tokens could still be large as can also be seen in Figure 1 of Ye et al. (2025). Figure 6 shows the row-normalised attention map of Modulated attention is relatively uniform(with small magnitude overall and sparse) with accumulation towards top left and at sparse key positions(vertical sharp columns) in the attention map due to shorter sequences and learned re-routing of information respectively. While Vanilla attention shows signs of attention sink.

3.5 Ablation Study

In this section we analyse the components that contribute to the performance of the Modulated Transformer. First, we study the effect of QK-norm on Vanilla and our method, see Table 3. With-

out QK-norm it still beats Vanilla Transformer. With QK-norm it becomes competitive with the Differential Transformer as indicted by Figure 4 and Table 1. Second, we study the effect of representational capacity of A_1 and A_2 on performance, where $[A_1; A_2] = split(X, W_q, W_k)$. See Figure 7 for two realisations of split operation. We can either split the QK tensor across the last dimension(Method 1) or split the A tensor equally across the head dimension(Method 2). In effect the two splitting methods allow an unequal or equal capacity split respectively. An equal split via the Method 1 gives slightly lower perplexity empirically, possibly due to coupling of $2d$ QK-head dimensions before splitting as a result of normalisation from RMSNorm, see Appendix A.3. The representational capacity of A_1 and A_2 is governed by the dimension d_{A_1} of the XW_{A_1} where $XW_{A_1} \in \mathbb{R}^{N \times d_{A_1}}$, similarly for A_2 and for both Query and Key. Note that $d_{A_1} + d_{A_2} = \frac{d_{model}}{\#\text{heads}}$.

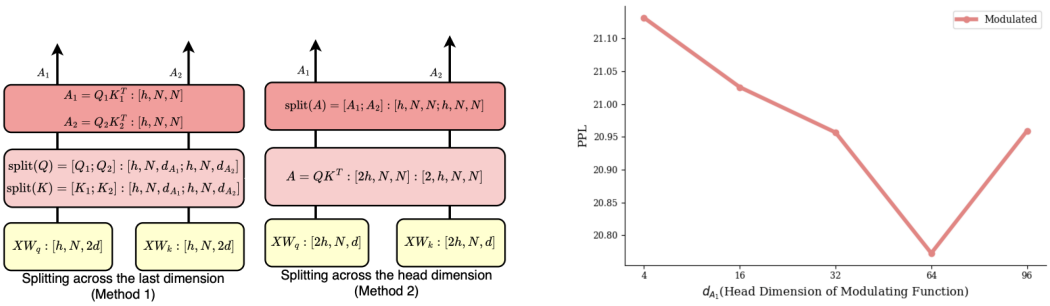

Figure 7: (**Left**) Two realisations of splitting the QK or A tensor. Method 1 gives a variant with adjustable capacity of our method, h, N and d notation as in Section 2.1. (**Right**) Capacity Ablation for 130m model size with $d_{A_1} + d_{A_2} = 128$, where capacity is given by dimensions d_{A_1} and d_{A_2} for the last dimension splitting method(Method 1). A_1 is used for computing the modulating function and A_2 is used for computing the base softmax. We note that PPL(lower is better) is lowest at $d_{A_1} = 64$. Hence, equal split is chosen by default for all other runs(Method 2).

Table 3: Ablations for QK-norm (Wortsman et al., 2023) at 130m scale.

Model	QK-Norm	#Heads	PPL ↓
Vanilla	×	12	22.56
Vanilla	✓	12	22.09
Modulated	×	6	21.46
Modulated	✓	6	**20.87**

Table 4: Performance for different pruning thresholds for 780m Modulated Transformer.

