GATED DELTA NETWORKS: IMPROVING MAMBA2 WITH DELTA RULE

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear Transformers have gained attention as efficient alternatives to standard Transformers, but their performance in retrieval and long-context tasks has been limited. To address these limitations, recent work has explored two distinct mechanisms: gating for adaptive memory control and the delta update rule for precise memory modifications. We observe that these mechanisms are complementary—gating enables rapid memory erasure while the delta rule facilitates targeted updates. Building on this insight, we introduce the gated delta rule and develop a parallel training algorithm optimized for modern hardware. Our proposed architecture, Gated DeltaNet, consistently surpasses existing models like Mamba2 and DeltaNet across multiple benchmarks, including language modeling, common-sense reasoning, in-context retrieval, length extrapolation, and long-context understanding. We further enhance performance by developing hybrid architectures that combine Gated DeltaNet layers with sliding window attention or Mamba2 layers, achieving both improved training efficiency and superior task performance.

1 INTRODUCTION

The Transformer architecture has significantly advanced the capabilities of Large Language Models (LLMs), showcasing exceptional performance across a wide range of tasks due to its effective attention mechanism. This mechanism excels in precise sequence modeling and leverages the parallel processing capabilities of modern GPUs during training. However, the self-attention component scales quadratically with sequence length, leading to substantial computational demands that pose challenges for both training and inference.

To mitigate these issues, researchers have explored alternatives such as linear Transformers (Katharopoulos et al., 2020a), which replace traditional softmax-based attention with kernelized dot-product-based linear attention, substantially reducing memory requirements during inference by reframing as a linear RNN with matrix-valued states. While early versions of linear Transformers underperformed in language modeling tasks compared to standard Transformers, recent enhancements—such as incorporating data-dependent gating mechanisms akin to those in LSTMs, exemplified by models like GLA (Yang et al., 2024a) and Mamba2 (Dao & Gu, 2024a)—have shown promising improvements. However, challenges persist in managing information over long sequences, particularly for in-context retrieval tasks where traditional Transformers maintain their advantage (Arora et al., 2023a; 2024a; Jelassi et al., 2024; Wen et al., 2024; Akyürek et al., 2024).

This phenomenon is not surprising: linear Transformers can be interpreted as implementing an outer-product-based key-value association memory, reminiscent of tensor product representation (Smolensky, 1990). However, the number of orthogonal key-value pairs they can store is *bounded* by the model's dimensionality. When the sequence length exceeds this dimension, "memory collisions" become inevitable, hindering exact retrieval (Schlag et al., 2021a).

Mamba2 addresses this limitation by introducing a simple gated update rule, $\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\intercal$, which uniformly decays all key-value associations at each time step by a dynamic ratio, α_t. However, this approach does not account for the varying importance of different key-value associations, potentially leading to inefficient memory utilization. If the model needs to forget a specific key-value association, all key-value associations are equally forgotten, making the process less targeted and efficient.

In contrast, the linear Transformer with the delta rule (Widrow et al., 1960), known as DeltaNet (Schlag et al., 2021a; Yang et al., 2024b), selectively updates memory by (softly) replacing an old key-value pair with the incoming one in a sequential manner. This method has demonstrated impressive performance in synthetic benchmarks for in-context retrieval. However, since this process only modifies a single key-value pair at a time, the model lacks the ability to rapidly clear outdated or irrelevant information, especially during context switches where previous data needs to be erased. Consequently, DeltaNet has been found to perform moderately on real-world tasks (Yang et al., 2024b), likely due to the absence of a robust memory-clearing mechanism.

Recognizing the complementary advantages of the gated update rule and the delta rule in memory management, we propose the *gated delta rule*, a simple and intuitive mechanism that combines both approaches. This unified rule enables flexible memory control: it can promptly clear memory by setting $\alpha_t \to 0$, while selectively updating specific content without affecting other information by setting $\alpha_t \to 1$ (effectively switching to the pure delta rule).

The remaining challenge lies in implementing the gated delta rule in a hardware-efficient manner. Building upon Yang et al. (2024b)'s efficient algorithm that parallelizes the delta rule computation using the WY representation (Bischof & Loan, 1985), we carefully extend their approach to incorporate the gating terms. Our extension preserves the benefits of chunkwise parallelism (Hua et al., 2022; Sun et al., 2023a; Yang et al., 2024a), enabling hardware-efficient training.

