Higher-order Linear Attention

Yifan Zhang[1]   **Zhen Qin**   **Quanquan Gu**[2]

[1]Princeton University   [2]University of California, Los Angeles

yifzhang@princeton.edu   qgu@cs.ucla.edu

October 30, 2025

Abstract

The quadratic cost of scaled dot-product attention is a central obstacle to scaling autoregressive language models to long contexts. Linear-time attention and State Space Models (SSMs) provide scalable alternatives but are typically restricted to first-order or kernel-based approximations, which can limit expressivity. We introduce **Higher-order Linear Attention** (**HLA**), a causal, streaming mechanism that realizes higher interactions via compact prefix sufficient statistics. In the second-order case, HLA maintains a constant-size state and computes per-token outputs in linear time without materializing any $n \times n$ matrices. We give closed-form streaming identities, a strictly causal masked variant using two additional summaries, and a chunk-parallel training scheme based on associative scans that reproduces the activations of a serial recurrence exactly. We further outline extensions to third and higher orders. Collectively, these results position HLA as a principled, scalable building block that combines attention-like, data-dependent mixing with the efficiency of modern recurrent architectures.

Project Page: https://github.com/yifanzhang-pro/HLA

1 Introduction

The Transformer architecture (Vaswani et al., 2017), powered by scaled dot-product attention, underpins modern large language models (LLMs). Yet the $O(n^2)$ computational and memory complexity in sequence length n constrains long-context use. A rich line of work therefore explores more efficient attention mechanisms (e.g., Linear Attention (Katharopoulos et al., 2020; Wang et al., 2020; Choromanski et al., 2020; Schlag et al., 2021; Sun et al., 2023; Qin et al., 2023; Yang et al., 2023; Qin et al., 2024; Yang et al., 2024b; von Oswald et al., 2025)), Modern Recurrent Neural Networks (RNNs) (Peng et al., 2024; Sun et al., 2024; Peng et al., 2025), Fast Weight Programmers (Delta Networks, Schlag et al. (2021)), State Space Models (SSMs) (Gu et al., 2021; Gu and Dao, 2023; Dao and Gu, 2024) and Memory Networks (Behrouz et al., 2024, 2025a,b), which admit $O(1)$ per-token state updates at inference.

We propose **Higher-order Linear Attention** (**HLA**), generalizing linear attention by incorporating higher interactions through compact prefix summaries (sufficient statistics). The key observation is that higher attention-like operators admit factorized forms in terms of low-order moments (e.g., sums of key outer products), enabling exact causal streaming without constructing attention matrices. In the second-order case, HLA maintains an constant-size state per head and produces outputs in

linear time per token ($O(d^2+d\,d_v)$, here d is the query/key dimension and d_v the value dimension, independent to the sequence length).

We address two central challenges: (i) enforcing strict autoregressive causality at second-order without sacrificing streaming updates or introducing any $n{\times}n$ intermediates; and (ii) enabling chunk-parallel training that exactly matches the activations of a serial recurrence. First, we derive an exact masked formulation that enforces strict autoregressive causality by augmenting the state with two additional summaries; the resulting algorithm remains streaming and efficient. Second, we present a chunk-parallel training scheme based on an associative (monoid/semidirect-product) operator that yields the same activations as a serial loop while exploiting intra- and inter-chunk parallelism.

Our contributions are summarized as follows:

1. **Exact masked streaming at second order.** We give a complete algebra of extended summaries that yields strictly causal second-order HLA with per-token constant cost, together with formal statements and proofs establishing masked streaming identities and online updates. The unnormalized HLA is the default operator; the ratio-normalized variant is an option built from the same summaries.

2. **Associative scans that match serial activations.** We define an associative (semidirect-product) operator for unmasked and masked settings (with and without exponential decay) and prove that a standard exclusive scan produces forward activations identical to those of a serial recurrence. We also state the reverse-mode algebra.

3. **Third-order extension.** We present the full masked third-order state and online updates, with a strictly causal streaming kernel.

HLA is intended as a drop-in, attention-like mixer for long-context models. It provides (i) attention-style, data-dependent weighting; (ii) strictly causal streaming with $O(1)$ per-token update memory independent of sequence length; and (iii) parallel training via scans without resorting to approximate backpropagation through time. We deliberately focus on algorithmic structure and implementation.

2   Background

Notations. We use bold lowercase for vectors and bold uppercase for matrices/tensors. Token index t denotes the current time; d is the query/key dimension; d_v is the value dimension. Unless otherwise stated, HLA outputs are in the default unnormalized form, which avoids length-dependent renormalization. We adopt row-vector outputs $\mathbf{o}_t \in \mathbb{R}^{1\times d_v}$; a ratio-normalized (row-normalized) variant divides by a masked scalar denominator built from the same summaries for scale control and comparability with linear attention. Throughout, *prefix summaries* are statistics computable in streaming fashion with $O(1)$ memory per token and per head.

2.1   Scaled dot-product attention

Given queries $\mathbf{Q} \in \mathbb{R}^{n\times d}$, keys $\mathbf{K} \in \mathbb{R}^{n\times d}$, and values $\mathbf{V} \in \mathbb{R}^{n\times d_v}$, scaled dot-product attention (Vaswani et al., 2017) is

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}} + \mathbf{\Lambda}\right)\mathbf{V},$$

where $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is the additive causal mask (zeros on and below the diagonal; $-\infty$ above). For algebraic manipulations outside the softmax (e.g., Section 3.1), we use the Hadamard product $\odot$ with a *binary* causal mask, denoted by $\mathbf{L}$ (ones on and below the diagonal; zeros above), to mask bilinear forms consistently.

2.2 Linear attention

Linear attentions (Wang et al., 2020; Katharopoulos et al., 2020; Choromanski et al., 2020) approximate the softmax kernel by a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^r$ (maybe unnormalized):

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})_i \approx \frac{\phi(\mathbf{q}_i)^\top \left(\sum_j \phi(\mathbf{k}_j) \mathbf{v}_j^\top \right)}{\phi(\mathbf{q}_i)^\top \left(\sum_j \phi(\mathbf{k}_j) \right)}.$$

Maintaining the running sums $\sum_j \phi(\mathbf{k}_j) \mathbf{v}_j^\top$ and $\sum_j \phi(\mathbf{k}_j)$ yields $O(n\,r(d + d_v))$ time and $O(r\,d_v)$ memory complexity.

3 Higher-order Linear Attention

In this section, we will introduce Higher-order Linear Attention (**HLA**). We begin with second-order linear attention as a warm-up, and present its extension to third-order linear attention in Section 7.

Second-order tensor attention mechanism. Second-order tensor attention can be written as

$$\mathbf{T}_2 := (\mathbf{Q}\mathbf{K}^\top)(\mathbf{Q}\mathbf{K}^\top)^\top = \mathbf{Q}(\mathbf{K}^\top \mathbf{K})\mathbf{Q}^\top \in \mathbb{R}^{n \times n},$$

so that $[\mathbf{T}_2]_{ij} = \mathbf{q}_i^\top (\mathbf{K}^\top \mathbf{K}) \mathbf{q}_j$. The right-hand side shows a dependence on the second moment $\mathbf{K}^\top \mathbf{K} \in \mathbb{R}^{d \times d}$, suggesting streaming implementations via prefix moments.

We maintain prefix summaries at time t:

$$\mathbf{S}_t^K := \sum_{i \leq t} \mathbf{k}_i \mathbf{k}_i^\top \in \mathbb{R}^{d \times d},$$

$$\mathbf{C}_t^{QV} := \sum_{i \leq t} \mathbf{q}_i \mathbf{v}_i^\top \in \mathbb{R}^{d \times d_v},$$

$$\mathbf{m}_t^Q := \sum_{i \leq t} \mathbf{q}_i \in \mathbb{R}^d.$$

All the above prefix summaries can be updated in a streaming fashion. In particular, the updates of $\mathbf{S}_t^K$ and $\mathbf{C}_t^{QV}$ cost $O(d^2)$ and $O(d\,d_v)$ time per token, respectively.

Unnormalized HLA. The output of second-order HLA at time t is the numerator-style bilinear form built from prefix moments:

$$\mathbf{o}_t := \mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{C}_t^{QV}. \tag{3.1}$$

This choice avoids length-dependent renormalization while preserving streaming updates and the same state as the normalized variant.

Normalized HLA. In order to define the normalized output of HLA, we define the numerator and denominator at t as follows:

$$\text{num}_t = \mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{C}_t^{QV}, \qquad \text{den}_t = \mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{m}_t^Q,$$

and the normalized output of HLA is given by

$$\mathbf{o}_t = \frac{\text{num}_t}{\text{den}_t + \varepsilon} = \frac{\mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{C}_t^{QV}}{\mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{m}_t^Q + \varepsilon}, \tag{3.2}$$

where $\varepsilon > 0$ is a small constant added for numerical stability.