Threshold	Avg Performance
-	50.30
10^{-5}	**50.39**
10^{-4}	50.36
10^{-3}	47.24

4 Related Work

There has been a lot of progress since the Vanilla attention (Vaswani et al., 2023), focusing on sparsity, improving time complexity and modeling long-context. Recent works can be broadly classified into softmax and non-softmax based attention approaches. Motivation for looking for alternatives of softmax or normalised attention can be found in Veličković et al. (2025), Huang et al. (2026), Lu et al. (2026). Our work is most closely related to Differential Transformer(Ye et al., 2025) and Gated Attention (Bondarenko et al., 2023). To our knowledge, the idea of sequence dependent modulation of post-softmax attention scores by a suitable modulating function is new given that there could be many potential modulating functions, but with the given sigmoid form it is also comparable to Linear Attention(Gu & Dao (2024); Yang et al. (2024); Sun et al. (2023)). The Hadamard product for modulation also has a loose analogue with the $(L \odot QK^T)V$ form of Mamba-2(Dao & Gu (2024)), although L there is a fixed decay mask.

Non-Softmax based attention. One of the earliest developments in an alternative to softmax is that of sparsemax(Martins & Astudillo, 2016). Sparsemax can produce sparse patterns like our method. ReLA (Zhang et al., 2021) uses ReLU instead of softmax and argues fixed sparsity patterns lack flexibility and generalization and also produces data-dependent sparse patterns and this sparsity

emerges during training and the scores not restricted to a probability distribution very much like our method. Sigmoid self-attention (Ramapuram et al., 2025) introduces a bias term in the exponential while applying sigmoid elementwise to the attention logits, which is analogous to $\log(\exp(\lambda))$ in our formulation in Section 2 but we apply it to one half of the equal split of the A tensor to serve as a correcting mechanism to softmax scores. Softpick (Zuhri et al., 2026) is another non-softmax based attention that learns sparse patterns and mitigates attention sink.

Gated Self-Attention. Qiu et al. (2025) presents a study of introducing gates to various positions in the Vanilla attention mechanism. Our method is related in that we modulate the post-softmax scores by a sigmoid modulating function to get sparse attention patterns that mitigate attention sink. Our work also has some similarities with Gated Attention (Bondarenko et al., 2023) which compute a token dependent gate applied to post-softmax, while we modulate post-softmax scores by a bounded sequence dependent gate. Similarly, The neural data router (Csordás et al., 2022) also aims to make Transformers capable of learning adaptive control flow with a combination of geometric attention and sigmoid gating mechanisms, we instead modulate the softmax amplitude with a sigmoid dependent on the entire sequence instead of per token-position.

Softmax based Self-Attention. LaS-attention (Zimerman & Wolf, 2024) computes exponential decay of attention logits, we on the other hand modulate the post softmax amplitudes by a stable F function, which originally draws inspiration from the exponential function which is relatively unstable as discussed in Section 2. FoX(Lin et al., 2025) is another work which is similar to our method, however it performs sequence level gating by product of quantities at token level while we modulate with sigmoid gate using another attention map. Selective Attention (Leviathan et al., 2025) performs operations on attention logits by only having an attenuating effect and learns sparse patterns. SSMax (Nakanishi, 2025) uses a softmax formulation and adds learned parameters in the exponent that amplifies the scores but is then renormalised, our method is similar in spirit that we aim to modulate the softmax signal by elementwise multiplication by an exponential of sequence dependent quantities with some additional bounding tricks via a careful design of modulating function.

5 Limitations and Conclusion

Limitations. This work shows significant improvements over relevant baselines, however, our method has some drawbacks. Our method is slower to pretrain compared with Vanilla and roughly the same as Differential, that is the wall-clock time for forward and backward passes is the largest for our method more so with the addition of QK-norm; this is compensated by the performance gains at relatively smaller scale of tokens and model size. We leave hardware-aware implementation of modulated attention for future work, as modulated attention is not directly compatible with FlashAttention (Dao et al., 2022). Due to limited compute we couldn't evaluate our method on long context training, tasks and datasets or test the scaling laws for our method.