Our resulting architecture, Gated DeltaNet, consistently outperforms both Mamba2 and DeltaNet across a comprehensive suite of benchmarks, including language modeling, commonsense reasoning, in-context retrieval, length extrapolation, and long-context understanding. Building on these results, we also develop hybrid architectures that strategically combine Gated DeltaNet layers with sliding window attention or Mamba2 layers, further enhancing both training efficiency and model performance.

2 Preliminary

2.1 Linear Attention with Chunkwise Parallel Form

It is known that the linear transformer (Katharopoulos et al., 2020b) can be formulated as the following linear recurrence when excluding normalization and query/key activations:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\intercal \in \mathbb{R}^{d_v \times d_k}, \qquad \boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t \in \mathbb{R}^{d_v}$$

where d_k and d_v represent the (head) dimensions for query/key and value, respectively. By expanding the recurrence, we can express it in both vector form (left) and matrix form (right) as follows:

$$\boldsymbol{o}_t = \sum_{i=1}^{t} (\boldsymbol{v}_i \boldsymbol{k}_i^\intercal) \boldsymbol{q}_t = \sum_{i=1}^{t} \boldsymbol{v}_i (\boldsymbol{k}_i^\intercal \boldsymbol{q}_t) \in \mathbb{R}^{d_v}, \qquad \mathbf{O} = (\mathbf{Q}\mathbf{K}^\intercal \odot \mathbf{M})\mathbf{V} \in \mathbb{R}^{L \times d_v}$$

where L is the sequence length, and $\mathbf{M} \in \mathbb{R}^{L \times L}$ is the causal mask defined by $\mathbf{M}_{ij} = 0$ when $i < j$, and 1 otherwise.

This formulation makes it clear that linear attention eliminates the softmax operation used in traditional attention mechanisms and instead leverages the linearity and associativity of matrix multiplications, leading to linear complexity. However, both the recurrent and parallel forms are not ideal for efficient training (Yang et al., 2024a), which motivates the use of the chunkwise parallel form (Hua et al., 2022; Sun et al., 2023a; Yang et al., 2024a) for hardware-efficient, linear-time training, as introduced below.

Chunkwise parallel form. To summarize, the chunkwise parallel form splits inputs and outputs into several chunks of size C, and computes outputs for each chunk based on the final state of the previous chunk and the query/key/value blocks of the current chunk. Following the notation of Sun et al. (2023b); Yang et al. (2024a;b), let's take the query block, $\boldsymbol{q}$, as an example. We denote $\mathbf{Q}_{[t]} := \boldsymbol{q}_{tC+1:(t+1)C+1}$ as the query block for chunk t, and $\boldsymbol{q}_{[t]}^r := \boldsymbol{q}_{tC+r}$ as the r-th query within chunk t. The initial state of chunk t is defined as $\mathbf{S}_{[t]} := \mathbf{S}_{[t]}^0 = \mathbf{S}_{[t-1]}^C$. By partially expanding the

recurrence, we have

$$\mathbf{S}_{[t]}^r = \mathbf{S}_{[t]} + \sum_{i=1}^{r} \boldsymbol{v}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal} \in \mathbb{R}^{d_v \times d_k}, \qquad \boldsymbol{o}_{[t]}^r = \mathbf{S}_{[t]}^r \boldsymbol{q}_{[t]}^r = \mathbf{S}_{[t]} \boldsymbol{q}_{[t]}^r + \sum_{i=1}^{r} \boldsymbol{v}_{[t]}^i \left(\boldsymbol{k}_{[t]}^{i\intercal} \boldsymbol{q}_{[t]}^r \right) \in \mathbb{R}^{d_v}$$

Equivalently, in matrix form:

$$\mathbf{S}_{[t+1]} = \mathbf{S}_{[t]} + \mathbf{V}_{[t]} \mathbf{K}_{[t]}^\intercal \in \mathbb{R}^{d_v \times d_k}, \qquad \mathbf{O}_{[t]} = \mathbf{Q}_{[t]} \mathbf{S}_{[t]}^\intercal + \left(\mathbf{Q}_{[t]} \mathbf{K}_{[t]}^\intercal \odot \mathbf{M} \right) \mathbf{V}_{[t]} \in \mathbb{R}^{C \times d_v}$$

where $\mathbf{M} \in \mathbb{R}^{C \times C}$ is the causal mask. The above equations are rich in matrix multiplications (matmuls), and by setting C to a multiple of 16, one can take advantage of tensor cores—specialized GPU units for efficient half-precision matmul operations—for hardware-efficient training. Typically, C is set to a small constant (e.g., 64 as implemented in FLA (Yang & Zhang, 2024)), ensuring that the overall computational complexity remains linear with respect to sequence length, enabling efficient modeling of extremely long sequences.

