ZeroS: Zero-Sum Linear Attention for Efficient Transformers

Jiecheng Lu[1†], **Xu Han**[2], **Yan Sun**[1], **Viresh Pati**[1], **Yubin Kim**[1], **Siddhartha Somani**[1], **Shihao Yang**[1‡]
Georgia Institute of Technology[1], Amazon Web Services[2]
jlu414@gatech.edu[†], shihao.yang@isye.gatech.edu[‡]

Abstract

Linear attention methods offer Transformers $O(N)$ complexity but typically underperform standard softmax attention. We identify two fundamental limitations affecting these approaches: the restriction to convex combinations that only permits additive information blending, and uniform accumulated weight bias that dilutes attention in long contexts. We propose Zero-Sum Linear Attention (ZeroS), which addresses these limitations by removing the constant zero-order term $1/t$ and reweighting the remaining zero-sum softmax residuals. This modification creates mathematically stable weights, enabling both positive and negative values and allowing a single attention layer to perform contrastive operations. While maintaining $O(N)$ complexity, ZeroS theoretically expands the set of representable functions compared to convex combinations. Empirically, it matches or exceeds standard softmax attention across various sequence modeling benchmarks. The code implementation is available at this link.

1 Introduction

The Transformer architecture [1] has revolutionized sequence modeling across NLP, vision, speech, and reinforcement learning [2–7]. While its self-attention mechanism offers exceptional modeling flexibility, the quadratic $O(N^2)$ complexity in both time and memory with sequence length N limits its efficient implementation to long-context scenarios [8, 9]. Researchers have developed numerous linear-time attention mechanisms [8, 10–14] that preserve Transformer's strengths while scaling to longer sequences. Approaches include sparse attention patterns [15–17], kernel methods [8, 13, 14], low-rank approximation [18, 19], and efficient factorizations [20, 21]. Despite reducing from $O(N^2)$ to $O(N)$, these variants often underperform standard softmax attention, raising the question: Why do linear approximations save computation but sacrifice accuracy? Recent efforts to bridge this gap typically: 1) hybridize linear attention with local quadratic windows [22, 23], 2) learn softmax matrix low-rank projections [19, 24], or 3) sharpen linear kernels through normalization and gating [11, 12, 25]. While offer incremental gains, these approaches often compromise $O(N)$ efficiency, rely on task-specific hyperparameters, or introduce instabilities, limiting their practical use.

In this paper, we identify two fundamental limitations affecting linear and even softmax attention: 1) Bottleneck of convex combination [26–28]: softmax attention produces convex combinations of value vectors, with linear attention also aiming to achieve this primarily for numerical stability. However, these combinations can only blend information additively, unable to express subtractive or contrastive operations directly, forcing models to use multiple layers even for simple differencing tasks. 2) Uniform weight bias and attention dilution [11, 12, 29]: In long contexts, attention mechanisms incorporate a roughly uniform $\frac{1}{N}$ component in their weight expansion, introducing a persistent averaging effect that weakens focused attention and limits modeling of complex patterns. These limitations stem from the Taylor expansion $\exp(\boldsymbol{q} \cdot \boldsymbol{k}) = 1 + \langle q, k \rangle + \frac{1}{2}\langle q, k \rangle^2 + \ldots$, where the constant zero-order term enforces non-negativity for stability but creates an average-pooling bias that diminishes high-order token interactions. Rather than designing complex kernels to approximate

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

softmax while preserving the constant term, we propose a simpler solution: **remove it**. Subtracting the uniform component creates naturally zero-sum weights that permit both positive and negative values, enabling contrastive updates and sharper attention distributions while maintaining stability.

From this insight, we introduce ZeroS (Zero-Sum linear attention), achieving linear complexity while matching or exceeding quadratic softmax attention performance through three key elements: 1) Zero-order subtraction: removing the uniform $1/t$ term from each softmax row to create stable zero-sum weights; 2) Radial–angular decoupling: separating magnitude from direction by applying learned gates to first-order (linear) and higher-order (non-linear) softmax residuals, then reintroducing signed $\cos\theta$ terms to restore directional effects; 3) Linear-time implementation: using separable logits and gating for the reweighted zero-sum softmax, combined with linearizable angular computations via prefix sums, maintaining $O(Nd^2)$ runtime and $O(d^2)$ memory.

Our contributions include: 1) Identifying why the uniform zero-order softmax term limits attention mechanisms and demonstrating that its removal is safe and beneficial. 2) Developing Zero-Sum Linear Attention (ZeroS), a linear-time attention supporting negative weights with theoretical stability independent of sequence length. 3) Proving ZeroS offers greater expressivity than convex combinations while maintaining numerical stability. 4) Demonstrating that ZeroS matches or exceeds standard softmax attention on various benchmarks while maintaining linear time complexity.

2 Background

2.1 Preliminaries: Attention Mechanisms

We consider an input token sequence of length N, represented by the feature matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$, where each row $\boldsymbol{x}_t \in \mathbb{R}^{1\times d}$ is the embedding at time step t. With $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k$, $\mathbf{V} = \mathbf{X}\mathbf{W}_v$, an autoregressive (causal) single-head attention layer can be written in its matrix form as

$$\mathrm{Attn}(\mathbf{X}) = \sigma\big(\mathbf{M} \odot (\mathbf{Q}\mathbf{K}^\top)\big)\,\mathbf{V}\,\mathbf{W}_o, \quad \mathbf{X} \leftarrow \mathbf{X} + \mathrm{Attn}\big(\mathrm{LN}(\mathbf{X})\big),$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o \in \mathbb{R}^{d\times d}$ are learned projections, $\mathrm{LN}(\cdot)$ denotes layer normalization, and $\mathbf{M} \in \mathbb{R}^{N\times N}$ is the causal mask with $\mathbf{M}_{ij} = 1\{i \geq j\} - \infty \cdot 1\{i < j\}$, ensuring each position attends only to itself and the past. When σ is the row-wise softmax with a $1/\sqrt{d}$ factor, this represents standard self-attention with $O(N^2)$ complexity; replacing σ by the linearized kernels yields the linear attention variants that can be computed in $O(N)$ [8, 14]. Omitting the causal mask $\mathbf{M}$ reverts this to encoder-only attention, attending to all pairs of positions.

Recurrent Form Attention admits an equivalent step-by-step formulation. At time t, let $\boldsymbol{q}_t = \boldsymbol{x}_t\mathbf{W}_q, \quad \boldsymbol{k}_t = \boldsymbol{x}_t\mathbf{W}_k, \quad \boldsymbol{v}_t = \boldsymbol{x}_t\mathbf{W}_v$. Then the output $\boldsymbol{o}_t \in \mathbb{R}^{1\times d}$ is $\boldsymbol{o}_t = \frac{\sum_{i=1}^{t}\sigma(\boldsymbol{q}_t,\boldsymbol{k}_i)\,\boldsymbol{v}_i}{\sum_{i=1}^{t}\sigma(\boldsymbol{q}_t,\boldsymbol{k}_i)}$ where $\sigma(\boldsymbol{q},\boldsymbol{k}) = \exp(\boldsymbol{q}\,\boldsymbol{k}^\top/\sqrt{d})$ for vanilla attention. By choosing a kernel feature map $\phi(\cdot)$ such that $\sigma(\boldsymbol{q}_t,\boldsymbol{k}_i) = \phi(\boldsymbol{q}_t)\,\phi(\boldsymbol{k}_i)^\top$, the summations can be rearranged to maintain only the $d\times d$ hidden state $\sum_{i=1}^{t}\phi(\boldsymbol{k}_i)^\top\boldsymbol{v}_i$, avoiding the full $N\times N$ matrix $\mathbf{Q}\mathbf{K}^\top$. This yields the linear attention formulation: $\boldsymbol{o}_t = \frac{\phi(\boldsymbol{q}_t)\sum_{i=1}^{t}\phi(\boldsymbol{k}_i)^\top\boldsymbol{v}_i}{\phi(\boldsymbol{q}_t)\sum_{i=1}^{t}\phi(\boldsymbol{k}_i)^\top}$. Replacing the summation limit t with N converts this from the decoder-only autoregression into a encoder-only global recurrence, summing over all positions.

2.2 The intuition from existing linear attention research

We begin with insights from previous research on linear attention to introduce two key elements of our ZeroS structure: 1) radial-angular decoupling, and 2) zero-sum reweighted softmax. In softmax attention, each value vector $\boldsymbol{v}_i$ is assigned a weight $\frac{\exp(\boldsymbol{q}_t\boldsymbol{k}_i)}{\sum_{i=1}^{t}\exp(\boldsymbol{q}_t\boldsymbol{k}_i)}$, forming a convex combination that ensures numerical stability by keeping outputs within the convex hull of $\{\boldsymbol{v}_i\}$ [26, 27, 30]. Linear attention variants attempt to approximate this using weights of linearized kernel form $\frac{\phi(\boldsymbol{q}_t)\phi(\boldsymbol{k}_i)}{\sum_{i=1}^{t}\phi(\boldsymbol{q}_t)\phi(\boldsymbol{k}_i)}$ [8, 11, 14]. However, without constraining the sign of $\phi(\boldsymbol{q}_t)\phi(\boldsymbol{k}_i)$, this reduces to an affine combination that lacks the stability-ensuring bounds of convexity. While researchers have addressed this using non-negative feature maps like 1+ELU and ReLU [8, 12, 31, 32], these stability-ensuring modifications still underperform compared to standard softmax attention [8, 11, 14].

Coupling Interaction Between Radial and Angular Components In softmax attention, the core weight term $\exp(\|\boldsymbol{q}_t\|\|\boldsymbol{k}_i\|\cos\theta)$ is controlled by both vector magnitudes and their angle θ. Crucially,

when cosine flips from positive to negative, large positive values transform into very small ones, with step t and i highly coupled within the exponential. In contrast, linear attention applies nonlinear mappings $\phi(\cdot)$ to query and key [8, 12, 33], calculating $\|\phi(\boldsymbol{q}_t)\|\|\phi(\boldsymbol{k}_i)\|\cos\theta'$. Since these mappings yield only positive values, angles between vectors become restricted to less than 90 degrees, and the angular representation loses its flipping effect—cosine values merely serve as smooth gating signals between (0, 1). Previous research shows minimal performance changes when replacing softmax with sigmoid, ReLU, or similar functions [34–39], indicating that softmax attention's performance derives from modeling coupled angular and magnitude of (t, i) pairs rather than from the exponential property itself. Therefore, when constructing linear attention, we should reimplement these complex interactions rather than attempting to approximate softmax or merely mimicking an inner product.