Conclusion. In this work we propose Modulated Transformer with a new Modulated Attention module. Experiments on short-context language modeling show the superior performance of our method compared to normalised softmax based attention under significant parameter or token count reduction. The learned attention map(and value-norm scaled map) exhibit sparse and sharp patterns with which there are new possibilities of either simply pruning these patterns to enhance the downstream performance of Transformers with modulated attention with no additional tweak or directly compress the KV-Cache which can be further optimised due to modulation mitigating attention sink(Gu et al., 2025). We empirically found a working modulating function that can be used as is on small-130m to large-780m model sizes and achieve performance competitive to another highly effective method of Differential Transformers, additionally we also explored a second way to split the tensors, which sheds light on how much capacity the base and correcting signals need. Thus, in the spirit of developments in the Linear Attention space like Mamba(Gu & Dao, 2024), Mamba-2(Dao & Gu, 2024) and GatedDeltaNet(Yang et al., 2025), we aim to develop more expressive and stable Modulating Functions.

AI use statement

On the experimental side: We used AI to suggest possible bounded functions suitable for stable modulation since with exponential function training was unstable, following these suggestions we

ran several experiments involving sigmoid, tanh, SiLU activations. The final form of the modulating function with sequence dependence was discovered to perform at par with differential transformer only after analysing these failed experiments. The Value scaled attention map reference and code (Kobayashi et al., 2020) and some of the references for non-softmax based attention mechanisms was suggested by AI tools. However, literature survey that follow our work closely was done manually and validity of references given by AI was checked before using it for citing it in our work. We used AI to generate roughly 70 lines of Python code for the evaluation pipeline and AI helped to debug the throughput estimation code. The rest of the code was developed by us on top of code provided by GaLore(Zhao et al., 2024) and DIFF(Ye et al., 2025). The evaluation code was verified to be bug free and the entire evaluation pipeline was validated by reasonable figures at the scale we present in our work compared with relevant literature.

On the presentation side: We used AI tools for feedback on early drafts which helped us with some technical corrections as in the scale-free property due to RMSNorm on head outputs. We also used AI to write code to automate parsing and converting files to human readable format, like json to LaTeX. Hence, the LaTeX format of the main table(Table 1) was generated by AI tools. We also used AI to polish some of our figures(2, 3, 4, 5, 6). No part of the text was generated by AI, but we did act on suggestions from AI feedback on early drafts. AI mainly helped us with presentation of information as in polishing the text; excluding the overall analysis and assessment which was written, reviewed and refined without the help of AI and we take full ownership and responsibility of the overall methodology, findings and results.

Reproducibility statement

Our work is built on top of the code provided by Ye et al. (2025) and Zhao et al. (2024), both of which are public. We kept the experimental setup roughly the same as GaLore (pretraining on C4, warmup, etc). This codebase is widely used and modified, and can serve as a starting point for further work. We discussed tensor shaping and reshaping with clear notation to make understanding our method easier. We also gave pseudocode to implement modulated attention in Section 2.1 using these tensor reshaping. All hyperparameters and experimental details are described in Appendix A and in Section 3 and can be used for stable training and reproducing the results of our paper.

References

Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, and et al. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports, 13 (1):7240*, 2023.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. URL https://arxiv.org/abs/1409.0473.

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token?, 2025. URL https://arxiv.org/abs/2504.02732.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing, 2023. URL https://arxiv.org/abs/2306.12929.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019a. URL https://arxiv.org/abs/1905.10044.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes (eds.), *Proceedings of the 2019 ACL Workshop BlackboxNLP:*

Analyzing and Interpreting Neural Networks for NLP, pp. 276–286, Florence, Italy, August 2019b. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828/.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. The neural data router: Adaptive control flow in transformers improves systematic generalization, 2022. URL https://arxiv.org/abs/2110.07732.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024. URL https://arxiv.org/abs/2405.21060.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL https://arxiv.org/abs/2302.05442.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

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, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view, 2025. URL https://arxiv.org/abs/2410.10781.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Xingyue Huang, Xueying Ding, Mingxuan Ju, Yozen Liu, Neil Shah, and Tong Zhao. Threshold differential attention for sink-free, ultra-sparse, and non-dispersive language modeling. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18073–18092, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.824. URL https://aclanthology.org/2026.acl-long.824/.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms, 2020. URL https://arxiv.org/abs/2004.10102.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Selective attention improves transformer, 2025. URL https://arxiv.org/abs/2410.02703.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate, 2025. URL https://arxiv.org/abs/2503.02130.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jiecheng Lu, Xu Han, Yan Sun, Viresh Pati, Yubin Kim, Siddhartha Somani, and Shihao Yang. Zeros: Zero-sum linear attention for efficient transformers, 2026. URL https://arxiv.org/abs/2602.05230.