2.2 Mamba2: Linear attention with scalar-valued data-dependent decay

Mamba2 (Dao & Gu, 2024a) can be represented by the following linear recurrence (up to specific parameterization):

$$\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\intercal, \qquad \boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$$

where $\alpha_t \in (0, 1)$ is a **data-dependent** scalar-valued decay term. In the following, we will highlight the decay terms in blue to facilitate a clearer comparison with vanilla linear attention. Define the cumulative decay product $\gamma_j = \prod_{i=1}^{j} \alpha_i$, and by expanding the recurrence, we can express the result in both a vector form (left) and a matrix parallel form (right):

$$\boldsymbol{o}_t = \sum_{i=1}^{t} \left(\frac{\gamma_t}{\gamma_i} \boldsymbol{v}_i \boldsymbol{k}_i^\intercal \right) \boldsymbol{q}_t = \sum_{i=1}^{t} \boldsymbol{v}_i \left(\frac{\gamma_t}{\gamma_i} \boldsymbol{k}_i^\intercal \boldsymbol{q}_t \right), \qquad \mathbf{O} = \left(\left(\mathbf{Q}\mathbf{K}^\intercal \right) \odot \Gamma \right) \mathbf{V}$$

Here, $\Gamma \in \mathbb{R}^{L \times L}$ is a decay-aware causal mask where $\Gamma_{ij} = \frac{\gamma_i}{\gamma_j}$ if $i \geq j$ and $\Gamma_{ij} = 0$ otherwise.

This parallel and recurrent formulation is referred to as state space duality (SSD) in Dao & Gu (2024a). Notably, this recurrence structure has also been employed in Gated RFA (Peng et al., 2021), xLSTM (Beck et al., 2024), and Gated RetNet (Sun et al., 2024b).

Chunkwise parallel form. Slightly abusing the notation, we define the local cumulative product of decays within the chunk as $\gamma_{[t]}^j = \prod_{i=tC+1}^{tC+j} \alpha_i$. Additionally, we define $(\Gamma_{[t]})_{ij} = \frac{\gamma_{[t]}^j}{\gamma_{[t]}^i}$ for $i \geq j$ and 0 otherwise. By partially expanding the recurrence, we obtain the following equations:

$$\mathbf{S}_{[t]}^r = \gamma_{[t]}^r \mathbf{S}_{[t]} + \sum_{i=1}^{r} \frac{\gamma_{[t]}^r}{\gamma_{[t]}^i} \boldsymbol{v}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal}, \qquad \boldsymbol{o}_{[t]}^r = \gamma_{[t]}^r \mathbf{S}_{[t]}^r \boldsymbol{q}_{[t]}^r = \mathbf{S}_{[t]} \boldsymbol{q}_{[t]}^r + \sum_{i=1}^{r} \boldsymbol{v}_{[t]}^i \left(\frac{\gamma_{[t]}^r}{\gamma_{[t]}^i} \boldsymbol{k}_{[t]}^{i\intercal} \boldsymbol{q}_{[t]}^r \right)$$

This can be equivalently expressed in matrix form as:

$$\mathbf{S}_{[t+1]} = \gamma_{[t]}^C \mathbf{S}_{[t]} + \mathbf{V}_{[t]}^\intercal \mathrm{Diag}\left(\frac{\gamma_{[t]}^C}{\gamma_{[t]}} \right) \mathbf{K}_{[t]}$$

$$\mathbf{O}_{[t]} = \mathrm{Diag}\left(\gamma_{[t]} \right) \mathbf{Q}_{[t]} \mathbf{S}_{[t]}^\intercal + \left(\mathbf{Q}_{[t]} \mathbf{K}_{[t]}^\intercal \odot \Gamma_{[t]} \right) \mathbf{V}_{[t]}$$

We observe that the (cumulative) decay term can be seamlessly integrated into the matmuls with minimal computational overhead. This ensures that the chunkwise parallel form remains efficient and compatible with high-performance tensor core-based acceleration.