Convexity of Sum-to-One Weights Under this perspective, we revisit the convex combination in softmax attention, which primarily serves numerical stability by preserving norm regardless of sequence length. As weights become more uniform, output norm expectation decreases at approximately $1/\sqrt{t}$ with sequence length t (assuming zero-mean vectors). However, these strictly positive weights mean input signals $\boldsymbol{v}_i$ can only contribute additively to outputs. In linear attention, without methods to suppress historical weights, this accumulation leads to attention dilution [11, 12, 29], where uniform signals increasingly dominate as sequence length grows. While some approaches address this using local windows or convolutional methods [9, 31, 40], these represent engineering solutions rather than resolving fundamental limitations of positive weights. Studies [27, 41–43] show that with softmax weights, a single attention layer cannot express differential or contrastive operations (even with just two tokens). The strictly positive convex combination inherently constrains ability to compress complex operations, limiting parameter efficiency. To enable more flexible parameterization with negative values, we must maintain numerical stability without relying on convex combinations' norm-preserving property while satisfying linear-time requirements.

Flexible Weighting in Related Works Implementing both the angular flipping effect and expressiveness requires numerically stable modeling of negative weights. Previous research [28, 44] demonstrated that negative weights improve model performance, while Differential transformer [45] showed benefits from differencing two attention matrices to obtain flexible weights. In linear attention, operations that reduce or delete historical state matrix elements outperform simple accumulation approaches [9, 14, 31, 46, 47]. In the following sections, we will show that our ZeroS method constructs zero-sum weights based on softmax, improving performance while maintaining numerical stability compared to both standard and linear attention variants. Compared to previous linear attention, ZeroS enables more effective control of radial weights and decoupled angular components in (t, i) pairs from step t information.

3 Methodology

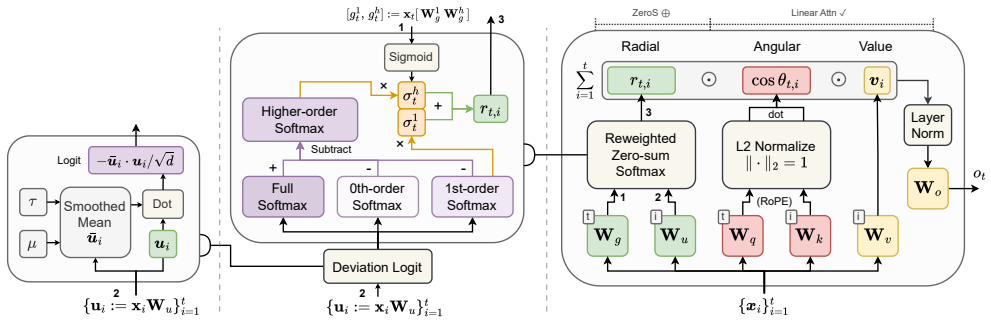
In this section,we demonstrate that using softmax residual terms with zero-sum weights (eliminating zero-order terms) and decoupling radial-angular components in linear attention achieves three key objectives: 1) enabling numerically stable negative weights in a single attention layer for expressing differential and contrastive operations, 2) capturing the essential length-angle interactions in attention weights that allow positive-negative flipping effects, and 3) permitting the current step t to effectively influence shareable accumulated weights while maintaining linear time complexity. The overall architecture of the final ZeroS block introduced in this section is shown in Fig. 1.

3.1 The Expansion of Softmax Function

Recent research has attempted to approximate softmax using Taylor expansions [48–51]. For input scalars $\{s_i\}_{i=1}^t$, with $\bar{s} = \frac{1}{t}\sum_{j=1}^t s_j$ and $\delta_i = s_i - \bar{s}$, the second-order Taylor expansion is:

$$\mathrm{softmax}(s_i) \approx \frac{1}{t} + \frac{1}{t}\delta_i + \frac{1}{2t}\Big(\delta_i^2 - \frac{1}{t}\sum_{j=1}^t \delta_j^2\Big) + O(\|s\|^3).$$

The zero-order term $\frac{1}{t}$ ensures $\sum_i \mathrm{softmax}(s_i) = 1$, while first-order terms reflect linear response, and higher-order terms capture nonlinear interactions and competitive relationships between weights. Computing second-order terms based on $s_{t,i} = \boldsymbol{q}_t\boldsymbol{k}_i^\top$ would require $O(d^3)$ complexity [49], making them impractical. Our approach differs: we use logits that depend only on step i, calculate full



(a) Calculation of the deviation logits (b) Reweighted zero-sum softmax block (c) The overall architecture of ZeroS (Zero-Sum Linear Attention)

Figure 1: Illustration of the zero-sum linear attention block, including the computation of deviation logits and the reweighted zero-sum softmax operation

softmax, zero and first-order terms, derive higher-order terms through their differentiation, and employ t-step-dependent gating factors to achieve interaction between (t, i) pairs at different orders.

The zero-order baseline primarily provides accumulated magnitude measurement, contributing $\frac{1}{\sqrt{t}}$-level norm reduction and convexity properties. However, it enables no interaction between scores. Eliminating this term creates zero-sum residual weights with both positive and negative values reflecting interaction strength. While full softmax encodes higher-order competitive effects only through positive weight magnitudes, zero-sum residual weights directly express these relationships between vectors based on the positive and negative weights, emphasizing contrastive components.

Proposition 3.1 (Convex vs. Zero-Sum Span)**.** *Let $\{\boldsymbol{v}_i\}_{i=1}^t \subset \mathbb{R}^d$, and write $\mathcal{C} = \{\sum_i \alpha_i \boldsymbol{v}_i : \alpha_i \geq 0,\ \sum_i \alpha_i = 1\}$, $\mathcal{Z} = \{\sum_i w_i \boldsymbol{v}_i : \sum_i w_i = 0\}$, where we denote the $(t-1)$-simplex by $\Delta_{t-1} = \{\alpha \in \mathbb{R}^t : \alpha_i \geq 0, \sum_i \alpha_i = 1\}$. Then, letting $\boldsymbol{v}_{\text{avg}} = \frac{1}{t}\sum_i \boldsymbol{v}_i$, $\{\sum_i \alpha_i \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}} : \alpha \in \Delta_{t-1}\} \subsetneq \{\sum_i w_i \boldsymbol{v}_i : \sum_i w_i = 0\}$, i.e. the zero-sum span of $\{\boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}}\}$ strictly contains the deviations achievable by convex weights, with strictness whenever the $\boldsymbol{v}_i$ are not all identical.*

Corollary 3.2 (Expressive Gain of Zero-Sum Attention)**.** *In a residual block $\boldsymbol{x}_t \mapsto \boldsymbol{x}_t + \sum_i w_i \boldsymbol{v}_i$, softmax weights $w_i = \alpha_i$ yield head deviations in $\{\sum_i \alpha_i \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}} : \alpha \in \Delta_{t-1}\}$; zero-order subtraction $w_i = \alpha_i - \frac{1}{t}$ yields head deviations in $\{\sum_i w_i \boldsymbol{v}_i : \sum_i w_i = 0\}$. Since $\{\sum_i \alpha_i \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}}\} \subsetneq \{\sum_i w_i \boldsymbol{v}_i : \sum_i w_i = 0\}$, zero-sum attention enlarges the set of deviation vectors the head can produce (and hence its expressivity), only without the uniform average direction.*

Zero-sum weights can express more complex interactions after removing the zero-order term, with expressivity reduction only in the orthogonal direction of $\boldsymbol{v}_{\text{avg}}$. This direction typically represents the lowest-cost basis since it requires only average pooling. We can recover this capability through multiple attention heads and layer stacking. To strictly ensure this direction is not lost, our implementation retains the zero-order term in the first layer, removing it in subsequent layers as described below.

3.2 Reweighted Zero-sum Softmax

We define the reweighted zero-sum softmax operation. For logit input $s_{t,i}$ at step t, we compute: $\bar{s}_t = \frac{1}{t}\sum_{j=1}^t s_{t,j}$, $\delta_{t,i} = s_{t,i} - \bar{s}_t$. Subtracting zero-order $(1/t)$ and first-order $(\delta_{t,i}/t)$ terms from softmax yields the residual:

$$\varepsilon_{t,i} = \frac{\exp(s_{t,i})}{\sum_{i=1}^t \exp(s_{t,i})} - \frac{1}{t} - \frac{\delta_{t,i}}{t} = O(\|\delta\|^2),$$

where $\sum_i \varepsilon_{t,i} = 0$ and $\sum_i \delta_{t,i}/t = 0$. We gate these components using learned scalars $\sigma_t^1 = \text{sigmoid}(g_t^1)$ and $\sigma_t^h = \text{sigmoid}(g_t^h)$, defining zero-sum weights:

$$w_{t,i} = \sigma_t^1 \frac{\delta_{t,i}}{t} + \sigma_t^h \varepsilon_{t,i}, \quad \sum_{i=1}^t w_{t,i} = 0.$$

This form assigns two gating weights: one for the first-order orthogonal direction and another for all directions of second-order and above. For the first attention layer, we can optionally preserve

the zero-order term using $\sigma_t^0 = \tanh(g_t^0)$, giving $w'_{t,i} = \sigma_t^0 \frac{1}{t} + \sigma_t^1 \frac{\delta_{t,i}}{t} + \sigma_t^h \varepsilon_{t,i}$, though experiments show this has minimal impact across most tasks.

Remark. A key advantage of this formulation for linear-time attention is that even with logits $s_{t,i} = s_i$ that are independent of t, we can still control interactions of different orders in the final weights $w_{t,i}$ through the t-step gating mechanism $\sigma_t^{(\cdot)}$ across orthogonal directions from softmax expansion. This gate reweighting approach $w_{t,i} = \sigma_t^1 \frac{\delta_i}{t} + \sigma_t^h \varepsilon_i$ effectively replaces the traditional linearization that decomposes $\exp(\boldsymbol{q}_t \boldsymbol{k}_i)$ into $\phi(\boldsymbol{q}_t)\phi(\boldsymbol{k}_i)$.