Notably, $\mathbf{S}_t^K$ acts as a learned, data-dependent metric on query space; $\mathbf{C}_t^{QV}$ is a value accumulator modulated by past queries; and $\mathbf{m}_t^Q$ provides a query mass for optional scale control. This mirrors a second-order polynomial kernel in $(\mathbf{q}, \mathbf{k})$ while remaining strictly streaming and causal once masked (Section 3.1).

Connection with linear attention. Setting $\mathbf{S}_t^K = \mathbf{I}$ yields

$$\text{num}_t = \mathbf{q}_t^\top \mathbf{C}_t^{QV} = \sum_{i \leq t} (\mathbf{q}_t^\top \mathbf{q}_i) \mathbf{v}_i^\top, \qquad \text{den}_t = \mathbf{q}_t^\top \mathbf{m}_t^Q = \sum_{i \leq t} \mathbf{q}_t^\top \mathbf{q}_i,$$

So the normalized output reduces to a linear-attention form with kernel $K(\mathbf{q}_t, \mathbf{q}_i) = \mathbf{q}_t^\top \mathbf{q}_i$. When queries and keys are tied ($\mathbf{q}_i \equiv \mathbf{k}_i$), this coincides with linear attention using the identity feature map $\phi(x) = x$. In general, second-order HLA implements the *data-adaptive* degree-2 polynomial kernel $K_t(\mathbf{q}, \mathbf{q}') = \mathbf{q}^\top \mathbf{S}_t^K \mathbf{q}'$ whose metric $\mathbf{S}_t^K = \sum_{i \leq t} \mathbf{k}_i \mathbf{k}_i^\top$ depends on the past keys, strictly enriching first-order linearizations while retaining streaming. Absent tying $\mathbf{q} \equiv \mathbf{k}$, this differs from identity-feature linear attention.

3.1 Causal masking via extended summaries

Let $\mathbf{L}$ denote the binary causal mask (lower-triangular, including the diagonal). For the masked second-order matrix,

$$\left[(\mathbf{L} \odot \mathbf{QK}^\top)(\mathbf{L} \odot \mathbf{QK}^\top)^\top \right]_{t,j} = \sum_{i \leq \min(t,j)} (\mathbf{q}_t^\top \mathbf{k}_i)(\mathbf{q}_j^\top \mathbf{k}_i) = \mathbf{q}_t^\top \mathbf{S}_{\min(t,j)}^K \mathbf{q}_j.$$

Equivalently, the strictly causal second-order output at time t can be written in matrix form by masking on the right before applying values:

$$\mathbf{o}_t = \left(\left[(\mathbf{L} \odot \mathbf{QK}^\top)(\mathbf{L} \odot \mathbf{QK}^\top)^\top \right] \odot \mathbf{L} \right)_{t,:} \mathbf{V}.$$

This row-wise $\odot \mathbf{L}$ enforces the restriction $j \leq t$ when multiplying by $\mathbf{V}$.

Define two additional prefix summaries

$$\mathbf{G}_t := \sum_{i \leq t} \left(\mathbf{k}_i \mathbf{k}_i^\top \right) \mathbf{C}_{i-1}^{QV} \in \mathbb{R}^{d \times d_v},$$

$$\mathbf{h}_t := \sum_{i \leq t} \left(\mathbf{k}_i \mathbf{k}_i^\top \right) \mathbf{m}_{i-1}^Q \in \mathbb{R}^d.$$

We have the following theorem, which gives the unnormalized and normalized outputs of HLA with a causal mask.

4

Theorem 3.1 (Masked streaming identity for second order). For each t, let

$$\text{num}_t^{\text{mask}} = \mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{C}_t^{QV} - \mathbf{G}_t \right), \qquad \text{den}_t^{\text{mask}} = \mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{m}_t^Q - \mathbf{h}_t \right).$$

Consequently, the strictly causal, masked default unnormalized output is

$$\mathbf{o}_t = \mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{C}_t^{QV} - \mathbf{G}_t \right). \tag{3.3}$$

An optional linear normalization divides by the masked denominator,

$$\mathbf{o}_t = \frac{\mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{C}_t^{QV} - \mathbf{G}_t \right)}{\mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{m}_t^Q - \mathbf{h}_t \right) + \varepsilon}, \tag{3.4}$$

where $\varepsilon > 0$ is a small constant added for numerical stability.

Proof. Let $\mathbf{W} = \mathbf{L} \odot (\mathbf{Q}\mathbf{K}^\top)$ with $\mathbf{L}$ lower-triangular including the diagonal. For the second-order weight matrix, we have $\mathbf{W}\mathbf{W}^\top$ with entries $(\mathbf{W}\mathbf{W}^\top)_{t,j} = \sum_{i \leq \min(t,j)} (\mathbf{q}_t^\top \mathbf{k}_i)(\mathbf{q}_j^\top \mathbf{k}_i)$. Then the masked, unnormalized numerator at time t is

$$\text{num}_t^{\text{mask}} = \sum_{j \leq t} (\mathbf{W}\mathbf{W}^\top)_{t,j} \, \mathbf{v}_j^\top = \sum_{j \leq t} \Big(\sum_{i \leq j} \mathbf{q}_t^\top \mathbf{k}_i \, \mathbf{k}_i^\top \mathbf{q}_j \Big) \mathbf{v}_j^\top = \mathbf{q}_t^\top \sum_{j \leq t} \Big(\sum_{i \leq j} \mathbf{k}_i \mathbf{k}_i^\top \Big) \mathbf{q}_j \mathbf{v}_j^\top,$$

where the second equality uses the fact that $\min(t,j) = j$ when $j \leq t$. Interchanging finite sums yields

$$\sum_{j \leq t} \Big(\sum_{i \leq j} \mathbf{k}_i \mathbf{k}_i^\top \Big) \mathbf{q}_j \mathbf{v}_j^\top = \sum_{j \leq t} \mathbf{S}_j^K \, \mathbf{q}_j \mathbf{v}_j^\top = \underbrace{\sum_{j \leq t} \mathbf{S}_t^K \, \mathbf{q}_j \mathbf{v}_j^\top}_{I_1} - \underbrace{\sum_{j \leq t} \Big(\sum_{j < i \leq t} \mathbf{k}_i \mathbf{k}_i^\top \Big) \mathbf{q}_j \mathbf{v}_j^\top}_{I_2}, \tag{3.5}$$

where the last equality holds due to $\mathbf{S}_j^K = \mathbf{S}_t^K - \sum_{j < i \leq t} \mathbf{k}_i \mathbf{k}_i^\top$.

In Eq. (3.5), the first term I_1 equals $\mathbf{S}_t^K \mathbf{C}_t^{QV}$. For the second term I_2, swap the order of summation: $\sum_{j \leq t} \sum_{i > j}(\cdot) = \sum_{i \leq t} \sum_{j < i}(\cdot)$, we can obtain $I_2 = \sum_{i \leq t} \big(\mathbf{k}_i \mathbf{k}_i^\top \big)\big(\sum_{j < i} \mathbf{q}_j \mathbf{v}_j^\top \big) = \mathbf{G}_t$. This proves the numerator identity. The proof for the denominator is analogous with $\mathbf{v}_j$ replaced by 1 (i.e., $\mathbf{q}_j$ replaced by 1-summaries), yielding $\mathbf{S}_t^K \mathbf{m}_t^Q - \mathbf{h}_t$. Finally, the division by $\text{den}_t^{\text{mask}} + \varepsilon$ gives Eq. (3.4). $\qquad\square$

Online updates. Using the fact that $(\mathbf{k}\mathbf{k}^\top)X = \mathbf{k}(\mathbf{k}^\top X)$, we have

$$\mathbf{S}_t^K = \mathbf{S}_{t-1}^K + \mathbf{k}_t \mathbf{k}_t^\top, \qquad \mathbf{C}_t^{QV} = \mathbf{C}_{t-1}^{QV} + \mathbf{q}_t \mathbf{v}_t^\top, \qquad \mathbf{m}_t^Q = \mathbf{m}_{t-1}^Q + \mathbf{q}_t,$$

$$\mathbf{G}_t = \mathbf{G}_{t-1} + \mathbf{k}_t(\mathbf{k}_t^\top \mathbf{C}_{t-1}^{QV}), \qquad \mathbf{h}_t = \mathbf{h}_{t-1} + \mathbf{k}_t(\mathbf{k}_t^\top \mathbf{m}_{t-1}^Q).$$

Therefore, the per-token cost remains $O(d^2 + d\,d_v)$ in total.