André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification, 2016. URL https://arxiv.org/abs/1602.02068.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018. URL https://arxiv.org/abs/1809.02789.

Ken M. Nakanishi. Scalable-softmax is superior for attention, 2025. URL https://arxiv.org/abs/2501.19399.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free, 2025. URL https://arxiv.org/abs/2505.06708.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Jason Ramapuram, Federico Danieli, Eeshan Gunesh Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, and Russ Webb. Theory, analysis, and best practices for sigmoid self-attention. In *ICLR*, 2025. URL https://arxiv.org/abs/2409.04431.

Thiziri Nait Saada, Alireza Naderi, and Jared Tanner. Mind the gap: a spectral analysis of rank collapse and signal propagation in attention layers, 2025. URL https://arxiv.org/abs/2410.07799.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks, 2015. URL https://arxiv.org/abs/1505.00387.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023. URL https://arxiv.org/abs/2307.08621.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. Softmax is not enough (for sharp size generalisation), 2025. URL https://arxiv.org/abs/2410.01104.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities, 2023. URL https://arxiv.org/abs/2309.14322.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training, 2024. URL https://arxiv.org/abs/2312.06635.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule, 2025. URL https://arxiv.org/abs/2412.06464.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer, 2025. URL https://arxiv.org/abs/2410.05258.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6507–6520, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.523. URL https://aclanthology.org/2021.emnlp-main.523/.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection, 2024. URL https://arxiv.org/abs/2403.03507.

Itamar Zimerman and Lior Wolf. Viewing transformers through the lens of long convolutions layers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62815–62831. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zimerman24b.html.

Zayd M. K. Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Softpick: No attention sink, no massive activations with rectified softmax, 2026. URL https://arxiv.org/abs/2504.20966.

A Model and Experimental details

A.1 Hyperparameters and Throughput

Table 5 lists different model sizes, architecture choices and number of respective training tokens with their tuned peak learning rates. Note that for tuning learning rate of 780m model size we start with the learning rate of 360m model size and tune around it.

Model(Model Size/Tokens)	n_{layers}	n_{heads}	d_{model}	Peak learning rate
Vanilla(130m/3B)	12	12	768	3.5×10^{-3}
Vanilla(360m/7.5B)	24	16	1024	2.5×10^{-3}
Vanilla(780m/12B)	24	32	1536	3×10^{-3}
Differential(130m/3B)	12	6	768	4×10^{-3}
Differential(360m/7.5B)	24	8	1024	3.5×10^{-3}
Differential(780m/12B)	24	16	1536	3.5×10^{-3}
Modulated(130m/3B)	12	6	768	5×10^{-3}
Modulated(360m/7.5B)	24	8	1024	6×10^{-3}
Modulated(780m/12B)	24	16	1536	7×10^{-3}

Table 5: Hyperparameters for all model sizes and baselines and Model Architecture details

Below is the forward throughput for our method and all baselines at 780m scale. Throughput is measured on 1 $\times$ NVIDIA RTX 6000 GPU.