Proposition 3.3 (Preservation of Affine Hull and Expressivity)**.** *Let $\{\boldsymbol{v}_i\}_{i=1}^t \subset \mathbb{R}^d$ and write $\boldsymbol{v}_{\text{avg}} = \frac{1}{t}\sum_{i=1}^t \boldsymbol{v}_i,\ \Delta_i = \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}}$. A single head with full softmax (or full reweighted softmax with the zero-order term kept) can produce any point in the affine hull*

$$\text{Aff}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t\} \;=\; \Big\{ \boldsymbol{v}_{\text{avg}} + \sum_{i=1}^t \alpha_i \, \Delta_i : \sum_{i=1}^t \alpha_i = 1 \Big\}.$$

A single head without *the zero-order term (i.e. zero-sum weights) can produce any point in the linear span*

$$\text{Span}\{\Delta_1, \ldots, \Delta_t\} \;=\; \Big\{ \sum_{i=1}^t w_i \, \Delta_i : \sum_{i=1}^t w_i = 0 \Big\}.$$

Therefore, if you use one head (or one layer) that retains the zero-order term and then stack one or more heads (layers) that subtract it, the Minkowski sum of their reachable sets is exactly

$$\text{Aff}\{\boldsymbol{v}_i\} \;+\; \text{Span}\{\Delta_i\} \;=\; \text{Aff}\{\boldsymbol{v}_i\}.$$

In other words, after the first full attention layer, it already cover the entire affine hull, and the subsequent zero-sum attentions do not shrink that. The overall network can still express any affine combination of the $\boldsymbol{v}_i$.

Residual Stream Alignment When value vectors $\{v_i\}$ are centered and i.i.d., zero-sum attention produces $o_t = \sum_{i=1}^t w_{t,i} v_i$ with $\sum_i w_{t,i} = 0$ and $\mathbb{E}[o_t] = 0$. This aligns with decoder-only Transformer's residual stream ideology where $x_t \leftarrow x_t + \text{Attn}(x_t)$ should provide pure updates without constant bias. Subtracting the zero-order term naturally centers these residuals, improving training stability.

We now show that the reweighted zero-sum softmax achieves the same level of numerical stability as the original softmax.

Lemma 3.4 (Numerical Stability of Zero-Sum Softmax)**.** *Let $w_{t,i}$ be the reweighted zero-sum softmax weights with $\sum_i w_{t,i} = 0$, and assume each value vector satisfies $\|v_i\| \le B$. Then for any step t,*

$$\Big\| \sum_{i=1}^t w_{t,i} v_i \Big\| \;\le\; \max_i |w_{t,i}| \; \sum_{i=1}^t \|v_i\| \;\le\; B\, t \, \max_i |w_{t,i}|.$$

Moreover, since $|w_{t,i}| \le \max\big(\frac{1}{t}, \frac{|\delta_{t,i}|}{t}, \frac{|\rho_{t,i}|}{2t}\big)$ and $\delta, \rho = O(1)$ under bounded logits, we have $\max_i |w_{t,i}| = O(1/t)$ and hence $\big\|\sum_i w_{t,i} v_i\big\| = O(B)$, independent of t.

With controllable logits generation methods independent of t, such as the scaled dot-product in original softmax attention, the numerical stability of the above method remains well-controlled. This allows us to employ zero-sum weights that permit negative values while still achieving numerically stable outputs, even without the norm-preserving property of convex combinations.

Proposition 3.5 (Uniform Lipschitz Bound of Zero-Sum Softmax with decay factor $1/\sqrt{t}$)**.** *Assume each value vector satisfies $\|\boldsymbol{v}_i\| \le B$, each pre-softmax logit $s_{t,i}(\boldsymbol{x}) \in [-S, S]$ is L_s-Lipschitz in the input $\boldsymbol{x}$, each residual weight $w_{t,i}(\boldsymbol{x})$ obeys the scaling $|\, w_{t,i}(\boldsymbol{x}) - w_{t,i}(\boldsymbol{x}')| \;\le\; \frac{L_w}{t} \, \|\boldsymbol{x} - \boldsymbol{x}'\|$, for some constant L_w depending only on S and the sigmoid gates. Let the head output be $\boldsymbol{o}_t(\boldsymbol{x}) = \frac{1}{\sqrt{t}} \sum_{i=1}^t w_{t,i}(\boldsymbol{x})\, \boldsymbol{v}_i,\ \sum_{i=1}^t w_{t,i}(\boldsymbol{x}) = 0$. Then for any two inputs $\boldsymbol{x}, \boldsymbol{x}'$,*

$$\big\| \boldsymbol{o}_t(\boldsymbol{x}) - \boldsymbol{o}_t(\boldsymbol{x}') \big\| \;\le\; \frac{B\, L_w}{\sqrt{t}} \, \|\boldsymbol{x} - \boldsymbol{x}'\|.$$

The zero-sum update is ℓ_2-Lipschitz in its inputs with constant $O(1/\sqrt{t})$, ensuring stable gradients and activations independent of sequence length.

This proposition introduces a $1/\sqrt{t}$ decay factor that ensures reweighted zero-sum softmax maintains variance reduction similar to convex combinations, promoting training stability. However, since linear attention methods typically apply Layer Normalization to control output variance, LayerNorm effectively supersedes this factor and is sufficient to ensure gradient stability during training.

Reweighted Zero-sum Softmax in Linear Time To achieve linear-time computation, we simplify logits from $s_{t,i}$ to s_i by removing t-dependency. While we could use basic forms like $s_i = \boldsymbol{x}_i \mathbf{W}_s^{d\times 1}$ or quadratic forms $s_i = \boldsymbol{x}_i \mathbf{W}_s \mathbf{W}_s^\top \boldsymbol{x}_i^\top / d$ to emulate dot-products, we instead propose a design with a more meaningful representation.

We want these logits to express the deviation of step i relative to previous steps. We calculate the negative inner product between each step's vector $\boldsymbol{u}_i = \boldsymbol{x}_i \mathbf{W}_u$ and its cumulative average. For better assessment of initial steps, we introduce trainable parameters $\mu \in \mathbb{R}^{1\times d}$ and $\tau \in \mathbb{R}$ as a smoothing prior, calculating deviation logits s_i as:

$$s_i = -\frac{1}{\sqrt{d}} \boldsymbol{u}_i \bar{\boldsymbol{u}}_i^\top, \text{ where } \bar{\boldsymbol{u}}_i = \frac{e^\tau \mu + \sum_{j=1}^i \boldsymbol{u}_j}{e^\tau + i}$$

Let $r_{t,i}$ represent the final computed reweighted softmax result:

$$\delta_{t,i} = s_i - \bar{s}_t,\ \varepsilon_{t,i} = \frac{\exp(s_i)}{\sum_{i=1}^t \exp(s_i)} - \frac{1}{t} - \frac{\delta_{t,i}}{t},$$

$$r_{t,i} = \sigma_t^1 \frac{\delta_{t,i}}{t} + \sigma_t^h \varepsilon_{t,i} = \frac{\sigma_t^h}{\sum_{i=1}^t \exp(s_i)} \exp(s_i) + \frac{(\sigma_t^1 - \sigma_t^h)}{t}(s_i - \bar{s}_t) - \frac{\sigma_t^h}{t}$$

where $\sigma_t^1 = \text{sigmoid}(\boldsymbol{x}_t \mathbf{W}_g^1),\ \sigma_t^h = \text{sigmoid}(\boldsymbol{x}_t \mathbf{W}_g^h),\ \mathbf{W}_g^1, \mathbf{W}_g^h \in \mathbb{R}^{d\times 1}$. The terms dependent on t and i are effectively separated, enabling linear-time computation through prefix sums.

3.3 ZeroS Linear Attention: Interaction Between Radial and Angular Components

The reweighted zero-sum softmax provides strong foundations for linear attention by yielding numerically stable weights (including negative values) with computational simplicity while enabling high-order (t, i) interactions in token mixing. We leverage linear-time logit inputs that depend only on step i and implement effects on different softmax orders through step t gating.

However, our earlier discussion showed that the angle-flipping effect in softmax attention's $\exp(\|\boldsymbol{q}_t\|\|\boldsymbol{k}_i\|\cos\theta)$ significantly impacts final weights. While reweighted zero-sum softmax effectively models length interactions through i-step logits and t-step gating, it lacks control over directional influence when measuring vector differences in (t, i) pairs. Since zero sum weights provide inherent stability, no longer need to place $\cos\theta$ in the denominator normalizer, we can directly multiply the angular component ($\cos\theta$) with the reweighted softmax radial component without positivity constraints. This approach enables seamless integration with rotary positional embedding (RoPE) [52], making the angle term's role in measuring relative distance more explicit.

Zero-Sum Linear Attention (ZeroS) We use normalized vectors $\hat{\boldsymbol{k}}_i = \boldsymbol{k}_i / \|\boldsymbol{k}_i\|$ and $\hat{\boldsymbol{q}}_t = \boldsymbol{q}_t / \|\boldsymbol{q}_t\|$, with $r_{t,i}$ as the radial component and $\cos\theta$ as the angular component. ZeroS produces the output:

$$\boldsymbol{o}_t = \sum_{i=1}^t r_{t,i}\ \cos\theta\ \boldsymbol{v}_i, \quad \cos\theta = \hat{\boldsymbol{q}}_t \hat{\boldsymbol{k}}_i^\top$$

With RoPE's block-diagonal rotary matrix applied, the angular term becomes $\cos\theta' = \hat{\boldsymbol{q}}_t \mathbf{R}_{t-i} \hat{\boldsymbol{k}}_i^\top$. Both $r_{t,i}$ and $\cos\theta$ are centered values, preserving zero-sum properties in the weights $r_{t,i}\cos\theta$. Though not strictly positive (unlike traditional radial components), $r_{t,i}$ captures magnitude effects from step i, reflecting length-related interactions between (t, i) pairs.

Linear-Time Scan With logits that depend only on step i (e.g. $s_i = -\frac{1}{\sqrt{d}} \boldsymbol{u}_i \bar{\boldsymbol{u}}_i^\top$ with $\boldsymbol{u}_i = \boldsymbol{x}_i \mathbf{W}_u$ and $\bar{\boldsymbol{u}}_i = \frac{e^\tau \mu + \sum_{j=1}^i \boldsymbol{u}_j}{e^\tau + i}$), the radial weight at time t can be decomposed into the full softmax term, the 0th-order baseline, and the 1st-order term: $\underbrace{\frac{e^{s_i}}{\sum_{j=1}^t e^{s_j}}}_{\text{Full}}$, $\underbrace{\frac{1}{t}}_{\text{0th}}$, $\underbrace{\frac{1}{t} s_i - \frac{1}{t^2} \sum_{j=1}^t s_j}_{\text{1st}}$. Hence the higher-order zero-sum residual is Full $-$ 0th $-$ 1st. We realize ZeroS by gating the first-order zero-sum and

Table 1: Evaluation Results of ZeroS on the MAD benchmark.