4 Chunk-parallel training via associative scans

In Section 3.1, we have presented the recurrent form for second-order HLA. As we know, training a purely recurrent model is inefficient on GPUs. We adopt within-chunk scans with width w and inter-chunk scans across B chunks (Blelloch, 1990). A similar technique has been widely used in the literature of linear attention (Yang et al., 2023; Qin et al., 2024). We write B_c for the number of chunks to avoid overloading B elsewhere; thus, inter-chunk scans are across B_c chunks.

4.1 Unmasked monoid

Let $\mathcal{S} = (\mathbf{S}, \mathbf{C}, \mathbf{m})$ with token "deltas" $\Delta\mathbf{S}_t = \mathbf{k}_t\mathbf{k}_t^\top$, $\Delta\mathbf{C}_t = \mathbf{q}_t\mathbf{v}_t^\top$, $\Delta\mathbf{m}_t = \mathbf{q}_t$. Define elementary segments $\mathcal{T}_t = (\Delta\mathbf{S}_t, \Delta\mathbf{C}_t, \Delta\mathbf{m}_t)$ and the additive monoid

$$(\mathbf{S}_A, \mathbf{C}_A, \mathbf{m}_A) \oplus (\mathbf{S}_B, \mathbf{C}_B, \mathbf{m}_B) = (\mathbf{S}_A + \mathbf{S}_B, \ \mathbf{C}_A + \mathbf{C}_B, \ \mathbf{m}_A + \mathbf{m}_B).$$

An exclusive Blelloch scan on $\{\mathcal{T}_1, \ldots, \mathcal{T}_w\}$ yields per-token prefixes $\mathcal{P}_t = \bigoplus_{i<t} \mathcal{T}_i$, from which the inclusive state at t is obtained locally by adding $\mathcal{T}_t$. Here and below, A then B denotes adjacent segments in time (all indices in A precede those in B).

4.2 Masked semidirect product

For the masked case use $\mathcal{S} = (\mathbf{S}, \mathbf{C}, \mathbf{m}, \mathbf{G}, \mathbf{h})$. For a single-token segment, $\mathbf{G} = \mathbf{h} = \mathbf{0}$. Concatenation is

$$\begin{aligned}
(\mathbf{S}_A, \mathbf{C}_A, \mathbf{m}_A, \mathbf{G}_A, \mathbf{h}_A) \oplus (\mathbf{S}_B, \mathbf{C}_B, \mathbf{m}_B, \mathbf{G}_B, \mathbf{h}_B) = \\
(\mathbf{S}_A + \mathbf{S}_B, \ \mathbf{C}_A + \mathbf{C}_B, \ \mathbf{m}_A + \mathbf{m}_B, \ \mathbf{G}_A + \mathbf{G}_B + \mathbf{S}_B\mathbf{C}_A, \ \mathbf{h}_A + \mathbf{h}_B + \mathbf{S}_B\mathbf{m}_A),
\end{aligned} \tag{4.1}$$

which is associative (direct expansion). Perform the same exclusive scan; per-token inclusive states follow by adding the local deltas and the cross-terms $\Delta\mathbf{S}_t\mathbf{C}_{t-1}$ and $\Delta\mathbf{S}_t\mathbf{m}_{t-1}$.

Decay-aware monoid. Let $\gamma \in (0, 1)$ be a fixed exponential decay and let a segment $\mathcal{X}$ carry its *length* $\ell(\mathcal{X})$ and attenuation $\rho(\mathcal{X}) := \gamma^{\ell(\mathcal{X})}$. For the unmasked triple $\mathcal{S} = (\mathbf{S}, \mathbf{C}, \mathbf{m})$ the decayed concatenation is

$$(\mathbf{S}_A, \mathbf{C}_A, \mathbf{m}_A, \rho_A) \oplus_\gamma (\mathbf{S}_B, \mathbf{C}_B, \mathbf{m}_B, \rho_B) = \big(\rho_B\mathbf{S}_A + \mathbf{S}_B, \ \rho_B\mathbf{C}_A + \mathbf{C}_B, \ \rho_B\mathbf{m}_A + \mathbf{m}_B, \ \rho_A\rho_B\big),$$

and analogously for the masked $(\mathbf{S}, \mathbf{C}, \mathbf{m}, \mathbf{G}, \mathbf{h})$ state:

$$\begin{aligned}
(\mathbf{S}_A, \mathbf{C}_A, \mathbf{m}_A, \mathbf{G}_A, \mathbf{h}_A, \rho_A) \oplus_\gamma (\mathbf{S}_B, \mathbf{C}_B, \mathbf{m}_B, \mathbf{G}_B, \mathbf{h}_B, \rho_B) = \\
\Big(\rho_B\mathbf{S}_A + \mathbf{S}_B, \ \rho_B\mathbf{C}_A + \mathbf{C}_B, \ \rho_B\mathbf{m}_A + \mathbf{m}_B, \\
\rho_B\mathbf{G}_A + \mathbf{G}_B + \mathbf{S}_B(\rho_B\mathbf{C}_A), \ \rho_B\mathbf{h}_A + \mathbf{h}_B + \mathbf{S}_B(\rho_B\mathbf{m}_A), \ \rho_A\rho_B\Big).
\end{aligned}$$

Associativity follows from bilinearity and ρ-multiplicativity.

Theorem 4.1 (Scan equivalence: serial vs. (decayed) associative scans)**.** Consider a sequence of token segments $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ and either $\oplus$ (no decay) or $\oplus_\gamma$ (with decay). Let $\mathcal{P}_t$ be the exclusive prefix obtained by a Blelloch scan under the chosen operator. For each t, the inclusive state computed locally from $\mathcal{P}_t$ and $\mathcal{T}_t$ equals the state produced by a serial left-to-right recurrence on tokens $1{:}t$. Consequently, the per-token masked outputs are identical to those of the serial algorithm.

Proof. We prove for the masked, decayed case; the other cases are specializations. Define the serial recurrence $\mathcal{X}_t = \Phi_\gamma(\mathcal{X}_{t-1}, \mathcal{T}_t)$ given by the online updates in Section 3.1 with decay γ. By construction, Φ_γ coincides with the binary map $f_\gamma(\mathcal{X}, \mathcal{Y}) := \mathcal{X} \oplus_\gamma \mathcal{Y}$ when $\mathcal{Y}$ is a single-token segment. Because $\oplus_\gamma$ is associative with identity the zero-length segment $\mathcal{E}$ (all-zero summaries, $\rho = 1$), the Blelloch scan yields $\mathcal{P}_t = \mathcal{E} \oplus_\gamma \mathcal{T}_1 \oplus_\gamma \cdots \oplus_\gamma \mathcal{T}_{t-1}$. The local inclusive update computes $\mathcal{P}_t \oplus_\gamma \mathcal{T}_t$, which equals $\mathcal{X}_t$ by associativity and the definition of Φ_γ. The masked outputs are functions only of the inclusive state (Theorem 3.1), hence coincide with the serial outputs. $\square$

Backward for gradients. Let $\oplus^*$ denote the vector-Jacobian adjoint of $\oplus$ evaluated at the forward states. A reverse (decayed) scan applying $\oplus_\gamma^*$ with checkpointing at tile boundaries yields gradients that match those of the serial recurrence, by Theorem 4.1 and the chain rule.

Remark 4.2 (Inclusive vs. exclusive scans). Given segments $(\mathcal{T}_1, \ldots, \mathcal{T}_w)$ and an associative operator $\oplus$ with identity $\mathcal{E}$, the *exclusive* scan returns prefixes $\mathcal{P}_t = \mathcal{E} \oplus \mathcal{T}_1 \oplus \cdots \oplus \mathcal{T}_{t-1}$, while the *inclusive* scan returns $\mathcal{I}_t = \mathcal{P}_t \oplus \mathcal{T}_t$. Our forward algorithms compute $\mathcal{P}_t$ via an exclusive Blelloch scan and then form the inclusive state locally by combining $\mathcal{P}_t$ with the token's deltas (and required cross-terms). This choice exposes maximal parallelism and ensures exact equality to the serial recurrence by Theorem 4.1. With decay, the identity is the zero-length segment $(\mathbf{0}, \ldots, \mathbf{0}, \rho{=}1)$; the exclusive/inclusive distinction is unchanged.

Intra-chunk parallelism. Within a chunk of width w, an exclusive Blelloch scan over $\{\mathcal{T}_1, \ldots, \mathcal{T}_w\}$ under $\oplus$ (or $\oplus_\gamma$) yields $\mathcal{P}_t$ for all t in $O(\log w)$ span and $O(1)$ auxiliary memory per position. The per-token inclusive states are then computed independently as $\mathcal{I}_t = \mathcal{P}_t \oplus \mathcal{T}_t$.