2.3 Delta Networks: Linear Attention with Delta Rule

The delta update rule (Widrow et al., 1960; Schlag et al., 2021b) *dynamically* erases the value ($\boldsymbol{v}_t^{\text{old}}$) associated with the current input key ($\boldsymbol{k}_t$) and writes a new value ($\boldsymbol{v}_t^{\text{new}}$), which is a linear combination

of the current input value and the old value. This process updates a key-value association pair at each time step, where the scalar $\beta_t \in (0,1)$ determines the extent to which the old association is replaced by the new one, as shown below.

$$\mathbf{S}_t = \mathbf{S}_{t-1} - \underbrace{(\mathbf{S}_{t-1}\boldsymbol{k}_t)}_{\boldsymbol{v}_t^{\text{old}}}\boldsymbol{k}_t^\intercal + \underbrace{(\beta_t\boldsymbol{v}_t + (1-\beta_t)\mathbf{S}_{t-1}\boldsymbol{k}_t))}_{\boldsymbol{v}_t^{\text{new}}}\boldsymbol{k}_t^\intercal = \mathbf{S}_{t-1}\left(\mathbf{I} - \beta_t\boldsymbol{k}_t\boldsymbol{k}_t^\intercal\right) + \beta_t\boldsymbol{v}_t\boldsymbol{k}_t^\intercal$$

Chunkwise parallel form. By partially expanding the recurrence, we have

$$\mathbf{S}_{[t]}^r = \mathbf{S}_{[t]}\underbrace{\left(\prod_{i=1}^{r}\mathbf{I} - \beta_{[t]}^i\boldsymbol{k}_{[t]}^i\boldsymbol{k}_{[t]}^{i\intercal}\right)}_{:=\mathbf{P}_{[t]}^r} + \underbrace{\sum_{i=1}^{r}\left(\beta_{[t]}^i\boldsymbol{v}_{[t]}^i\boldsymbol{k}_{[t]}^{i\intercal}\prod_{j=i+1}^{r}\left(\mathbf{I} - \beta_{[t]}^j\boldsymbol{k}_{[t]}^j\boldsymbol{k}_{[t]}^{j\intercal}\right)\right)}_{:=\mathbf{H}_t^r} \tag{1}$$

where $\mathbf{P}_{[t]}^j$ involves cumulative products of transition matrices. Yang et al. (2024b) show these take the form of (generalized) Householder matrices, allowing memory-efficient computation through classical WY representation (Bischof & Loan, 1985). Based on this, they introduce two compact representations to optimize the process:

$$\mathbf{P}_{[t]}^r = \mathbf{I} - \sum_{i=1}^{r}\mathbf{w}_{[t]}^i\boldsymbol{k}_{[t]}^{i\intercal} \in \mathbb{R}^{d_k\times d_k} \qquad \mathbf{H}_{[t]}^r = \sum_{i=1}^{r}\mathbf{u}_{[t]}^i\boldsymbol{k}_{[t]}^{i\intercal} \in \mathbb{R}^{d_v\times d_k} \tag{2}$$

$$\mathbf{w}_{[t]}^r = \beta_{[t]}^r\left(\boldsymbol{k}_{[t]}^r - \sum_{i=1}^{r-1}\left(\mathbf{w}_{[t]}^i(\boldsymbol{k}_{[t]}^{i\intercal}\boldsymbol{k}_{[t]}^r)\right)\right) \qquad \mathbf{u}_{[t]}^r = \beta_{[t]}^r\left(\boldsymbol{v}_{[t]}^r - \sum_{i=1}^{r-1}\left(\mathbf{u}_{[t]}^i(\boldsymbol{k}_{[t]}^{i\intercal}\boldsymbol{k}_{[t]}^r)\right)\right) \tag{3}$$

where $\mathbf{w}_{[t]}^r \in \mathbb{R}^{d_k}$ and $\mathbf{u}_{[t]}^r \in \mathbb{R}^{d_v}$. Substituting these back into Eq. 1 and in matrix form, we have:

$$\mathbf{S}_{[t+1]} = \mathbf{S}_{[t]} + \left(\mathbf{U}_{[t]} - \mathbf{W}_{[t]}\mathbf{S}_{[t]}^\intercal\right)^\intercal\mathbf{K}_{[t]} \tag{4}$$

$$\mathbf{O}_{[t]} = \mathbf{Q}_{[t]}\mathbf{S}_{[t]}^\intercal + (\mathbf{Q}_{[t]}\mathbf{K}_{[t]}^\intercal \odot \mathbf{M})\left(\mathbf{U}_{[t]} - \mathbf{W}_{[t]}\mathbf{S}_{[t]}^\intercal\right) \tag{5}$$