Model	Compress	Fuzzy Recall	In-Context Recall	Memorize	Noisy Recall	Selective Copy	Average
Hyena	45.2	7.90	81.7	89.5	78.8	93.1	66.0
MultiHead Hyena	44.8	14.4	99.0	89.4	98.6	93.0	73.2
Mamba	52.7	6.70	90.4	89.5	90.1	86.3	69.3
GLA	38.8	6.90	80.8	63.3	81.6	88.6	60.0
DeltaNet	42.2	35.7	100	52.8	100	100	71.8
LinAttn	31.1	8.15	91.0	74.9	75.6	93.1	62.3
Transformer	51.6	29.8	94.1	85.2	86.8	99.6	74.5
ZeroS	44.0	14.9	99.9	88.1	96.1	97.8	73.5
ZeroS-SM	45.2	28.0	100	84.3	96.6	98.5	75.4

the higher-order residual with $\sigma_t^1 = \mathrm{sigmoid}(\boldsymbol{x}_t \mathbf{W}_g^1)$ and $\sigma_t^h = \mathrm{sigmoid}(\boldsymbol{x}_t \mathbf{W}_g^h)$, and optionally in the first layer retaining the 0th-order baseline $\sigma_t^0 = \mathrm{sigmoid}(\boldsymbol{x}_t \mathbf{W}_g^0)$ (with fixed $\sigma_t^0 = 0$ by default). Using normalized directions $\hat{\boldsymbol{q}}_t = \boldsymbol{q}_t / \|\boldsymbol{q}_t\|$ and $\hat{\boldsymbol{k}}_i = \boldsymbol{k}_i / \|\boldsymbol{k}_i\|$, we maintain the following prefix scans at step t:

$$E_t = \sum_{i=1}^{t} e^{s_i}, \quad P_t = \sum_{i=1}^{t} s_i, \quad \mathbf{F}_t = \sum_{i=1}^{t} e^{s_i}\, \hat{\boldsymbol{k}}_i^\top \boldsymbol{v}_i, \quad \mathbf{G}_t = \sum_{i=1}^{t} s_i\, \hat{\boldsymbol{k}}_i^\top \boldsymbol{v}_i, \quad \mathbf{H}_t = \sum_{i=1}^{t} \hat{\boldsymbol{k}}_i^\top \boldsymbol{v}_i.$$

The output is then a gated activation of these scans by the current step's angular vector $\hat{\boldsymbol{q}}_t$:

$$\begin{aligned} \boldsymbol{o}_t &= \hat{\boldsymbol{q}}_t \Bigg[\underbrace{\sigma_t^h \left(\frac{1}{E_t}\mathbf{F}_t - \frac{1}{t}\mathbf{G}_t + \left(\frac{P_t}{t^2} - \frac{1}{t}\right)\mathbf{H}_t \right)}_{\sigma_t^h \text{ (Full } - \text{ 0th } - \text{ 1st)}} + \underbrace{\sigma_t^1 \left(\frac{1}{t}\mathbf{G}_t - \frac{P_t}{t^2}\mathbf{H}_t \right)}_{\sigma_t^1 \text{ (1st restore)}} + \underbrace{\sigma_t^0 \frac{1}{t}\mathbf{H}_t}_{\sigma_t^0 \text{ (optional 0th restore)}} \Bigg] \\ &= \hat{\boldsymbol{q}}_t \big(\alpha_t \mathbf{F}_t + \beta_t \mathbf{G}_t + \gamma_t \mathbf{H}_t \big), \end{aligned}$$

where

$$\alpha_t = \frac{\sigma_t^h}{E_t}, \qquad \beta_t = \frac{\sigma_t^1 - \sigma_t^h}{t}, \qquad \gamma_t = \left(\frac{P_t}{t^2} - \frac{1}{t}\right)\sigma_t^h - \frac{P_t}{t^2}\sigma_t^1 + \frac{\sigma_t^0}{t}.$$

This scan keeps only $O(d^2)$ state $(\mathbf{F}_t, \mathbf{G}_t, \mathbf{H}_t)$ and updates in $O(d^2)$ per step, yielding overall $O(Nd^2)$ time and $O(d^2)$ memory while implementing the zero-sum weighting. Moreover, our reweighted zero-sum approach can also be directly applied to standard softmax attention. See section A.1.6 for more details.

4 Experiments

ZeroS's zero-sum formulation enhances the attention layer's expressivity for complex operations, particularly evident in in-context learning tasks [37, 53]. We evaluate both linear-time ZeroS and quadratic-time ZeroS-SM on recent in-context learning benchmarks, along with experiments on NLP, image, and time series tasks. In all experiments, we directly replaced the multi-head attention module with ZeroS under original benchmark settings, preserving all other components (MLP/GLU, embeddings, hyperparameters) to ensure strict alignment with previous standards.

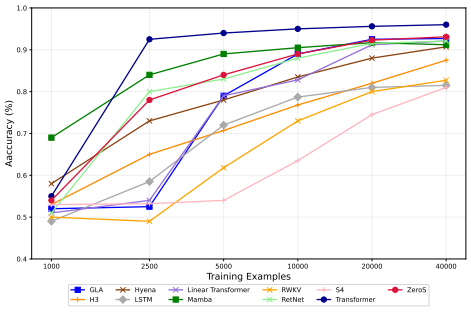


Figure 2: Evaluation of ZeroS on RegBench.

We previously described the prefix-sum computation of autoregressive ZeroS. For the encoder-only ZeroS, the summation simply spans all timesteps. We use the causal version of ZeroS for all datasets except image modeling. We provide a more detailed description of the experimental datasets in Appendix A.5. For simplicity, we do not apply first-layer 0-th order term addition in our experiments.

MAD We evaluate ZeroS on the MAD benchmark [54], which tests sequence models on in-context tasks. As shown in Table 1, ZeroS outperforms other linear-time models (Hyena, Mamba, GLA, DeltaNet, LinAttn [9, 14, 31, 55]), achieving performance closest to Transformer, while ZeroS-SM further improves upon Transformer's average score. Task-level analysis shows ZeroS significantly outperforms LinAttn on In-Context and Noisy Recall tasks, supporting our hypothesis that zero-sum weights enhance algorithmic abilities. However, on tasks like Compress and Memorize that rely less on complex representations, ZeroS provides minimal gains. Unlike DeltaNet, which actively deletes memory states, ZeroS maintains strong memorization despite using negative weights, indicating that our zero-order modifications preserve sequence memory capacity.

Table 2: Evaluation Results on WikiText

Model	PPL (val)	PPL (test)	Params (M)
FLASH	25.92	26.7	42.17
1+elu	27.44	28.05	44.65
Performer	62.5	63.16	44.65
cosFormer	26.53	27.06	44.65
Syn(D)	31.31	32.43	46.75
Syn(R)	33.68	34.78	46.75
gMLP	28.08	29.13	47.83
S4	38.34	39.66	45.69
DSS	39.39	41.07	45.63
GSS	29.61	30.74	43.84
RWKV-4	24.31	25.07	46.23
LRU	29.86	31.12	46.75
TNN	23.98	24.67	48.66
Mamba	22.58	23.19	44.99
HGRN2	23.1	23.73	44.66
Transformer	24.4	24.78	44.65
ZeroS	23.91	24.61	46.31
ZeroS-SM	23.62	24.17	44.69

MQAR We follow the setup of [56] for the MQAR task, which evaluates models' ability to learn induction heads for in-context associative recall. Using the same hyperparameter sweep, Fig. 3 shows ZeroS performs comparably to vanilla attention across most configurations.

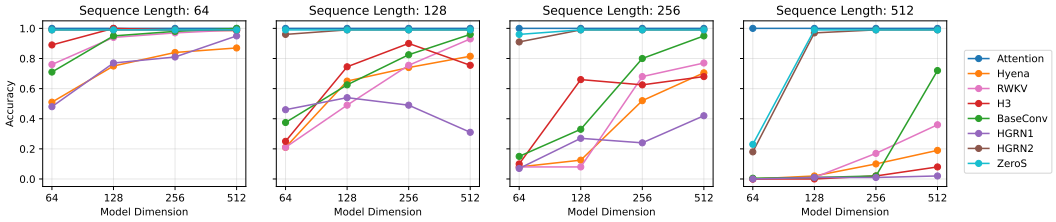


Figure 3: Performance evaluation on the MQAR benchmark, illustrating the relationship between model dimension (x-axis) and accuracy (y-axis). ZeroS demonstrates consistent performance advantages over other structures across all experimental configurations.

RegBench We evaluate ZeroS on RegBench [57] following the original experimental setup (Figure 2). RegBench tests models' ability to infer regular language structures from examples. ZeroS outperforms linear-time baselines including GLA, RetNet [58], and RWKV [59].

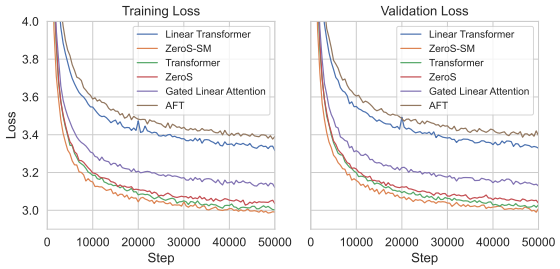


Figure 4: Performance Evaluation of ZeroS on OWT2

4.1 Language Modeling

WikiText We conduct language modeling on WikiText-103 following [60]'s setup, with results in Table 2. ZeroS outperforms vanilla Transformer at this smaller scale, demonstrating its efficiency. ZeroS-SM yields further improvements, showing enhanced reasoning capability from the zero-order term removal.

OWT2 We evaluate ZeroS on OpenWebText2 (OWT2) [61] using a 12-layer, 768-dimensional GPT-2 architecture with various token layers (see Appendix A.5). Figure 4 shows ZeroS tracks much closer to vanilla Transformer than other linear methods like AFT [62] and GLA, while ZeroS-SM further improves upon vanilla Transformer performance.

Image Modeling Following HGRN2 [60], wee valuate ZeroS on ImageNet by replacing the DeiT-Tiny architecture's softmax atten-

Table 3: Comparative analysis of image classification performance on ImageNet-1k.