Inter-chunk parallelism. For B_c chunks, each chunk c produces a single *summary* $\mathcal{S}^{(c)} = \bigoplus_{t \in \text{chunk } c} \mathcal{T}_t$. An exclusive scan across the B_c summaries gives carry-in prefixes $\widehat{\mathcal{P}}^{(c)}$ for every chunk. Each position t in chunk c then uses the merged prefix $\widehat{\mathcal{P}}^{(c)} \oplus \mathcal{P}_t$ before adding its local $\mathcal{T}_t$ to obtain the inclusive state. This is the same parallel skeleton widely used in modern linear-attention and recurrent networks that maintain streaming sufficient statistics (Sun et al., 2023; Qin et al., 2023; Yang et al., 2023; Qin et al., 2024; Yang et al., 2024b).

Connection to linear attention. First-order linear attentions and related modern RNN kernels scan additive/decayed summaries (e.g., $\sum \phi(\mathbf{k})\mathbf{v}^\top$ and denominators) using exactly this intra-/inter-chunk pattern. **HLA** plugs into the same infrastructure: only the state tuple and cross-terms change (e.g., $(\mathbf{S}, \mathbf{C}, \mathbf{m}, \mathbf{G}, \mathbf{h})$ for second order), while the exclusive/inclusive logic and two-level scan strategy remain identical. Thus, HLA inherits the throughput characteristics of these systems with strictly higher expressivity.

4.3 Adding decay and regularization

Decayed states. Introduce a time decay $\gamma \in (0, 1)$:

$$\mathbf{S}_t^K = \gamma \mathbf{S}_{t-1}^K + \mathbf{k}_t \mathbf{k}_t^\top, \quad \mathbf{C}_t^{QV} = \gamma \mathbf{C}_{t-1}^{QV} + \mathbf{q}_t \mathbf{v}_t^\top, \quad \mathbf{m}_t^Q = \gamma \mathbf{m}_{t-1}^Q + \mathbf{q}_t,$$

and the cross-summaries obey

$$\mathbf{G}_t = \gamma \mathbf{G}_{t-1} + \mathbf{k}_t \big(\mathbf{k}_t^\top \mathbf{C}_{t-1}^{QV} \big), \qquad \mathbf{h}_t = \gamma \mathbf{h}_{t-1} + \mathbf{k}_t \big(\mathbf{k}_t^\top \mathbf{m}_{t-1}^Q \big),$$

which are the decayed analogues of the online updates in Section 3.1. Decay controls spectral growth and improves recency bias while maintaining associativity (with respect to segment-local normalization) (Sun et al., 2023; Qin et al., 2023; Yang et al., 2023, 2024a; Peng et al., 2024; Behrouz et al., 2024; Peng et al., 2025; Behrouz et al., 2025a,b).

5 Implementation details and complexity

In this section, we discuss the implementation details and provide a complexity analysis.

Recall that for each token and each head (second order), we have

- **State:** $\mathbf{S}_t^K \in \mathbb{R}^{d \times d}$, $\mathbf{C}_t^{QV} \in \mathbb{R}^{d \times d_v}$, $\mathbf{m}_t^Q \in \mathbb{R}^d$ (and masked $\mathbf{G}_t \in \mathbb{R}^{d \times d_v}$, $\mathbf{h}_t \in \mathbb{R}^d$).

- **Compute:** evaluate $\mathbf{u}_t = \mathbf{q}_t^\top \mathbf{S}_t^K$ (mat–vec) and then $\mathbf{u}_t \mathbf{C}_t^{QV}$ (row–matrix), with masked corrections $-\mathbf{q}_t^\top \mathbf{G}_t$; the denominator uses $\mathbf{u}_t \mathbf{m}_t^Q - \mathbf{q}_t^\top \mathbf{h}_t$. This avoids forming $\mathbf{S}_t^K \mathbf{C}_t^{QV}$ explicitly; masked cross-terms still use $\mathbf{k}_t^\top X$ to avoid cubic cost.

- **Parallelism:** within-chunk Blelloch scans (span $O(\log w)$) and inter-chunk exclusive scans across B_c chunks, both using the same $\oplus$.

Algorithm 1 Masked (Second Order) HLA with Within-Chunk Scan

Require: Chunk tokens $(\mathbf{q}[1{:}w], \mathbf{k}[1{:}w], \mathbf{v}[1{:}w])$, ε, optional ridge λ, optional decay γ, optional flag `normalize`
1: **Token segments:** for $t = 1..w$, set $\Delta\mathbf{S}_t \leftarrow \mathbf{k}_t \mathbf{k}_t^\top$, $\Delta\mathbf{C}_t \leftarrow \mathbf{q}_t \mathbf{v}_t^\top$, $\Delta\mathbf{m}_t \leftarrow \mathbf{q}_t$, and initialize $\mathbf{G}_t{=}\mathbf{0}$, $\mathbf{h}_t{=}\mathbf{0}$.
2: **Exclusive scan** over $\{(\Delta\mathbf{S}_t, \Delta\mathbf{C}_t, \Delta\mathbf{m}_t, \mathbf{0}, \mathbf{0})\}_{t=1}^w$ using $\oplus$ in Eq. (4.1) (with decay if used) to obtain prefixes $\mathcal{P}_t = (\mathbf{S}_{t-1}, \mathbf{C}_{t-1}, \mathbf{m}_{t-1}, \mathbf{G}_{t-1}, \mathbf{h}_{t-1})$.
3: **for** $t = 1$ to w **in parallel do**
4:     **Inclusive state:**
5:     $\mathbf{S}_t \leftarrow \gamma\mathbf{S}_{t-1} + \Delta\mathbf{S}_t$; $\mathbf{C}_t \leftarrow \gamma\mathbf{C}_{t-1} + \Delta\mathbf{C}_t$; $\mathbf{m}_t \leftarrow \gamma\mathbf{m}_{t-1} + \Delta\mathbf{m}_t$
6:     $\mathbf{G}_t \leftarrow \gamma\mathbf{G}_{t-1} + \Delta\mathbf{S}_t \mathbf{C}_{t-1}$; $\mathbf{h}_t \leftarrow \gamma\mathbf{h}_{t-1} + \Delta\mathbf{S}_t \mathbf{m}_{t-1}$
7:     **Effective S:** $\mathbf{S}_t^{\text{eff}} \leftarrow \mathbf{S}_t + \lambda\mathbf{I}$ ▷ optional ridge for stability
8:     **Default masked unnormalized output:**
9:     $\mathbf{u} \leftarrow \mathbf{q}_t^\top \mathbf{S}_t^{\text{eff}}$ ▷ $O(d^2)$ matvec
10:     $\text{num} \leftarrow \mathbf{u}\,\mathbf{C}_t - \mathbf{q}_t^\top \mathbf{G}_t$ ▷ $O(d\,d_v)$
11:     $\mathbf{o}_t^{\text{hla}} \leftarrow \text{num}$
12:     **Optional normalization:**
13:     **if** `normalize` **then**
14:         $\text{den} \leftarrow \mathbf{u}\,\mathbf{m}_t - \mathbf{q}_t^\top \mathbf{h}_t + \varepsilon$
15:         $\mathbf{o}_t^{\text{hla}} \leftarrow \mathbf{o}_t^{\text{hla}}/\text{den}$
16:     **end if**
17: **end for**
18: **return** $\{\mathbf{o}_t^{\text{hla}}\}_{t=1}^w$

5.1 Pseudocode

We present a PyTorch-like reference for masked second-order HLA with a within-chunk exclusive scan. Unmasked and/or diagonal-regularized variants follow by removing $(\mathbf{G}, \mathbf{h})$. Normalization is optional; by default, the implementation returns the unnormalized output and may divide by the masked denominator if requested.

Remark. Adding $\lambda\mathbf{I}$ yields a stabilized causal variant of the masked operator; it does not correspond to the exact masked bilinear form of $(\mathbf{L} \odot \mathbf{Q}\mathbf{K}^\top)$.

5.2 Implementation considerations

HLA only replaces the standard attention sublayer in the transformer block, while the feed-forward sublayer and normalization sublayers remain unchanged. Drop-in replacement requires only swapping the kernel while keeping positional encodings and masking identical to the baseline. Multi-query keys/values (sharing $\mathbf{K}, \mathbf{V}$ across heads) reduce state from $O(h\,d^2)$ to $O(d^2 + h\,d\,d_v)$ while preserving the algebra.

The summaries $(\mathbf{S}, \mathbf{C}, \mathbf{m}, \mathbf{G}, \mathbf{h})$ are per head. With *multi-query* ($\mathbf{K}, \mathbf{V}$ shared across heads), the key moment $\mathbf{S}_t^K$ is shared and stored once per layer ($O(d^2)$), while $(\mathbf{C}_t^{QV}, \mathbf{m}_t^Q, \mathbf{G}_t, \mathbf{h}_t)$ remain per-head ($O(h\,d\,d_v + h\,d)$). This yields a total memory of $O(d^2 + h\,d\,d_v)$ instead of $O(h\,d^2 + h\,d\,d_v)$ when each head maintains its own $\mathbf{S}_t^K$.