3 Gated Delta Networks

3.1 Formulation: Gated Delta Rule

The proposed gated delta rule is simple yet effective:

$$\mathbf{S}_t = \mathbf{S}_{t-1}\left(\alpha_t(\mathbf{I} - \beta_t\boldsymbol{k}_t\boldsymbol{k}_t^\intercal)\right) + \beta_t\boldsymbol{v}_t\boldsymbol{k}_t^\intercal \tag{6}$$

where the data-dependent gating term $\alpha_t \in (0,1)$ controls state decay. This formulation unifies the advantages of both gating mechanisms and the delta rule: the gating term enables adaptive memory management, while the delta update structure facilitates effective key-value association learning.

We present a formal analysis of the gated delta rule through the lens of the online learning framework introduced by Liu et al. (2024). In this framework, recurrent state updates emerge as solutions to an online learning problem with objective function $\boldsymbol{L_t}(S)$. As shown in Table 1, recent linear RNN architectures typically incorporate a regularization term in their online learning objective to prevent state divergence from previous values, thereby enabling memory retention. However, this retention mechanism becomes problematic when the state becomes saturated with information. In such cases, each state must encode a superposition of multiple information pieces, making precise retrieval challenging. To address this limitation, Mamba2 and Gated DeltaNet introduce an adaptive scaling factor α_t that relaxes the regularization term, allowing controlled deviations between S_t and S_{t-1}. This modification enables dynamic memory management through selective forgetting.

On the other hand, Linear Attention (LA) and Mamba2 use a simple linear prediction loss $\langle\mathbf{S}_t\boldsymbol{k}_t, \boldsymbol{v}_t\rangle$, while Longhorn (Liu et al., 2024) uses a more expressive online regression objective $\|\mathbf{S}_t\boldsymbol{k}_t - \boldsymbol{v}_t\|^2$ for better modeling of key-value associations. The resulting Longhorn's update rule closely resembles

Table 1: Comparison of different linear RNN models and their corresponding online learning objectives using the framework from Liu et al. (2024). For convenience, we simplify Longhorn's vector-valued $\boldsymbol{\beta}$ to scalar β.

Method	Online Learning Objective $L_t(S)$	Online Update
LA	$\|\mathbf{S}_t - \mathbf{S}_{t-1}\|_F^2 - 2\langle \mathbf{S}_t \boldsymbol{k}_t, \boldsymbol{v}_t \rangle$	$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^T$
Mamba2	$\|\mathbf{S}_t - \alpha_t \mathbf{S}_{t-1}\|_F^2 - 2\langle \mathbf{S}_t \boldsymbol{k}_t, \boldsymbol{v}_t \rangle$	$\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^T$
Longhorn	$\|\mathbf{S}_t - \mathbf{S}_{t-1}\|_F^2 - \beta_t \|\mathbf{S}_t \boldsymbol{k}_t - \boldsymbol{v}_t\|^2$	$\mathbf{S}_t = \mathbf{S}_{t-1}(\mathbf{I} - \epsilon \boldsymbol{k}_t \boldsymbol{k}_t^T) + \epsilon_t \boldsymbol{v}_t \boldsymbol{k}_t^T, \epsilon_t = \dfrac{\beta_t}{1 + \beta_t \boldsymbol{k}_t^\top \boldsymbol{k}_t}$
DeltaNet	$\|\mathbf{S}_t - \mathbf{S}_{t-1}\|_F^2 - 2\langle \mathbf{S}_t \boldsymbol{k}_t, \beta_t (\boldsymbol{v}_t - \mathbf{S}_{t-1} \boldsymbol{k}_t) \rangle$	$\mathbf{S}_t = \mathbf{S}_{t-1}(\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^T) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^T$
Gated DeltaNet	$\|\mathbf{S}_t - \alpha_t \mathbf{S}_{t-1}\|_F^2 - 2\langle \mathbf{S}_t \boldsymbol{k}_t, \beta_t (\boldsymbol{v}_t - \alpha_t \mathbf{S}_{t-1} \boldsymbol{k}_t) \rangle$	$\mathbf{S}_t = \mathbf{S}_{t-1} \left(\alpha_t (\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^T) \right) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^T$

the delta update rule [1], suggesting the superiority of the (gated) delta rule over Mamba2 in in-context associative recall.