	DeiT-Tiny	
Model	**Top-1 Acc**	**Params (M)**
DeiT	72.20	5.7
TNN	72.29	6.4
HGRN1	74.40	6.1
HGRN2	75.39	6.1
ZeroS	75.51	6.0

Table 4: Evaluation of the ZeroS performance on the Time Series Forecasting Benchmark

Models	**ZeroS**		GLA		AFT		iTransformer		PatchTST		DLinear	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	0.218	0.265	0.223	0.267	0.220	0.266	0.232	0.274	0.221	0.261	0.233	0.282
Solar	0.192	0.256	0.204	0.266	0.198	0.259	0.219	0.284	0.202	0.254	0.216	0.277
ETTh1	0.414	0.433	0.418	0.439	0.409	0.433	0.454	0.467	0.413	0.431	0.422	0.436
ETTh2	0.341	0.392	0.342	0.390	0.337	0.390	0.374	0.410	0.330	0.379	0.426	0.444
ETTm1	0.347	0.387	0.357	0.394	0.348	0.386	0.373	0.401	0.346	0.380	0.347	0.376
ETTm2	0.245	0.312	0.250	0.315	0.246	0.311	0.265	0.332	0.247	0.312	0.252	0.326

Table 6: Ablation Study on the MAD benchmark.

Model	Compress	Fuzzy Recall	In-Context Recall	Memorize	Noisy Recall	Selective Copy	Average
ZeroS	44.0	14.9	99.9	88.1	96.1	97.8	73.5
ZeroS w/ 0-th	42.0	10.5	91.4	85.2	90.0	97.1	69.4
ZeroS w/o RWSM	36.3	10.6	91.8	81.7	89.7	95.3	67.6
ZeroS w/o Gating	39.7	13.5	96.3	83.0	94.6	97.8	70.8
ZeroS w/o Norm	39.1	12.3	89.0	87.0	91.7	97.1	69.4
ZeroS-SM	45.2	28.0	100	84.3	96.6	98.5	75.4

tion with our encoder-only implementation. As shown in Table 3, ZeroS outperforms previous 294 methods including TNN [63] and HGRN1 [64] under comparable parameter budgets.

Time Series Following the setup in [65], we evaluate ZeroS on time series forecasting tasks. ZeroS outperforms both efficient sequence models (GLA, AFT) and domain-specific approaches (iTransformer [66], PatchTST [67]) on most datasets.

4.2 Ablation Studies

Table 5: Ablation Study on WikiText-103

Model	PPL (val)	PPL (test)	Params (M)
ZeroS	23.91	24.61	46.31
ZeroS w/ 0-th	24.05	24.74	46.31
ZeroS w/o RWSM	24.21	24.97	46.31
ZeroS-SM	23.62	24.17	44.69

We conduct ablation studies on MAD and WikiText-103 to analyze key components of ZeroS. Reintroducing the 0-th order softmax term reduces performance on In-Context Recall, Noisy Recall, and WikiText, confirming the representational advantage of zero-sum weights. Replacing the reweighted zero-sum softmax with standard softmax further degrades performance, highlighting the expressive gap between convex combinations and our flexible zero-sum mechanism. Ablating the gating component causes moderate performance drops across most tasks, suggesting it contributes broadly to model flexibility. Finally, removing LayerNorm notably impacts performance on In-Context Recall but not on simpler tasks like Memorize, indicating stable variance is particularly critical for algorithmic reasoning: consistent with normalization's role in linear attention mechanisms. See §A.3 for additional baselines and ablations.

5 Conclusion and Limitation

We introduced Zero-Sum Linear Attention (ZeroS), addressing fundamental limitations of linear attention by removing the constant zero-order term from softmax and reweighting the resulting zero-sum residuals. Our approach enables higher-order token interactions while maintaining O(N) complexity, bridging the performance gap between linear and quadratic attention methods. Evaluations across diverse tasks show ZeroS matches or exceeds standard softmax attention while offering significant efficiency advantages, challenging the belief that expressivity-efficiency tradeoffs in attention mechanisms are inevitable.

As for the limitation, our research prioritizes improving attention's algorithmic expressivity rather than providing engineering optimizations like GPU acceleration implementations found in Mamba or GLA [9, 14]. Also, our resource constraints prevented large-scale model training and evaluation on LLM benchmarks, which would involve numerous factors. This focused approach allowed us to precisely identify ZeroS's algorithmic improvements without requiring extensive engineering or computational resources that are typically needed for optimizing large benchmark metrics. Additionally, our evaluation of ZeroS primarily focuses on autoregressive tasks. Future work may explore its capabilities on non-causal tasks to further extend its applicability.

References

[1] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[4] Alec Radford. Improving language understanding by generative pre-training. *OpenAI technical report*, 2018.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5884–5888. IEEE, 2018.

[7] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[8] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[9] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[10] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *International conference on machine learning*, pages 9099–9117. PMLR, 2022.

[11] Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. The devil in linear transformer. *arXiv preprint arXiv:2210.10340*, 2022.

[12] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

[13] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[14] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024.

[15] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[16] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[17] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[18] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14138–14148, 2021.

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

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

[21] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[22] Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

[23] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.

[24] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pages 10183–10192. PMLR, 2021.

[25] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[26] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in neural information processing systems*, 32, 2019.

[27] Oliver Richter and Roger Wattenhofer. Normalized attention without probability cage. *arXiv preprint arXiv:2005.09561*, 2020.

[28] Pierre Baldi and Roman Vershynin. The quarks of attention: Structure and capacity of neural attention building blocks. *Artificial Intelligence*, 319:103901, 2023.

[29] Weixuan Sun, Zhen Qin, Hui Deng, Jianyuan Wang, Yi Zhang, Kaihao Zhang, Nick Barnes, Stan Birchfield, Lingpeng Kong, and Yiran Zhong. Vicinity vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12635–12649, 2023.

[30] Krzysztof Choromanski, Haoxian Chen, Han Lin, Yuanzhe Ma, Arijit Sehanobish, Deepali Jain, Michael S Ryoo, Jake Varley, Andy Zeng, Valerii Likhosherstov, et al. Hybrid random features. *arXiv preprint arXiv:2110.04367*, 2021.

[31] Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv:2406.06484*, 2024.

[32] Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Xiao Luo, Yu Qiao, et al. Transnormerllm: A faster and better large language model with improved transnormer. *arXiv preprint arXiv:2307.14995*, 2023.

[33] 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. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[34] Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with relu in vision transformers. *arXiv preprint arXiv:2309.08586*, 2023.

[35] Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, et al. Theory, analysis, and best practices for sigmoid self-attention. *arXiv preprint arXiv:2409.04431*, 2024.

[36] Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on relu and softmax in transformer. *arXiv preprint arXiv:2302.06461*, 2023.

[37] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 57125–57211. Curran Associates, Inc., 2023.

[38] Hengyu Fu, Tianyu Guo, Yu Bai, and Song Mei. What can a single attention layer learn? a study through the random features lens. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11912–11951. Curran Associates, Inc., 2023.

[39] Fanqi Yan, Huy Nguyen, Pedram Akbarian, Nhat Ho, and Alessandro Rinaldo. Sigmoid self-attention is better than softmax self-attention: A mixture-of-experts perspective. *arXiv preprint arXiv:2502.00281*, 2025.

[40] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[41] Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 01 2020.

[42] Shaked Brody, Uri Alon, and Eran Yahav. On the expressivity role of layernorm in transformers' attention, 2023.

[43] Ang Lv, Ruobing Xie, Shuaipeng Li, Jiayi Liao, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. More expressive attention with negative weights, 2025.

[44] Dongchen Han, Yifan Pu, Zhuofan Xia, Yizeng Han, Xuran Pan, Xiu Li, Jiwen Lu, Shiji Song, and Gao Huang. Bridging the divide: Reconsidering softmax and linear attention. *Advances in Neural Information Processing Systems*, 37:79221–79245, 2024.

[45] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Yingcong Li, Davoud Ataee Tarzanagh, Ankit Singh Rawat, Maryam Fazel, and Samet Oymak. Gating is weighting: Understanding gated linear attention through in-context learning, 2025.

[47] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

[48] Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, and Emad Barsoum. Qt-vit: Improving linear attention in vit with quadratic taylor expansion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 83048–83067. Curran Associates, Inc., 2024.

[49] Tobias Christian Nauen, Sebastian Palacio, and Andreas Dengel. Taylorshift: Shifting the complexity of self-attention from squared to linear (and back) using taylor-softmax. In Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa, Cheng-Lin Liu, Saumik Bhattacharya, and Umapada Pal, editors, *Pattern Recognition*, pages 1–16, Cham, 2025. Springer Nature Switzerland.

[50] Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff, 2025.

[51] Yuwei Qiu, Kaihao Zhang, Chenxi Wang, Wenhan Luo, Hongdong Li, and Zhi Jin. Mb-taylorformer: Multi-branch efficient transformer expanded by taylor formula for image dehazing, 2023.

[52] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[53] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022.

[54] Michael Poli, Armin W Thomas, Eric Nguyen, Pragaash Ponnusamy, Björn Deiseroth, Kristian Kersting, Taiji Suzuki, Brian Hie, Stefano Ermon, Christopher Ré, Ce Zhang, and Stefano Massaroli. Mechanistic design and scaling of hybrid architectures, 2024.

[55] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models, 2023.

[56] Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models, 2023.

[57] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms, 2024.

[58] 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.

[59] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023.

[60] Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*, 2024.

[61] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

[62] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer, 2021.

[63] Zhen Qin, Xiaodong Han, Weixuan Sun, Bowen He, Dong Li, Dongxu Li, Yuchao Dai, Lingpeng Kong, and Yiran Zhong. Toeplitz neural network for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[64] Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion, 2024.

[65] Jiecheng Lu and Shihao Yang. Linear transformers as var models: Aligning autoregressive attention mechanisms with autoregressive forecasting, 2025.

[66] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[67] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[68] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[69] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 22419–22430, 2021.

[70] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 95–104. ACM, 2018.

[71] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11106–11115. AAAI Press, 2021.