For throughput, maintain $\mathbf{S}_t^K$ in a packed symmetric layout (store only the upper triangle, $\frac{1}{2}d(d+1)$ entries) to reduce bandwidth without changing the algebra. Within a chunk of width w, use an exclusive Blelloch scan to obtain prefixes in $O(\log w)$ span and constant extra memory per position; inter-chunk scans use the same operator across B_c chunks.

6 Asymmetric Higher-order Linear Attention

Motivation. The second-order HLA in Section 3 realizes the symmetric triple product $\mathbf{A}\mathbf{A}^\top\mathbf{V}$ with $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$ (masked later). We introduce a complementary *asymmetric* variant that uses the left-cascaded product

$$\underbrace{\mathbf{A}\mathbf{A}\mathbf{V}}_{\text{AAV}} = \mathbf{Q}\,(\mathbf{K}^\top\mathbf{Q})\,(\mathbf{K}^\top\mathbf{V}),$$

and show it admits strictly causal streaming with $O(d^2 + d\,d_v)$ per-token cost. We call this operator **AHLA** (*Asymmetric Higher-order Linear Attention*).

6.1 Definition and masked streaming identity

Let $\mathbf{A} = \mathbf{L} \odot (\mathbf{Q}\mathbf{K}^\top)$ be the causally masked affinity, where $\mathbf{L}$ is the binary lower-triangular mask (including the diagonal). The AAV weights are

$$(\mathbf{A}\mathbf{A})_{t,j} = \sum_{i=j}^{t} (\mathbf{q}_t^\top \mathbf{k}_i)\,(\mathbf{q}_i^\top \mathbf{k}_j), \qquad j \le t.$$

To obtain strictly causal outputs when applying to values, interpret the final multiplication as $\big((\mathbf{A}\mathbf{A}) \odot \mathbf{L}\big)\mathbf{V}$; the streaming identity in Theorem 6.1 implements exactly this row-wise masking.

Consequently, the (unnormalized) output is

$$\mathbf{o}_t^{\text{AHLA}} = \sum_{j \le t} \sum_{i=j}^{t} (\mathbf{q}_t^\top \mathbf{k}_i)(\mathbf{q}_i^\top \mathbf{k}_j)\,\mathbf{v}_j^\top. \tag{6.1}$$

Introduce the streaming prefix summaries

$$\mathbf{P}_t^{KV} := \sum_{j \le t} \mathbf{k}_j \mathbf{v}_j^\top \in \mathbb{R}^{d \times d_v}, \qquad\qquad \mathbf{m}_t^K := \sum_{j \le t} \mathbf{k}_j \in \mathbb{R}^d,$$

$$\mathbf{E}_t := \sum_{i \le t} \mathbf{k}_i \big(\mathbf{q}_i^\top \mathbf{P}_i^{KV}\big) \in \mathbb{R}^{d \times d_v},$$

$$\mathbf{n}_t := \sum_{i \le t} \mathbf{k}_i \big(\mathbf{q}_i^\top \mathbf{m}_i^K\big) \in \mathbb{R}^d.$$

Note. For chunk-parallel scans used in training, we additionally introduce a segment-level cross moment $\mathbf{R}^{KQ}$; see Section 6.2 for its definition and role in the concatenation operator.

Theorem 6.1 (Masked streaming identity for AHLA). With the above definitions,

$$\mathbf{o}_t^{\text{AHLA}} \;=\; \mathbf{q}_t^\top \mathbf{E}_t \qquad \text{and} \qquad \widehat{\mathbf{o}}_t^{\text{AHLA}} \;=\; \frac{\mathbf{q}_t^\top \mathbf{E}_t}{\mathbf{q}_t^\top \mathbf{n}_t + \varepsilon},$$

where the second expression is an optional linear normalization using the masked denominator. The online (strictly causal) updates are

$$\mathbf{P}_t^{KV} = \mathbf{P}_{t-1}^{KV} + \mathbf{k}_t \mathbf{v}_t^\top, \qquad \mathbf{m}_t^K = \mathbf{m}_{t-1}^K + \mathbf{k}_t,$$
$$\mathbf{E}_t = \mathbf{E}_{t-1} + \mathbf{k}_t(\mathbf{q}_t^\top \mathbf{P}_t^{KV}), \qquad \mathbf{n}_t = \mathbf{n}_{t-1} + \mathbf{k}_t(\mathbf{q}_t^\top \mathbf{m}_t^K).$$

Proof. From Eq. (6.1), fix i and sum over $j \leq i$: $\sum_{j \leq i}(\mathbf{q}_i^\top \mathbf{k}_j)\,\mathbf{v}_j^\top = \mathbf{q}_i^\top \mathbf{P}_i^{KV}$. Then $\mathbf{o}_t^{\text{AHLA}} = \sum_{i \leq t}(\mathbf{q}_t^\top \mathbf{k}_i)(\mathbf{q}_i^\top \mathbf{P}_i^{KV}) = \mathbf{q}_t^\top\left(\sum_{i \leq t}\mathbf{k}_i(\mathbf{q}_i^\top \mathbf{P}_i^{KV})\right) = \mathbf{q}_t^\top \mathbf{E}_t$. Replacing $\mathbf{v}_j$ by 1 gives the denominator with $\mathbf{m}_i^K$, hence $\mathbf{n}_t$. The stated updates follow by isolating index $i{=}t$ and using that $\mathbf{P}_t^{KV} = \mathbf{P}_{t-1}^{KV} + \mathbf{k}_t\mathbf{v}_t^\top$ and $\mathbf{m}_t^K = \mathbf{m}_{t-1}^K + \mathbf{k}_t$. □

Cost. For *streaming/serial inference*, the dominant work is forming $\mathbf{q}_t^\top \mathbf{P}_t^{KV} \in \mathbb{R}^{1 \times d_v}$ and the outer product $\mathbf{k}_t(\cdot)$; the total is $O(d\,d_v)$ time and $O(d\,d_v{+}d)$ state per head (for $\mathbf{P}, \mathbf{E}, \mathbf{m}, \mathbf{n}$). For *chunk-parallel scans* used in training, an additional block statistic $\mathbf{R}^{KQ}$ appears only inside the concatenation operator (Section 6.2), contributing $O(d^2)$ memory per *chunk summary* but not to the streaming path.

Decay mechanism. With exponential decay $\gamma \in (0,1)$,

$$\mathbf{P}_t^{KV} = \gamma \mathbf{P}_{t-1}^{KV} + \mathbf{k}_t \mathbf{v}_t^\top, \qquad \mathbf{m}_t^K = \gamma \mathbf{m}_{t-1}^K + \mathbf{k}_t,$$
$$\mathbf{E}_t = \gamma \mathbf{E}_{t-1} + \mathbf{k}_t(\mathbf{q}_t^\top \mathbf{P}_t^{KV}), \qquad \mathbf{n}_t = \gamma \mathbf{n}_{t-1} + \mathbf{k}_t(\mathbf{q}_t^\top \mathbf{m}_t^K),$$

which preserves associativity of the scan operator below.

6.2 Chunk-parallel scans for AHLA

Unmasked/masked concatenation. For segment A followed by B, consider the augmented state

$$\mathcal{S} = (\mathbf{R}^{KQ}, \mathbf{P}^{KV}, \mathbf{m}^K, \mathbf{E}, \mathbf{n}).$$

Here $\mathbf{R}^{KQ}$ is the segment-level key–query cross moment, defined by

$$\mathbf{R}^{KQ} \;:=\; \sum_{i \in \text{segment}} \mathbf{k}_i \mathbf{q}_i^\top \;\in\; \mathbb{R}^{d \times d}.$$

It is used *only* during chunk concatenation to form the cross terms $\mathbf{R}_B \mathbf{P}_A$ and $\mathbf{R}_B \mathbf{m}_A$ in Eq. (6.2); It is *not* required by the serial/streaming forward path in Algorithm 2. With exponential decay, the segment's $\mathbf{R}^{KQ}$ attenuates as $\rho_B \mathbf{R}_A + \mathbf{R}_B$ in the decayed concatenation.