Model	Batch Size	Length	Throughput(tokens/sec)
Vanilla	128	512	47,800
Differential	128	512	43,354
Modulated	128	512	43,425

Table 6: Inference throughput for 780m model size

A.2 Capacity Ablation Implementation Details

For ablation on capacity of the operations of $F(A_1)$ and softmax(A_2) where we split across the last dimension leading to unequal(or equal) partition into d_{A_1} and d_{A_2} dimensions used to compute A_1 and A_2. A sample code is provided below.

```
def ModulatedAttn(X, W_q, W_k, W_v, W_o, lambda_, Slice):
    seq_len, d_model = X.size()
    q = W_q(X).view(seq_len, h, 2 * d) # changed
    k = W_k(X).view(seq_len, h, 2 * d) # changed
    v = W_v(X).view(seq_len, h, 2 * d)
    q, k = RMSNorm(q).transpose(0, 1), RMSNorm(k).transpose(0, 1)
    v = v.transpose(0, 1)
    q_1, k_1 = q[:,:,-Slice:], k[:,:,-Slice:] # new parameter Slice
    q_2, k_2 = q[:,:,:-Slice], k[:,:,:-Slice] # new parameter Slice
    A_1 = q_1 @ k_1.transpose(-1, -2)/sqrt(d)
    A_2 = q_2 @ k_2.transpose(-1, -2)/sqrt(d)
    attn = Hadamard(F(A_1, lambda_), softmax(A_2, dim=-1)) @ v
    attn = RMSNorm(attn).transpose(0, 1).reshape(seq_len, h * 2 * d)
    attn = W_o(attn)
    return attn
```

A.3 Empirical Findings

Learned λ is small

As stated in the main text, Section 2.1, $\exp(\lambda)$ plays a crucial role in both stable and expressive modulation. We observe for 780m Modulated Transformer, after training, $\exp(\lambda)$ ranges from 0.0342 to 0.2713 across all layers. Hence, the maximum range of the Modulating function $F(A_1)$ is $[0, \exp(\lambda)]$ is $[0, 0.2713]$. This gives bounded and sparse attention patterns.
This is achieved because λ_q and λ_k are sampled i.i.d from $\mathcal{N}(0, \sigma^2)$ with $\sigma = 0.1$ as stated below.

$$\exp(\lambda) = \exp(\lambda_q \cdot \lambda_k)$$

$$\lambda_{q,i}, \lambda_{k,i} \sim \mathcal{N}(0, 0.01)$$

Let, $X = \lambda_q \cdot \lambda_k$, where $\lambda_q, \lambda_k \in \mathbb{R}^d$, then standard deviation of X is $0.01\sqrt{d}$, which is small for $d = 64, 64, 48$ for the three model sizes considered and $\exp(\lambda) \approx 1$ at initialisation. And the training drives this ceiling lower. Hence, with this initialisation scheme the training is stable.

Effect of scaling in RMSNorm on value heads

After weighing the value vectors from computed final attention scores we apply a RMSNorm layer across the $2d$ dimension of head output before concatenating and projecting by W_o. A formulation is given below:

$$\text{RMSNorm}(x) = \frac{x}{\sqrt{\frac{1}{2d}\sum_{i=1}^{2d} x_i^2 + \varepsilon}} \odot \gamma$$

This in principle should be agnostic to any additional positive scaling to x, i.e. $\text{RMSNorm}(x) = \text{RMSNorm}(\alpha \cdot x)$ where $\alpha > 0$ could be $(\exp(\lambda))^2$ and x are the weighted values per head considering a candidate modulating function as $\bar{F}(A_1) = \frac{1}{\exp(\lambda)+\exp(A_1)}$. We experimented by adding and removing this scaling term $(\exp(\lambda))^2$ in the modulating function and we also note this learned scaling is small from the previous section and that we chose $\varepsilon = 10^{-5}$.

$$\text{RMSNorm}((\exp(\lambda))^2 \cdot x) = \frac{x}{\sqrt{\frac{1}{2d}\sum_{i=1}^{2d} x_i^2 + \varepsilon \cdot (\exp(\lambda))^{-4}}} \odot \gamma$$