A Technical Appendices and Supplementary Material

A.1 Additional Theoretical Discussion

Proposition A.1 (Convex vs. Zero-sum Span)**.** *Let $v_1, \dots, v_t \in \mathbb{R}^d$ and denote their centroid by $v_{\text{avg}} = \frac{1}{t}\sum_{i=1}^t v_i$. Write the deviation matrix*

$$\Delta V = [\, v_1 - v_{\text{avg}},\, \dots,\, v_t - v_{\text{avg}} \,] \in \mathbb{R}^{d\times t}.$$

Define the convex hull and zero-sum spaces

$$\mathcal{C} := \Big\{ \sum_{i=1}^t \alpha_i v_i : \alpha_i \geq 0,\ \sum_{i=1}^t \alpha_i = 1 \Big\}, \qquad \mathcal{Z} := \Big\{ \sum_{i=1}^t w_i v_i : \sum_{i=1}^t w_i = 0 \Big\}.$$

Then

$$\mathcal{C} \;=\; v_{\text{avg}} + \big\{ \Delta V\, \alpha \;:\; \alpha \in \Delta_{t-1} \big\} \;\subsetneq\; v_{\text{avg}} + \big\{ \Delta V\, w \;:\; w \in \mathbb{R}^t,\ \mathbf{1}^\top w = 0 \big\} \;=\; v_{\text{avg}} + \mathcal{Z},$$

where Δ_{t-1} is the $(t-1)$-dimensional probability simplex.

Corollary A.2 (Expressive Capacity)**.** *Assume the v_i are affinely independent and $d \geq t-1$, so that* $\operatorname{rank} \Delta V = t-1$. *Then*

$$\dim(\mathcal{C}) \;=\; t-1, \qquad \dim(\mathcal{Z}) \;=\; t-1,$$

but $\mathcal{C}$ is bounded, whereas $\mathcal{Z}$ is an unbounded linear subspace of the same dimension. Consequently $\mathcal{C} \subsetneq v_{\text{avg}} + \mathcal{Z}$, and zero-sum attention can realise outputs unattainable by any convex-combination attention.

Proof sketch of Proposition A.1 and Corollary A.2.

1. **Convex representation.** For any $\alpha \in \Delta_{t-1}$, $\sum_i \alpha_i v_i = v_{\text{avg}} + \Delta V\, \alpha$.
2. **Zero-sum representation.** If $w \in \mathbb{R}^t$ satisfies $\mathbf{1}^\top w = 0$, then $\sum_i w_i v_i = \Delta V\, w$.
3. **Strict inclusion.** The vector $w = (1, -1, 0, \dots, 0)$ lies in the hyperplane $\mathbf{1}^\top w = 0$ but not in the simplex Δ_{t-1}; hence $v_{\text{avg}} + \Delta V\, w \in v_{\text{avg}} + \mathcal{Z} \setminus \mathcal{C}$.
4. **Dimensionality.** Under affine independence and $d \geq t-1$, the matrix ΔV has rank $t-1$. Linear images preserve dimension, giving the stated dimensions of $\mathcal{C}$ and $\mathcal{Z}$ and proving the corollary.

□

A.1.1 Proof of Proposition 3.1 (Convex vs. Zero-Sum Span).

Let $\{\boldsymbol{v}_i\}_{i=1}^t \subset \mathbb{R}^d$ and define

$$\mathcal{C} = \Big\{ \sum_{i=1}^t \alpha_i \boldsymbol{v}_i : \alpha_i \geq 0,\ \sum_{i=1}^t \alpha_i = 1 \Big\}, \quad \mathcal{Z} = \Big\{ \sum_{i=1}^t w_i \boldsymbol{v}_i : \sum_{i=1}^t w_i = 0 \Big\},$$

and $\boldsymbol{v}_{\text{avg}} = \frac{1}{t}\sum_{i=1}^t \boldsymbol{v}_i$. For any $\boldsymbol{y} \in \mathcal{C}$ with weights $\alpha \in \Delta_{t-1}$,

$$\boldsymbol{y} - \boldsymbol{v}_{\text{avg}} = \sum_{i=1}^t \Big(\alpha_i - \tfrac{1}{t}\Big)\boldsymbol{v}_i, \qquad \text{with} \quad \sum_{i=1}^t \Big(\alpha_i - \tfrac{1}{t}\Big) = 0.$$

Hence the centered convex set equals

$$\mathcal{C}_{\text{dev}} = \Big\{ \sum_{i=1}^t w_i \boldsymbol{v}_i : \sum_{i=1}^t w_i = 0,\ \ w_i \geq -\tfrac{1}{t} \Big\},$$

which is $\mathcal{Z}$ with extra lower bounds on the coefficients. Therefore $\mathcal{C}_{\text{dev}} \subseteq \mathcal{Z}$.

To see the inclusion is strict unless all $\boldsymbol{v}_i$ are identical, pick $j \neq k$ with $\boldsymbol{v}_j \neq \boldsymbol{v}_k$ and consider $\boldsymbol{v}_j - \boldsymbol{v}_k \in \mathcal{Z}$ (take $w_j = 1, w_k = -1$, others 0). If $\boldsymbol{v}_j - \boldsymbol{v}_k \in \mathcal{C}_{\text{dev}}$, we would have

$$\alpha_j - \tfrac{1}{t} = 1, \quad \alpha_k - \tfrac{1}{t} = -1, \quad \alpha_i - \tfrac{1}{t} = 0 \ (i \notin \{j, k\}),$$

implying $\alpha_k = -1 + \frac{1}{t} < 0$ for $t \geq 2$, a contradiction. Thus $\boldsymbol{v}_j - \boldsymbol{v}_k \notin \mathcal{C}_{\text{dev}}$, and $\mathcal{C}_{\text{dev}} \subsetneq \mathcal{Z}$ whenever the $\boldsymbol{v}_i$ are not all equal. If all $\boldsymbol{v}_i$ coincide, both sets reduce to $\{0\}$. □

A.1.2 Proof of Corollary 3.2 (Expressive Gain of Zero-Sum Attention).

In a residual head $\boldsymbol{x}_t \mapsto \boldsymbol{x}_t + \sum_i w_i \boldsymbol{v}_i$, the deviation (relative to the average direction) produced by softmax weights α is

$$\sum_i \alpha_i \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}} \in \mathcal{C}_{\text{dev}}.$$

Subtracting the zero-order term corresponds to $w_i = \alpha_i - \frac{1}{t}$ with $\sum_i w_i = 0$, hence deviations lie in $\mathcal{Z}$. By Proposition 3.1, $\mathcal{C}_{\text{dev}} \subsetneq \mathcal{Z}$ (for non-degenerate $\{\boldsymbol{v}_i\}$), so zero-sum attention strictly enlarges the attainable deviation set and therefore the head's expressivity. The only removed direction is the uniform average $\boldsymbol{v}_{\text{avg}}$, which can be recovered across heads or layers. □

A.1.3 Proof of Proposition 3.3 (Preservation of Affine Hull and Expressivity).

Let $\{\boldsymbol{v}_i\}_{i=1}^t \subset \mathbb{R}^d$, $\boldsymbol{v}_{\text{avg}} = \frac{1}{t}\sum_{i=1}^t \boldsymbol{v}_i$, and $\Delta_i = \boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}}$.

(i) Full softmax / with zero-order term. A single head with full softmax produces

$$\mathcal{R}_{\text{full}} = \Big\{ \sum_{i=1}^t a_i \boldsymbol{v}_i : \; a_i \geq 0, \; \sum_{i=1}^t a_i = 1 \Big\} = \text{Conv}\{\boldsymbol{v}_i\} \subseteq \text{Aff}\{\boldsymbol{v}_i\}.$$

(ii) Zero-sum (without zero-order term). If $\sum_{i=1}^t w_i = 0$, then

$$\sum_{i=1}^t w_i \boldsymbol{v}_i = \sum_{i=1}^t w_i(\boldsymbol{v}_{\text{avg}} + \Delta_i) = \boldsymbol{v}_{\text{avg}} \sum_{i=1}^t w_i + \sum_{i=1}^t w_i \Delta_i = \sum_{i=1}^t w_i \Delta_i,$$

hence

$$\mathcal{R}_{\text{zero-sum}} = \Big\{ \sum_{i=1}^t w_i \boldsymbol{v}_i : \; \sum_{i=1}^t w_i = 0 \Big\} = \text{Span}\{\Delta_1, \ldots, \Delta_t\}.$$

Conversely, for any $s = \sum_{i=1}^t u_i \, \Delta_i \in \text{Span}\{\Delta_i\}$, let $\bar{u} = \frac{1}{t}\sum_{i=1}^t u_i$ and define $w_i = u_i - \bar{u}$. Then $\sum_{i=1}^t w_i = 0$ and $\sum_{i=1}^t w_i \boldsymbol{v}_i = \sum_{i=1}^t u_i \boldsymbol{v}_i - \bar{u} \sum_{i=1}^t \boldsymbol{v}_i = \sum_{i=1}^t u_i(\boldsymbol{v}_i - \boldsymbol{v}_{\text{avg}}) = \sum_{i=1}^t u_i \Delta_i = s$. Hence $\text{Span}\{\Delta_i\} \subseteq \mathcal{R}_{\text{zero-sum}}$, and thus $\mathcal{R}_{\text{zero-sum}} = \text{Span}\{\Delta_i\}$.

(iii) Stacking and Minkowski sum. For any $\boldsymbol{y} \in \text{Aff}\{\boldsymbol{v}_i\}$ we can write

$$\boldsymbol{y} = \boldsymbol{v}_{\text{avg}} + (\boldsymbol{y} - \boldsymbol{v}_{\text{avg}}),$$

where $\boldsymbol{v}_{\text{avg}} \in \text{Conv}\{\boldsymbol{v}_i\}$ and, since $\sum_i \Delta_i = 0$, we have $\boldsymbol{y} - \boldsymbol{v}_{\text{avg}} \in \text{Span}\{\Delta_i\}$. Therefore

$$\text{Conv}\{\boldsymbol{v}_i\} + \text{Span}\{\Delta_i\} = \text{Aff}\{\boldsymbol{v}_i\}.$$

Combining (i)–(iii) gives the claimed reachable sets for single heads and their equality to the affine hull when stacked. □

A.1.4 Proof of Lemma 3.4 (Numerical Stability of Zero-Sum Softmax).