The undecayed associative concatenation is

$$(\mathbf{R}_A, \mathbf{P}_A, \mathbf{m}_A, \mathbf{E}_A, \mathbf{n}_A) \oplus_{\text{AHLA}} (\mathbf{R}_B, \mathbf{P}_B, \mathbf{m}_B, \mathbf{E}_B, \mathbf{n}_B) =$$
$$\left(\mathbf{R}_A{+}\mathbf{R}_B,\; \mathbf{P}_A{+}\mathbf{P}_B,\; \mathbf{m}_A{+}\mathbf{m}_B,\; \mathbf{E}_A{+}\mathbf{E}_B{+}\mathbf{R}_B\mathbf{P}_A,\; \mathbf{n}_A{+}\mathbf{n}_B{+}\mathbf{R}_B\mathbf{m}_A\right), \tag{6.2}$$

which is associative by direct expansion of $\sum_{i \in A \cup B} \mathbf{k}_i(\mathbf{q}_i^\top \mathbf{P}_{\leq i})$ and the observation that for $i \in B$ the missing cross-prefix equals $\mathbf{R}_B \mathbf{P}_A$ (and analogously for $\mathbf{n}$).

Decay-aware concatenation. Let each segment carry its attenuation $\rho(\cdot) = \gamma^{\ell(\cdot)}$. Then

$$(\mathbf{R}, \mathbf{P}, \mathbf{m}, \mathbf{E}, \mathbf{n}, \rho) = (\rho_B \mathbf{R}_A + \mathbf{R}_B, \; \rho_B \mathbf{P}_A + \mathbf{P}_B, \; \rho_B \mathbf{m}_A + \mathbf{m}_B,$$
$$\rho_B \mathbf{E}_A + \mathbf{E}_B + \mathbf{R}_B(\rho_B \mathbf{P}_A), \; \rho_B \mathbf{n}_A + \mathbf{n}_B + \mathbf{R}_B(\rho_B \mathbf{m}_A), \; \rho_A \rho_B),$$

which is associative by bilinearity and multiplicativity of ρ.

Scan equivalence. An exclusive Blelloch scan under $\oplus_{\mathrm{AHLA}}$ (or its decayed form) followed by local inclusion reproduces exactly the activations of the serial recurrence given above.

6.3 Pseudocode

Algorithm 2 AHLA (Second-order) streaming with causal mask and optional decay

---
Require: $\{\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t\}_{t=1}^n$, decay $\gamma \in (0, 1]$, stability $\varepsilon > 0$, flag `normalize`
1: **Init:** $\mathbf{P} = \mathbf{0}_{d \times d_v}$, $\mathbf{m} = \mathbf{0}_d$, $\mathbf{E} = \mathbf{0}_{d \times d_v}$, $\mathbf{n} = \mathbf{0}_d$
2: **for** $t = 1$ to n **do**
3:      $\mathbf{P} \leftarrow \gamma \mathbf{P} + \mathbf{k}_t \mathbf{v}_t^\top$; $\mathbf{m} \leftarrow \gamma \mathbf{m} + \mathbf{k}_t$
4:      $r \leftarrow \mathbf{q}_t^\top \mathbf{P}$ $\triangleright \; 1 \times d_v$
5:      $s \leftarrow \mathbf{q}_t^\top \mathbf{m}$ $\triangleright \; \text{scalar}$
6:      $\mathbf{E} \leftarrow \gamma \mathbf{E} + \mathbf{k}_t \, r$; $\mathbf{n} \leftarrow \gamma \mathbf{n} + s \, \mathbf{k}_t$
7:      $\mathbf{o}_t \leftarrow \mathbf{q}_t^\top \mathbf{E}$
8:      **if** `normalize` **then**
9:          $\mathrm{den} \leftarrow \mathbf{q}_t^\top \mathbf{n} + \varepsilon$; $\mathbf{o}_t \leftarrow \mathbf{o}_t / \mathrm{den}$
10:     **end if**
11: **end for**
12: **return** $\{\mathbf{o}_t\}_{t=1}^n$

Relation to $\mathbf{A}\mathbf{A}^\top \mathbf{V}$. AHLA emphasizes a matrix power of $\mathbf{A}$, weighting each value $\mathbf{v}_j$ through a single pass $\mathbf{q}_i^\top \mathbf{k}_j$ routed by an intermediate key index i. In contrast, the symmetric $\mathbf{A}\mathbf{A}^\top \mathbf{V}$ aggregates via the metric $\mathbf{S}^K$ and query summaries. Both are second-order, strictly causal, and stream with identical asymptotic costs but induce different inductive biases.

7 Third-Order Linear Attention

In this section, we will introduce third-order HLA.

7.1 Steaming form of Causal HLA

Third-order tensor attention mechanism. Let $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{n \times n}$ and $\mathbf{L}$ be the binary causal mask. Unmasked third-order tensor attention uses the matrix $\mathbf{A}\mathbf{A}^\top \mathbf{A}$. Its (t, j)-entry is

$$[(\mathbf{A}\mathbf{A}^\top \mathbf{A})]_{t,j} = \sum_{u \leq n} \left(\sum_{i \leq n} (\mathbf{q}_t^\top \mathbf{k}_i)(\mathbf{q}_u^\top \mathbf{k}_i) \right) (\mathbf{q}_u^\top \mathbf{k}_j) = \mathbf{q}_t^\top (\mathbf{K}^\top \mathbf{K}) \left(\sum_u \mathbf{q}_u \mathbf{q}_u^\top \right) \mathbf{k}_j,$$

which immediately yields a streaming factorization through prefix moments.

Unmasked factorization. Define prefix summaries $\mathbf{S}_t^K = \sum_{i \leq t} \mathbf{k}_i \mathbf{k}_i^\top \in \mathbb{R}^{d \times d}$, $\mathbf{S}_t^Q = \sum_{i \leq t} \mathbf{q}_i \mathbf{q}_i^\top \in \mathbb{R}^{d \times d}$, $\mathbf{P}_t^{KV} = \sum_{i \leq t} \mathbf{k}_i \mathbf{v}_i^\top \in \mathbb{R}^{d \times d_v}$, $\mathbf{m}_t^K = \sum_{i \leq t} \mathbf{k}_i \in \mathbb{R}^d$. The default (unnormalized) third-order operator is

$$\mathbf{o}_t^{(3)} = \mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{S}_t^Q \mathbf{P}_t^{KV}.$$

An optional normalization divides by $\mathbf{q}_t^\top \mathbf{S}_t^K \mathbf{S}_t^Q \mathbf{m}_t^K + \varepsilon$ if desired.

Masked streaming summaries. To impose strict causality, we introduce cross-summaries:

$$\mathbf{G}_t^{(1)} := \sum_{i \leq t} (\mathbf{k}_i \mathbf{k}_i^\top) \mathbf{S}_{i-1}^Q \mathbf{P}_{i-1}^{KV} \in \mathbb{R}^{d \times d_v}, \qquad \mathbf{h}_t^{(1)} := \sum_{i \leq t} (\mathbf{k}_i \mathbf{k}_i^\top) \mathbf{S}_{i-1}^Q \mathbf{m}_{i-1}^K \in \mathbb{R}^d,$$

$$\mathbf{G}_t^{(2)} := \sum_{i \leq t} \mathbf{S}_{i-1}^K (\mathbf{q}_i \mathbf{q}_i^\top) \mathbf{P}_{i-1}^{KV} \in \mathbb{R}^{d \times d_v}, \qquad \mathbf{h}_t^{(2)} := \sum_{i \leq t} \mathbf{S}_{i-1}^K (\mathbf{q}_i \mathbf{q}_i^\top) \mathbf{m}_{i-1}^K \in \mathbb{R}^d,$$

$$\mathbf{G}_t^{(3)} := \sum_{i \leq t} \mathbf{S}_{i-1}^K \mathbf{S}_{i-1}^Q (\mathbf{k}_i \mathbf{v}_i^\top) \in \mathbb{R}^{d \times d_v}, \qquad \mathbf{h}_t^{(3)} := \sum_{i \leq t} \mathbf{S}_{i-1}^K \mathbf{S}_{i-1}^Q \mathbf{k}_i \in \mathbb{R}^d.$$

Then the masked, unnormalized quantities are defined as follows:

$$\text{num}_t^{(3)\text{mask}} = \mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{S}_t^Q \mathbf{P}_t^{KV} - \mathbf{G}_t^{(1)} - \mathbf{G}_t^{(2)} - \mathbf{G}_t^{(3)} \right),$$

$$\text{den}_t^{(3)\text{mask}} = \mathbf{q}_t^\top \left(\mathbf{S}_t^K \mathbf{S}_t^Q \mathbf{m}_t^K - \mathbf{h}_t^{(1)} - \mathbf{h}_t^{(2)} - \mathbf{h}_t^{(3)} \right).$$

The following theorem shows that the (normalized) output of third-order HLA can be computed based on $\text{num}_t^{(3)\text{mask}}$ and $\text{den}_t^{(3)\text{mask}}$.