From the perspective of fast weight programming (Irie et al., 2022a) and test-time training (Sun et al., 2024a), the hidden state $\mathbf{S}$ can be interpreted as a weight matrix, with the delta rule optimizing the objective $L(\mathbf{S}_t) = \frac{1}{2}\|\mathbf{S}_t \boldsymbol{k}_t - \boldsymbol{v}_t\|^2$ via *online* stochastic gradient descent (SGD):

$$\mathbf{S}_{t+1} = \mathbf{S}_t - \beta_t \nabla_S L(\mathbf{S}_t) = \mathbf{S}_t - \beta_t (\mathbf{S}_t \boldsymbol{k}_t - \boldsymbol{v}_t) \boldsymbol{k}_t^\intercal = \mathbf{S}_t \left(\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\intercal\right) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^\intercal$$

where β_t represents the (adaptive) learning rate. From this perspective, the gated delta rule can be viewed as incorporating an adaptive weight decay term α_t into the SGD update, a technique widely used in deep learning (Krogh & Hertz, 1991; Andriushchenko et al., 2023).

3.2 Algorithm: Hardware-efficient Chunkwise Training

In this subsection, we describe an efficient chunkwise algorithm for gated delta rule.

Chunkwise parallel form. By partially expanding the recurrence, we have

$$\mathbf{S}_{[t]}^r = \mathbf{S}_{[t]} \underbrace{\left(\prod_{i=1}^{r} \alpha_{[t]}^i \left(\mathbf{I} - \beta_{[t]}^i \boldsymbol{k}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal} \right) \right)}_{:=\mathbf{P}_{[t]}^r} + \underbrace{\sum_{i=1}^{r} \left(\beta_{[t]}^i \boldsymbol{v}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal} \prod_{j=i+1}^{r} \alpha_{[t]}^j \left(\mathbf{I} - \beta_{[t]}^j \boldsymbol{k}_{[t]}^j \boldsymbol{k}_{[t]}^{j\intercal} \right) \right)}_{:=\mathbf{H}_{[t]}^r}$$

We adapt the WY representation in Eq. 2-3 to incorporate the decay term as below,

$$\mathbf{P}_{[t]}^r = \gamma_{[t]}^r \left(\mathbf{I} - \sum_{i=1}^{r} \mathbf{w}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal} \right) \qquad \mathbf{H}_{[t]}^r = \sum_{i=1}^{r} \frac{\gamma_t^r}{\gamma_t^i} \mathbf{u}_{[t]}^i \boldsymbol{k}_{[t]}^{i\intercal} \tag{7}$$

$$\mathbf{w}_{[t]}^r = \beta_{[t]}^r \left(\boldsymbol{k}_{[t]}^r - \sum_{i=1}^{r-1} \left(\mathbf{w}_{[t]}^i (\boldsymbol{k}_{[t]}^{i\intercal} \boldsymbol{k}_{[t]}^r) \right) \right) \qquad \mathbf{u}_{[t]}^r = \beta_{[t]}^r \left(\boldsymbol{v}_{[t]}^r - \sum_{i=1}^{r-1} \left(\mathbf{u}_{[t]}^i (\frac{\gamma_{[t]}^r}{\gamma_{[t]}^i} \boldsymbol{k}_{[t]}^{i\intercal} \boldsymbol{k}_{[t]}^r) \right) \right) \tag{8}$$

and the proof of correctness can be found at Appendix. Equivalently, in matrix form:

$$\mathbf{S}_{[t+1]} = \gamma_{[t]}^C \mathbf{S}_{[t]} + \left(\mathbf{U}_{[t]} - \mathrm{Diag}\left(\gamma_{[t]}\right) \mathbf{W}_{[t]} \mathbf{S}_{[t]}^\intercal \right)^\intercal \mathbf{K}_{[t]} \tag{9}$$

$$\mathbf{O}_{[t]} = \mathrm{Diag}\left(\gamma_{[t]}\right) \mathbf{Q}_{[t]} \mathbf{S}_{[t]}^\intercal + \left(\mathbf{Q}_{[t]} \mathbf{K}_{[t]}^\intercal \odot \Gamma_{[t]} \right) \left(\mathbf{U}_{[t]} - \mathrm{Diag}\left(\gamma_{[t]}\right) \mathbf{W}_{[t]} \mathbf{S}_{[t]}^\intercal \right) \tag{10}$$

UT transform. To maximize hardware efficiency, we apply the UT transform (Joffrain et al., 2006) to Eq. 8. This technique reformulates operations into matmul form, reducing non-matmul FLOPs,

[1] The theoretical distinction lies in the optimization approach: Longhorn uses implicit online learning (Kulis & Bartlett, 2010) to derive closed-form globally optimal updates, while DeltaNet optimizes the same objective through one-step explicit gradient descent, as noted by Liu et al. (2024). Despite Longhorn's stronger theoretical grounding, we found no significant empirical performance differences between these approaches and thus maintain DeltaNet's original formulations.