Assume $\|\boldsymbol{v}_i\| \leq B$ for all i. By the triangle inequality,

$$\Big\| \sum_{i=1}^t w_{t,i} \, \boldsymbol{v}_i \Big\| \leq \sum_{i=1}^t |w_{t,i}| \, \|\boldsymbol{v}_i\| \leq B \, t \, \max_i |w_{t,i}|.$$

By the zero-sum softmax construction (see the main text), under bounded logits we have

$$|w_{t,i}| \;\leq\; \max\Big\{ \tfrac{1}{t}, \; \tfrac{|\delta_{t,i}|}{t}, \; \tfrac{|\rho_{t,i}|}{2t} \Big\}, \qquad \delta_{t,i} = s_{t,i} - \tfrac{1}{t} \sum_{j=1}^t s_{t,j},$$

and $\delta_{t,i}, \rho_{t,i} = O(1)$. Hence $\max_i |w_{t,i}| = O(1/t)$, and therefore

$$\Big\| \sum_{i=1}^t w_{t,i} \, \boldsymbol{v}_i \Big\| \;\leq\; B \, t \cdot O\big(\tfrac{1}{t}\big) = O(B),$$

which is independent of t. □

A.1.5 Proof of Proposition 3.5 (Uniform Lipschitz Bound with $1/\sqrt{t}$ Decay).

Let $\boldsymbol{o}_t(\boldsymbol{x}) = t^{-1/2} \sum_{i=1}^{t} w_{t,i}(\boldsymbol{x})\, \boldsymbol{v}_i$ with $\sum_{i=1}^{t} w_{t,i}(\boldsymbol{x}) = 0$ and $\|\boldsymbol{v}_i\| \le B$. For any $\boldsymbol{x}, \boldsymbol{x}'$,

$$\begin{aligned}\|\boldsymbol{o}_t(\boldsymbol{x}) - \boldsymbol{o}_t(\boldsymbol{x}')\| &= \frac{1}{\sqrt{t}} \Big\| \sum_{i=1}^{t} \big(w_{t,i}(\boldsymbol{x}) - w_{t,i}(\boldsymbol{x}')\big) \boldsymbol{v}_i \Big\| \\ &\le \frac{1}{\sqrt{t}} \sum_{i=1}^{t} |w_{t,i}(\boldsymbol{x}) - w_{t,i}(\boldsymbol{x}')|\, \|\boldsymbol{v}_i\| \le \frac{B}{\sqrt{t}} \sum_{i=1}^{t} |w_{t,i}(\boldsymbol{x}) - w_{t,i}(\boldsymbol{x}')|.\end{aligned}$$

By the Lipschitz assumption on the weights, $|w_{t,i}(\boldsymbol{x}) - w_{t,i}(\boldsymbol{x}')| \le (L_w/t)\, \|\boldsymbol{x} - \boldsymbol{x}'\|$, hence

$$\|\boldsymbol{o}_t(\boldsymbol{x}) - \boldsymbol{o}_t(\boldsymbol{x}')\| \le \frac{B}{\sqrt{t}} \sum_{i=1}^{t} \frac{L_w}{t}\, \|\boldsymbol{x} - \boldsymbol{x}'\| = \frac{B\, L_w}{\sqrt{t}}\, \|\boldsymbol{x} - \boldsymbol{x}'\|.$$

Thus the head is uniformly Lipschitz with constant $BL_w/\sqrt{t}$. □

A.1.6 Implementation of ZeroS Softmax Attention (ZeroS-SM)

Zero-sum for Standard Softmax Attention (ZeroS-SM) As shown in Figure 5, our reweighted zero-sum approach can be directly applied to standard softmax attention using logits $s_{t,i} = \boldsymbol{q}_t \boldsymbol{k}_i^\top / \sqrt{d}$, with matrix form:

$$\mathbf{S} = \frac{1}{\sqrt{d}}\, \mathbf{Q}\, \mathbf{K}^\top + \mathbf{M},\ \mathbf{A} = \mathrm{softmax}(\mathbf{S}),\ \mathbf{u} = \big[1, \tfrac{1}{2}, \ldots, \tfrac{1}{N}\big]^\top,\ \bar{\mathbf{S}} = \mathrm{diag}(\mathbf{u})\, (\mathbf{S}\, \mathbf{1}_N)$$

$$\boldsymbol{\Delta} = \mathrm{diag}(\mathbf{u})\, \big(\mathbf{S} - \bar{\mathbf{S}}\, \mathbf{1}_N^\top\big),\ \boldsymbol{\varepsilon} = \mathbf{A}\ -\ \mathrm{diag}(\mathbf{u})\, \mathbf{1}_N\, \mathbf{1}_N^\top\ -\ \boldsymbol{\Delta}$$

$$\mathbf{W} = (\mathbf{g}^1\, \mathbf{1}_N^\top) \odot \boldsymbol{\Delta}\ +\ (\mathbf{g}^h\, \mathbf{1}_N^\top) \odot \boldsymbol{\varepsilon},\ \mathbf{O} = \mathbf{W}\, \mathbf{V}$$

A.2 Runtime Efficiency of the ZeroS Implementation

In recurrent (scan) form, ZeroS maintains three $d \times d$ hidden-state bases

$$e^{s_i}\, \hat{\mathbf{k}}_i^\top \mathbf{v}_i, \quad s_i\, \hat{\mathbf{k}}_i^\top \mathbf{v}_i, \quad \hat{\mathbf{k}}_i^\top \mathbf{v}_i,$$

and reads them out with query-dependent gates; the outputs are summed. While a single fused CUDA kernel is not implemented, we obtain a practical implementation by invoking an existing linear-attention scan three times with different key/value bases:

```
# prepare reweighted queries q1, q2, q3 from q ...
out1 = run_linattn(q1, k * s_i_exp, v, mode='fused_chunk')  # e^{s_i}
out2 = run_linattn(q2, k * s_i,      v, mode='fused_chunk')  # s_i
out3 = run_linattn(q3, k,            v, mode='fused_chunk')  # 1
out  = out1 + out2 + out3
```

We replace the attention layer in a GPT-2 style Transformer (hidden size $= 768$, 12 heads, 12 layers) with various alternatives and evaluate at sequence length 1024. Baselines include implementations from the same library: LinAttn, GatedLinAttn, HGRN2, RWKV6, RWKV7, and softmax attention (naïve and FlashAttention). All runs use a single NVIDIA L40S, batch size 8, FP32. We report mean latency after warm-up. "Fwd" denotes full-sequence inference (no KV/hidden-state cache). As shown in Table 7. Under this three-scan implementation, ZeroS attains latency, throughput, and memory usage within the range of established linear-attention variants and close to FlashAttention on this setup.

A.3 Additional Baselines and Ablations

Setup. We augment the main results with recent sequence-modeling baselines: Mamba2, Hawk, GatedDeltaNet, and HedgeDog. The evaluation protocol, datasets, and metrics follow the main text.

Model	FwdLat (s)	FwdStd	TrainLat (s)	TrainStd	Thr.Fwd (tok/s)	Thr.Train (tok/s)	MemFwd (GB)	MemTrain (GB)
Softmax Attn (naïve)	0.1306	0.0006	0.3334	0.0009	62,740.89	24,574.18	9.61	10.34
RWKV7	0.0876	0.0016	0.2626	0.0010	93,491.90	31,199.35	9.81	10.61
RWKV6	0.0761	0.0005	0.2252	0.0010	107,653.00	36,382.92	9.49	9.62
ZeroS	0.0720	0.0008	0.1974	0.0011	113,855.38	41,491.29	7.48	7.61
HGRN2	0.0672	0.0013	0.1480	0.0009	121,955.16	55,336.55	6.14	6.43
LinAttn	0.0666	0.0010	0.1477	0.0009	122,949.71	55,447.86	5.79	5.90
Softmax Attn (FlashAttn)	0.0651	0.0014	0.1473	0.0008	125,836.28	55,620.75	5.45	5.56
GatedLinAttn	0.0600	0.0009	0.1331	0.0008	136,633.08	61,533.68	5.74	5.89

Table 7: Latency, throughput, and peak GPU memory on GPT-2 (768/12/12), sequence length 1024, batch size 8, FP32 on a single L40S. "Fwd" = full-sequence inference without caches.

Model	Compress	FuzzyRecall	In-ContextRecall	Memorize	NoisyRecall	SelectiveCopy	Average
ZeroS (Lin)	44.0	14.9	99.9	88.1	96.1	97.8	73.5
LinAttn	33.1	8.2	91.0	74.9	75.6	93.1	62.3
ZeroS (SoftmaxAttn)	45.2	28.0	100.0	84.3	96.6	98.5	75.4
SoftmaxAttn	51.6	29.8	94.1	85.2	86.8	99.6	74.5
Mamba2	43.6	21.1	96.4	86.9	96.7	93.3	73.0
Hawk	47.7	13.6	93.0	91.3	93.0	77.0	64.5
GatedDeltaNet	45.0	29.8	100.0	80.2	100.0	94.3	74.9
HedgeDog	43.2	17.9	55.9	83.4	46.0	98.4	57.4
Ablation Study							
ZeroS	44.0	14.9	99.9	88.1	96.1	97.8	73.5
w/o Angular	39.5	8.5	42.8	54.5	44.8	63.3	42.2
Ang: w/o PosEmb	35.8	9.4	73.3	46.2	66.2	45.8	46.1
Ang: additive PosEmb	38.1	14.2	94.1	86.6	87.2	93.8	69.0
w/o Radial	35.9	9.6	84.8	86.3	86.5	92.3	65.9
Rad: u_i (linear proj)	41.2	15.5	91.9	88.3	86.1	97.2	70.0
Rad: u_i (quad form)	40.9	15.4	92.6	86.6	90.8	98.5	70.8
Rad: u_i (2-distance)	40.1	14.8	97.6	82.5	93.6	97.8	71.1
Rad: u_i (averaging)	41.0	15.0	99.9	93.3	89.6	98.5	73.0

Table 8: Additional baselines and ablations on six MAD tasks; higher is better.

Results. Table 8 reports task accuracies (%). ZeroS attains high scores on In-Context Recall and Noisy Recall and yields a strong overall average. Ablations indicate that removing the angular component substantially degrades performance; additive positional embeddings help but do not match RoPE; and the default radial scoring (with negative similarity) achieves the best average among radial variants.

A.4 Illustrative Zero-Sum Construction Examples

Setup. Let $\{\mathbf{v}_i\}_{i=1}^t \subset \mathbb{R}^d$. A single softmax-attention layer produces $\mathbf{o} = \sum_i \alpha_i \mathbf{v}_i$ with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$, hence $\mathbf{o} \in \mathrm{Conv}(\{\mathbf{v}_i\})$. A single ZeroS layer can produce signed, zero-sum combinations $\mathbf{o} = \sum_i w_i \mathbf{v}_i$ with $\sum_i w_i = 0$. Below we list simple sequence-to-sequence mappings that are not representable by a single softmax-attention layer but are representable by a single ZeroS layer.