Theorem 7.1 (Masked streaming identity for third order). For each t, the strictly causal third-order output in the default (unnormalized) form is

$$\mathbf{o}_t^{(3)} = \text{num}_t^{(3)\text{mask}}.$$

An optional normalized variant divides by the masked denominator,

$$\mathbf{o}_t^{(3)} = \frac{\text{num}_t^{(3)\text{mask}}}{\text{den}_t^{(3)\text{mask}} + \varepsilon}.$$

and the online updates are

$$\mathbf{S}_t^K = \mathbf{S}_{t-1}^K + \mathbf{k}_t \mathbf{k}_t^\top, \qquad\qquad \mathbf{S}_t^Q = \mathbf{S}_{t-1}^Q + \mathbf{q}_t \mathbf{q}_t^\top,$$
$$\mathbf{P}_t^{KV} = \mathbf{P}_{t-1}^{KV} + \mathbf{k}_t \mathbf{v}_t^\top, \qquad\qquad \mathbf{m}_t^K = \mathbf{m}_{t-1}^K + \mathbf{k}_t. \qquad (7.1)$$

$$\mathbf{G}_t^{(1)} = \mathbf{G}_{t-1}^{(1)} + (\mathbf{k}_t \mathbf{k}_t^\top) \mathbf{S}_{t-1}^Q \mathbf{P}_{t-1}^{KV},$$
$$\mathbf{G}_t^{(2)} = \mathbf{G}_{t-1}^{(2)} + \mathbf{S}_{t-1}^K (\mathbf{q}_t \mathbf{q}_t^\top) \mathbf{P}_{t-1}^{KV},$$
$$\mathbf{G}_t^{(3)} = \mathbf{G}_{t-1}^{(3)} + \mathbf{S}_{t-1}^K \mathbf{S}_{t-1}^Q (\mathbf{k}_t \mathbf{v}_t^\top). \qquad (7.2)$$

$$\mathbf{h}_t^{(1)} = \mathbf{h}_{t-1}^{(1)} + (\mathbf{k}_t \mathbf{k}_t^\top) \mathbf{S}_{t-1}^Q \mathbf{m}_{t-1}^K,$$
$$\mathbf{h}_t^{(2)} = \mathbf{h}_{t-1}^{(2)} + \mathbf{S}_{t-1}^K (\mathbf{q}_t \mathbf{q}_t^\top) \mathbf{m}_{t-1}^K,$$
$$\mathbf{h}_t^{(3)} = \mathbf{h}_{t-1}^{(3)} + \mathbf{S}_{t-1}^K \mathbf{S}_{t-1}^Q \mathbf{k}_t. \qquad (7.3)$$

Proof. Let $\mathbf{W} = \mathbf{L} \odot (\mathbf{Q}\mathbf{K}^\top)$ and consider $(\mathbf{W}\mathbf{W}^\top\mathbf{W})\mathbf{V}$. The t-th row applied to $\mathbf{V}$ is

$$\sum_{j \leq t} \left(\sum_{u \leq t} (\mathbf{W}\mathbf{W}^\top)_{t,u} \, \mathbf{W}_{u,j} \right) \mathbf{v}_j^\top \;=\; \mathbf{q}_t^\top \sum_{j \leq t} \left(\sum_{u \leq t} \mathbf{S}_u^K \, \mathbf{q}_u \mathbf{q}_u^\top \right) \mathbf{k}_j \mathbf{v}_j^\top.$$

Using $\sum_{u \leq t} \mathbf{S}_u^K = \mathbf{S}_t^K + \sum_{u \leq t-1} (\mathbf{S}_u^K)$ and repeatedly applying $\sum_{i \leq u} = \sum_{i \leq t} - \sum_{u < i \leq t}$ to peel off the dependence on future indices relative to each summation boundary yields

$$\sum_{j \leq t} \mathbf{S}_t^K \mathbf{S}_t^Q \, \mathbf{k}_j \mathbf{v}_j^\top - \sum_{i \leq t} (\mathbf{k}_i \mathbf{k}_i^\top) \mathbf{S}_{i-1}^Q \mathbf{P}_{i-1}^{KV} - \sum_{i \leq t} \mathbf{S}_{i-1}^K (\mathbf{q}_i \mathbf{q}_i^\top) \mathbf{P}_{i-1}^{KV} - \sum_{i \leq t} \mathbf{S}_{i-1}^K \mathbf{S}_{i-1}^Q (\mathbf{k}_i \mathbf{v}_i^\top),$$

which is precisely $\mathbf{S}_t^K \mathbf{S}_t^Q \mathbf{P}_t^{KV} - \mathbf{G}_t^{(1)} - \mathbf{G}_t^{(2)} - \mathbf{G}_t^{(3)}$. Left-multiplication by $\mathbf{q}_t^\top$ gives the masked numerator, and replacing $\mathbf{v}_j$ by 1 yields the denominator. Online updates follow by isolating the $i = t$ contributions and using $(\mathbf{k}\mathbf{k}^\top)X = \mathbf{k}(\mathbf{k}^\top X)$. $\qquad\square$

7.2 Pseudocode

We present explicit pseudocode for masked third-order HLA in two parts: (i) a strictly causal streaming kernel for inference, and (ii) the associative scan operator used for chunk-parallel training. All operations are per head; shapes follow Section 7.1.

Algorithm 3 Masked (Third Order) HLA Streaming Kernel

Require: Sequences $\{\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t\}_{t=1}^n$, decay $\gamma \in (0,1]$, stability $\varepsilon > 0$, flag `normalize`
1: **Init:** $\mathbf{S}^K = \mathbf{0}_{d \times d}$, $\mathbf{S}^Q = \mathbf{0}_{d \times d}$, $\mathbf{P}^{KV} = \mathbf{0}_{d \times d_v}$, $\mathbf{m}^K = \mathbf{0}_d$
2: $\qquad \mathbf{G}^{(1)} = \mathbf{0}_{d \times d_v}$, $\mathbf{G}^{(2)} = \mathbf{0}_{d \times d_v}$, $\mathbf{G}^{(3)} = \mathbf{0}_{d \times d_v}$, $\mathbf{h}^{(1)} = \mathbf{0}_d$, $\mathbf{h}^{(2)} = \mathbf{0}_d$, $\mathbf{h}^{(3)} = \mathbf{0}_d$
3: **for** $t = 1$ to n **do**
4: $\qquad \mathbf{S}_{\text{prev}}^K \leftarrow \mathbf{S}^K$; $\;\mathbf{S}_{\text{prev}}^Q \leftarrow \mathbf{S}^Q$; $\;\mathbf{P}_{\text{prev}} \leftarrow \mathbf{P}^{KV}$; $\;\mathbf{m}_{\text{prev}} \leftarrow \mathbf{m}^K$
5: $\qquad$ **Inclusive first-order updates (with decay):**
6: $\qquad \mathbf{S}^K \leftarrow \gamma \mathbf{S}_{\text{prev}}^K + \mathbf{k}_t \mathbf{k}_t^\top$; $\quad \mathbf{S}^Q \leftarrow \gamma \mathbf{S}_{\text{prev}}^Q + \mathbf{q}_t \mathbf{q}_t^\top$
7: $\qquad \mathbf{P}^{KV} \leftarrow \gamma \mathbf{P}_{\text{prev}} + \mathbf{k}_t \mathbf{v}_t^\top$; $\quad \mathbf{m}^K \leftarrow \gamma \mathbf{m}_{\text{prev}} + \mathbf{k}_t$
8: $\qquad$ **Cross-summaries (matvec/outer-product forms):**
9: $\qquad u_1 \leftarrow \mathbf{S}_{\text{prev}}^Q \mathbf{k}_t$; $\quad \mathbf{G}^{(1)} \leftarrow \gamma \mathbf{G}^{(1)} + \mathbf{k}_t(u_1^\top \mathbf{P}_{\text{prev}})$; $\quad \mathbf{h}^{(1)} \leftarrow \gamma \mathbf{h}^{(1)} + \mathbf{k}_t(u_1^\top \mathbf{m}_{\text{prev}})$
10: $\qquad a_2 \leftarrow \mathbf{S}_{\text{prev}}^K \mathbf{q}_t$; $\quad \mathbf{G}^{(2)} \leftarrow \gamma \mathbf{G}^{(2)} + a_2(\mathbf{q}_t^\top \mathbf{P}_{\text{prev}})$; $\quad \mathbf{h}^{(2)} \leftarrow \gamma \mathbf{h}^{(2)} + a_2(\mathbf{q}_t^\top \mathbf{m}_{\text{prev}})$
11: $\qquad u_3 \leftarrow \mathbf{S}_{\text{prev}}^Q \mathbf{k}_t$; $\quad a_3 \leftarrow \mathbf{S}_{\text{prev}}^K u_3$; $\quad \mathbf{G}^{(3)} \leftarrow \gamma \mathbf{G}^{(3)} + a_3 \mathbf{v}_t^\top$; $\quad \mathbf{h}^{(3)} \leftarrow \gamma \mathbf{h}^{(3)} + a_3$
12: $\qquad$ **Output (unnormalized by default):**
13: $\qquad y \leftarrow \mathbf{S}^K \mathbf{q}_t$; $\;z \leftarrow \mathbf{S}^Q y$; $\;\text{termA} \leftarrow z^\top \mathbf{P}^{KV}$; $\;\text{termB} \leftarrow \mathbf{q}_t^\top \mathbf{G}^{(1)}$; $\;\text{termC} \leftarrow \mathbf{q}_t^\top \mathbf{G}^{(2)}$; $\;\text{termD} \leftarrow \mathbf{q}_t^\top \mathbf{G}^{(3)}$
14: $\qquad \mathbf{o}_t \leftarrow \text{termA} - \text{termB} - \text{termC} - \text{termD}$
15: $\qquad$ **if** `normalize` **then**
16: $\qquad\qquad \text{denvec} \leftarrow \mathbf{S}^K (\mathbf{S}^Q \mathbf{m}^K) - \mathbf{h}^{(1)} - \mathbf{h}^{(2)} - \mathbf{h}^{(3)}$
17: $\qquad\qquad \text{den} \leftarrow \mathbf{q}_t^\top \text{denvec} + \varepsilon$; $\quad \mathbf{o}_t \leftarrow \mathbf{o}_t / \text{den}$
18: $\qquad$ **end if**
19: **end for**
20: **return** $\{\mathbf{o}_t\}_{t=1}^n$