We empirically found that due the formulation of RMSNorm for numerical stability the scaling symmetry breaks and the $\varepsilon \cdot (\exp(\lambda))^{-4}$ term contributes to the effectiveness of the chosen modulating function $F(A_1) = \frac{(\exp(\lambda))^2}{\exp(\lambda)+\exp(A_1)}$, resulting in better perplexity scores and downstream performance, this can be attributed to a change in the effective learning rate for the gate and due to a learned "no-op head" mechanism (Bondarenko et al. (2023); Clark et al. (2019b)). This mechanism is learned from the contribution of ε term, since $(\exp(\lambda))^{-4}$ term is relatively large, as in for 780m Modulated Transformer $(\exp(\lambda))^{-4}$ can be as large as 7.3×10^5 effectively amplifying the effect of ε especially when the x signal or the head output is small i.e. mean square of the order of 10^{-4} or lower. In that regime the RMSNorm acts as a soft gate: a head can switch itself off for a token by attenuating all of its scores, this allows "no-op heads" to be learned. When $(\exp(\lambda))^2$ is removed from the modulating function i.e. $\bar{F}(A_1) = \frac{1}{\exp(\lambda)+\exp(A_1)}$ the "no-op head" behaviour of head wise RMSNorm is not as profound by what the model learns and we see a drop in performance(**21.43** PPL as compared to **20.87** PPL for the final formulation at 130m scale), this observation could lead to potentially new normalisation schemes where the mean square term in the denominator could have a learned bias scalar.

Coupled QK-norm Improves Perplexity

We present two methods of splitting QK or A tensors for computing the attention scores, see Section 3.5. An implementation decision we made to minimize performance degradation in switching between either methods, especially when the both methods are used for an equal split of capacity is that of applying QK-norm or RMSNorm. A major change for Method 1 is that instead of RMSNorm(q) where $q \in \mathbb{R}^d$ we apply RMSNorm(q) where $q \in \mathbb{R}^{2d}$, see the implementation above A.2. This implies when we split across the head dimension for Method 2 each d dimensional q or k vector

is normalised independent of other vectors, while for Method 1 when we split the $2d$ dimensional vector after RMSNorm, the first d_{A_1} dimensions are coupled with the last d_{A_2} dimensions due the shared normalisation factor. With this coupled QK-norm we achieve slightly lower perplexity of **20.77** and **14.53** as compared to 20.87 and 14.6 at 130m and 780m scales respectively. We however, pick Method 2 by default for the main results.

B Attention Patterns

Figure 8 and Figure 9 show value-scaled attention map and the learned final $(F(A_1) \odot \text{softmax}(A_2))$ attention map for the first 50 tokens of a sample from C4:

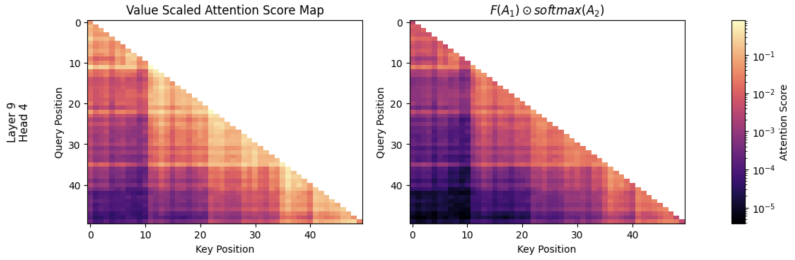

Figure 8: Value Scaled Attention Map for Layer 9

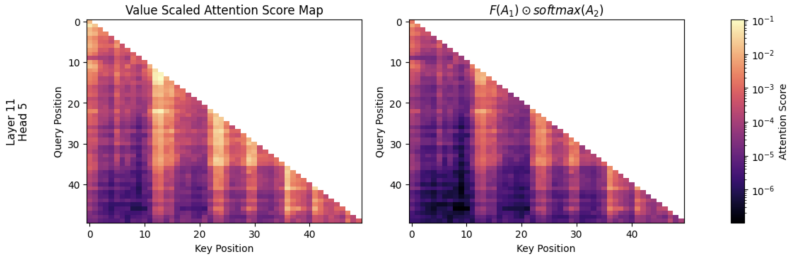

Figure 9: Value Scaled Attention Map for Layer 11