Example: Two-token difference. Target $\mathbf{o} = \mathbf{v}_1 - \mathbf{v}_2$. Softmax requires $\alpha_1 = 1, \alpha_2 = -1$ (invalid). ZeroS: $w_1 = 1$, $w_2 = -1$, others 0.

Example: Difference from the mean. Target $\mathbf{o} = \mathbf{v}_1 - \dfrac{1}{t}\sum_{i=1}^{t} \mathbf{v}_i$. Softmax needs negative mass on $\{\mathbf{v}_i\}_{i>1}$ (invalid). ZeroS: $w_1 = 1 - \frac{1}{t}$ and $w_{i>1} = -\frac{1}{t}$ (so $\sum_i w_i = 0$).

Example: Alternating differences. For even t, target $\mathbf{o} = \sum_{i=1}^{t/2} (\mathbf{v}_{2i-1} - \mathbf{v}_{2i})$. Softmax cannot realize alternating $\pm$ weights in one layer. ZeroS: $w_{2i-1} = 1$, $w_{2i} = -1$ (others 0).

A.5 Additional Dateset Description

A.5.1 MQAR

We adopt the Multi-Query Associative Recall (MQAR) task introduced by [56] to characterize a model's performance on repeated, input-dependent lookups over a large vocabulary in a single

forward pass. In the classic associative recall (AR) problem, we store a small, static dictionary of key-value pairs and issue a single, fixed-position query. MQAR generalizes AR by interleaving multiple key-value pairs, each encoded as two consecutive tokens (k_j, v_j), and allows multiple query tokens anywhere in a sequence of length N. Formally, given

$$\mathbf{x} = (x_0, ..., x_{N-1}), \qquad x_i \in \mathcal{V}$$

whenever $x_i = k_j$ for some $j < i$, the correct output is

$$y_i = v_j = x_{j+1}$$

and the model must satisfy this for all $1 \leq i < N$. By requiring repeated lookups at arbitrary positions, MQAR provides a sharp test of dynamic routing and associative recall, directly contrasting these mechanisms with softmax attention's flexibility and capacity to handle multiple simultaneous queries.

A.5.2 REGBENCH

RegBench [57] is a synthetic in-context learning benchmark that evaluates a model's ability to infer the structure of regular languages from only a few example strings provided in the prompt. Each problem instance presents $K \in [10, 20]$ example strings $\{d_1^{(i)}, \ldots, d_K^{(i)}\}$ drawn from the same stochastic regular language $L^{(i)}$ defined by a probabilistic finite automaton (PFA). To construct the PFA, RegBench samples a minimal deterministic finite automaton (DFA), the canonical formalization of regular languages. RegBench draws

$$n \sim \mathrm{Uniform}(4, 12), \quad c \sim \mathrm{Uniform}(4, 18), \quad m_i \sim \mathrm{Uniform}(1, 4),$$

and then samples a language-specific alphabet Σ of size c uniformly without replacement from a global symbol set of size $c_{\max} = 18$. Define the state set $\mathcal{S} = \{S_1, \ldots, S_n\} \cup \{S_0\}$ with accepting subset $\mathcal{S}_a = \{S_1, \ldots, S_n\}$. For each S_i, uniformly without replacement select m_i symbols $x_j \in \Sigma$ and m_i target states $S_j \in \mathcal{S} \setminus \{S_i\}$ to form edges (S_i, x_j, S_j), send all other symbols to S_0, and minimize via Hopcroft's algorithm to obtain the canonical DFA A'. The PFA inherits A''s topology, assigning

$$T(S_i, x_j, S_j) = \frac{1}{m_i}, \quad T(S_i, x', S') = 0 \quad \text{otherwise},$$

so that $\sum_{a \in \Sigma} \sum_{s' \in \mathcal{S}} T(s, a, s') = 1, \quad \forall\, s \in \mathcal{S}.$. From this PFA, K strings of length $\ell \sim \mathrm{Uniform}(1, 50)$ are sampled from S_0 by simulating $(x_t, S_t) \sim T(S_{t-1}, \cdot, \cdot)$ and concatenating $x_1 x_2 \cdots x_\ell$. Models then perform greedy next-token predictions

$$\hat{x}_j \;=\; \arg\max_x \, p_\theta\big(x \mid \mathbf{d}_{<j}^{(i)}\big),$$

and we report *DFA accuracy* as in Akyürek et al. (2024)

$$\mathrm{accuracy}(p_\theta, L_i) = \frac{1}{\mathrm{NT}} \sum_{\mathbf{d}^{(i)}} \sum_j [\mathbf{1}\Big[\hat{x}_j \in \{x' : L_i(x' \mid \mathbf{d}_{<j}^{(i)}) > 0\}\Big]],$$

where NT is the total number of tokens in the test set and $L_i(x' \mid \mathbf{d}_{<j}^{(i)})$ is the probability of predicting x' following context $\mathbf{d}_{<j}^{(i)}$ in the language L_i. We consider DFA accuracy, the fraction of predictions that correspond to valid transitions in the original DFA, as a direct measure of how faithfully the model has internalized the underlying regular-language structure.

A.5.3 MAD

We evaluate our proposed architecture using the Mechanistic Architecture Design (MAD) framework, a recently developed methodology for cost-effective evaluation of deep learning architectures [54]. MAD consists of a suite of capability-targeted benchmarks, including in-context recall, fuzzy recall, selective copying, and compression, that probe fundamental sequence modeling capabilities. This approach has been rigorously validated through extensive experimentation spanning over 500 language models from 70M to 7B parameters, demonstrating a strong correlation between performance on these targeted synthetic tasks and compute-optimal perplexity at scale. Through the employment of MAD, which serves as a reliable predictor of large-scale performance, we identify performance advantages without the need for prohibitive computational resources typically associated with architecture validation.

A.5.4 WikiText-103

WikiText-103 [68] is a large-scale language modeling dataset of over 103 million words compiled from 23,805 *Good* and 4,790 *Featured* Wikipedia articles that have been reviewed by humans, represent broad coverage, and meet common editorial standards. The dataset has long context windows, a large vocabulary of 267,735 types, and requires preservation of case, punctuation, and numerical information so that WikiText-103 accurately reflects the challenges of real-world text.

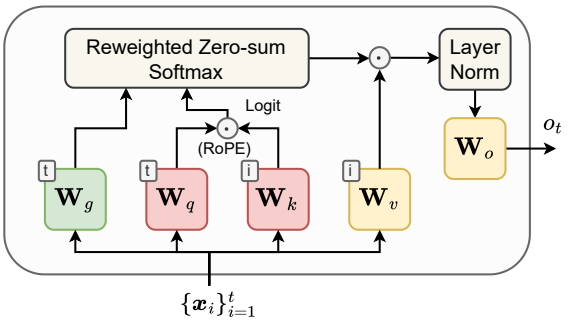


Figure 5: Block Architecture of The ZeroS-SM Layer

A.5.5 OpenWebText2

OpenWebText2[1] is a large-scale, cleaned and deduplicated web-text corpus created as an open reproduction of OpenAI's WebText dataset: URLs are first extracted from all Reddit submissions with a combined score greater than 3, then scraped, filtered, and deduplicated at both URL and document levels (using MinHash-LSH) to remove low-quality or redundant content. The resulting "plug-and-play" release comprises 17,103,059 documents (around 65.86 GB uncompressed), covering Reddit submissions from 2005 through April 2020, and serves as a high-diversity, up-to-date pretraining corpus for large language models. We use the code environment provided by nanoGPT[2] to implement this dataset training.

A.5.6 Time Series

We evaluate our module on the time series forecasting benchmark datasets below, following the experimental setup of [65]. **(1) Weather** [69][3]: 21 meteorological variables (e.g., temperature, humidity) collected every 10 minutes in 2020 from a weather station in Germany. **(2) Solar** [70][4]: Solar power outputs recorded every 10 minutes in 2006 from 137 photovoltaic plants in the U.S. **(3) ETT** [71][5]: Transformer load and temperature data sampled at 15-minute (ETTm1/ETTm2) and hourly (ETTh1/ETTh2) intervals from July 2016 to July 2018, including 7 key operational features.

A.6 Additional Description of Experimental Settings

Our detailed experimental setup is available at the provided code repository. All experiments introduced in this paper can be run on a single Nvidia RTX 4090. For faster training, we parallelize experiments across multiple GPUs. In all benchmarks, we replace the multi-head attention layers with ZeroS layers without modifying any other settings. We do not apply first-layer 0-th order term correction or the $1/\sqrt{t}$ variance scaling in any of our experiments.

A.7 Impact Statement

This paper introduces an efficient attention mechanism for transformer-based models. As a fundamental architectural improvement, ZeroS primarily affects upstream model capabilities rather than specific applications. The positive impacts include potential reductions in computational costs when processing long sequences. Like most foundational ML research, this work could indirectly contribute to both beneficial and potentially harmful applications depending on how downstream models implement it. However, as an architectural component rather than a deployed system, ZeroS itself poses minimal direct societal concerns.

[1] https://openwebtext2.readthedocs.io/en/latest
[2] https://github.com/karpathy/nanoGPT
[3] https://www.bgc-jena.mpg.de/wetter/
[4] http://www.nrel.gov/grid/solar-power-data.html
[5] https://github.com/zhouhaoyi/ETDataset

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's scope, as shown in section 2-4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Yes, as shown in last section of the main text.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, the paper make sure the assumptions and proof are accurate.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes. Please see the provided anonymous code repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, we have listed all the code environments used in the experiment, which includes an introduction to the data that was applied.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we fully inherited all the code from the benchmark code environment used, without making any modifications to the hyperparameters. This paper, as a simple extension to the softmax attention and linear attention modules, does not specify any additional hyperparameters.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We fully rely on the scores generated by the referenced benchmark experimental environments. In experiments such as MAD and RegBench, they conduct large-scale hyperparameter searches and select the optimal results. Error bars are not applicable to these scores.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Yes, all tasks used in this paper can be trained on the single Nvidia RTX 4090 GPU that we used. We accelerated the experiments by running multiple tasks in parallel, and we did not have any large-scale tasks that involved using multiple GPUs for distributed training for a single task.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, this study fully complies with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we include a impact statement section in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we have provided a complete list of all environments used in the experimental section, where all related assets information can be found.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we have presented the complete code we used in an anonymous code repository.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.