Usage. We provide the strictly causal streaming kernel in Algorithm 3. Designing a chunk-parallel scan operator that exactly reproduces serial activations requires additional segment-level summaries beyond $(\mathbf{S}^K, \mathbf{S}^Q, \mathbf{P}^{KV}, \mathbf{m}^K, \mathbf{G}^{(1:3)}, \mathbf{h}^{(1:3)})$; we leave this composition to future work.

13

8 Related Work

The literature on subquadratic sequence modeling spans (i) fast-weight style dynamic-parameter models, (ii) kernel/feature-map linearizations of attention, and (iii) recurrent/state-space approaches. **HLA** belongs to a complementary class: it preserves attention-style, data-dependent mixing but realizes higher interactions through compact prefix moments with exact causal masking and scan-parallel training.

Fast weights and fast weight programmers (FWPs). Fast weights, dating to early connectionist memory models (Hinton and Plaut, 1987), implement short-term, input-dependent synaptic changes. Schmidhuber's fast weight programmers (Schmidhuber, 1992) introduced differentiable controllers that program a separate fast-weight matrix; later, Ba et al. (2016) revived this idea to attend to the recent past. A series of works made the connection to modern attention explicit: Schlag et al. (2021) showed a formal equivalence between linearized self-attention and FWPs, where outer-product updates $\Delta W_t \propto \mathbf{k}_t \mathbf{v}_t^\top$ accumulate an associative memory queried by $\mathbf{q}_t$; Irie et al. (2021) extended FWPs with recurrence in the programmer and the fast net. Yang et al. (2024b) utilizes WY Transformations (Bischof and Van Loan, 1987) to implement chunkwise parallel training of Delta Network (Schlag et al., 2021). Parallel lines explore higher or preconditioned mixing by maintaining or inverting second-moment matrices. In our formulation, $\mathbf{S}_t^K$ plays the role of a learned kernel; working directly with $\mathbf{S}_t^K$ avoids explicit matrix inversion and preserves streaming updates, whereas inverse-based methods typically require heavier linear algebra (Behrouz et al., 2025a,b; von Oswald et al., 2025).

Linear Attention Mechanisms. A common route is to replace the softmax kernel by explicit features ϕ to enable streaming via running sums. Representative examples include Linear Transformers (Katharopoulos et al., 2020), Performer's FAVOR$^+$ random features (Choromanski et al., 2020), and Random Feature Attention (Peng et al., 2021). Earlier work also proposed multiplicative rearrangements that yield linear-complexity-efficient attention (Shen et al., 2021). These methods achieve $O(ndr)$ time with r feature dimension but are typically *first-order* in the sense that they maintain only $\sum \phi(\mathbf{k}) \mathbf{v}^\top$ and (optionally) a scalar denominator. By contrast, second-order **HLA** maintains the full key moment $\mathbf{S}_t^K = \sum_{i \le t} \mathbf{k}_i \mathbf{k}_i^\top$ together with query-value and query mass summaries and their masked cross-summaries, yielding strictly causal higher interactions while remaining streaming. Recent linear attention variants include Sun et al. (2023); Qin et al. (2023); Yang et al. (2023); Qin et al. (2024); Yang et al. (2024b); von Oswald et al. (2025).

State Space Models. SSMs (e.g., S4) (Gu et al., 2021) and selective SSMs (e.g., Mamba) (Gu and Dao, 2023; Dao and Gu, 2024) realize $O(1)$ per-token state updates via linear recurrences and convolutions. These architectures excel at long-range dependencies but express data-dependent mixing differently from attention. **HLA** sits in between: it is attention-like (data-dependent queries/keys) yet streams via compact prefix statistics like recurrent models.

Modern RNNs. Recent modern RNN designs emphasize gating, decays, and associative scan–friendly recurrences that enable parallel training while preserving strictly constant per-token state at inference. Examples include gated linear mixers and decay-aware updates (Yang et al., 2023, 2024a), efficient gradient routing and training strategies for long sequences (Qin et al., 2024; Peng et al., 2024; Sun et al., 2024), and RWKV-style architectures that replace attention with learned decays and elementwise gating (Peng et al., 2025). These methods typically maintain first-order sufficient statistics and rely on fixed linear dynamics. In contrast, **HLA** retains attention-style, data-dependent metrics via $\mathbf{S}_t^K$ and higher cross-summaries while keeping the same $O(1)$ per-token state update

paradigm, offering a complementary inductive bias to RNNs and SSMs.

Test Time Training and Memory Networks. Test-time adaptation and explicit long-term memory are emerging tools for extending context without quadratic cost. Test-time training variants adapt parameters from recent tokens to improve local coherence (Sun et al., 2024), whereas memory networks maintain external stores addressable by content keys (Behrouz et al., 2024, 2025a,b). The **HLA** view is orthogonal: it encodes higher interactions in compact *prefix moments* that are sufficient for exact masked streaming and scan-parallel training.

Associative memory and Hopfield views. Modern Hopfield networks show that transformer attention is a single-step retrieval in an energy-based associative memory (Ramsauer et al., 2020; Zhong et al., 2025). While this perspective clarifies why attention uses content-addressable memory, standard Hopfield-style layers remain first-order in their sufficient statistics. **HLA** complements this view by providing explicit higher sufficient statistics with strict causality.

9   Conclusion

We introduced Higher-order Linear Attention (**HLA**), a causal higher attention mechanism with exact streaming updates, a strictly causal masked formulation via extended summaries, and associative scans for parallel training that provably match serial recurrences. At second order, HLA maintains $O(d^2)$ state per head and computes each token in $O(d^2)$ time, with optional normalization and decay that preserve associativity. We further developed an asymmetric variant (AHLA) and a complete third-order masked algebra with streaming formulas and scan operators.

References

Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

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

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Atlas: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025a.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. *arXiv preprint arXiv:2504.13173*, 2025b.

Christian Bischof and Charles Van Loan. The wy representation for products of householder matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(1):s2–s13, 1987.

Guy E Blelloch. Prefix sums and their applications. 1990.

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.

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

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

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society*, pages 177–186, 1987.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. *Advances in neural information processing systems*, 34:7703–7717, 2021.

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.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, et al. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*, 2024.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, et al. Rwkv-7" goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*, 2025.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

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.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Lightning attention-2: A free lunch for handling unlimited sequence lengths in large language models. *arXiv preprint arXiv:2401.04658*, 2024.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

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

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

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. *arXiv preprint arXiv:2307.08621*, 2023.

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

Johannes von Oswald, Nino Scherrer, Seijin Kobayashi, Luca Versari, Songlin Yang, Maximilian Schlegel, Kaitlin Maile, Yanick Schimpf, Oliver Sieberling, Alexander Meulemans, et al. Mesanet: Sequence modeling by locally optimal test-time training. *arXiv preprint arXiv:2506.05233*, 2025.

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

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

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024a.

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*, 2024b.

Shu Zhong, Mingyu Xu, Tenglong Ao, and Guang Shi. Understanding transformer from the perspective of associative memory. *arXiv preprint arXiv:2505.19488*, 2025.