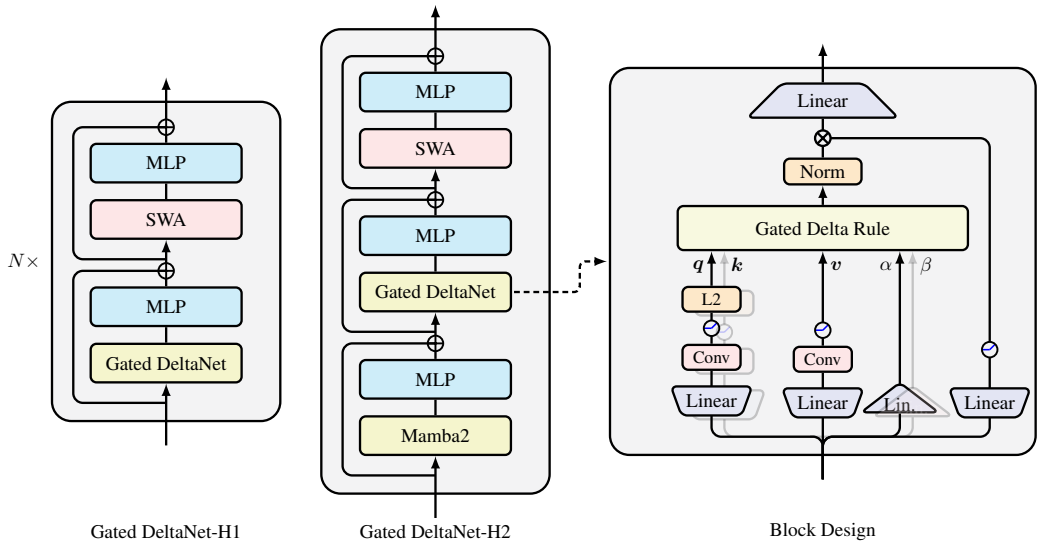

Figure 1: Visualization of the (hybrid) architecture and block design of Gated DeltaNet models. Gated DeltaNet-H1 and H2 use Gated DeltaNet + SWA and Mamba2 + Gated DeltaNet + SWA patterns, respectively. In the block design, query/key paths consist of linear proj., shortconv., SiLU and L2 norm; value path includes linear proj., shortconv. and SiLU; alpha/beta use linear proj.; and output gate applies linear proj. with SiLU.

which is crucial to enable better hardware utilization during training (Dao, 2023; Fu et al., 2023; Yang et al., 2024a).

$$\mathbf{W}_{[t]} = \mathbf{A}^W_{[t]} \operatorname{Diag}(\beta_{[t]})\mathbf{K}_{[t]}, \qquad \mathbf{A}^W_{[t]} = \left(\mathbf{I} - \operatorname{lower}(\operatorname{Diag}(\beta_{[t]})\mathbf{K}_{[t]}\mathbf{K}_{[t]}^\intercal)\right)^{-1}$$

$$\mathbf{U}_{[t]} = \mathbf{A}^U_{[t]} \operatorname{Diag}\left(\beta_{[t]}\right)\mathbf{V}_{[t]}, \qquad \mathbf{A}^U_{[t]} = \left(\mathbf{I} - \operatorname{lower}\left(\operatorname{Diag}(\beta_{[t]})\left(\Gamma_{[t]} \odot \mathbf{K}_{[t]}\mathbf{K}_{[t]}^\intercal\right)\right)\right)^{-1}$$

where $\operatorname{lower}(\cdot) := \operatorname{tril}(\cdot, -1)$; and the inverse of a lower triangle matrix can be calculated efficiently by forward substitution.

Remark on speed. Similar to Mamba2, the gating term (colored in blue) only performs elementwise multiplication with (intermediate) variables without affecting matrix multiply structures, enabling tensor core GPU optimization. As shown in Fig. 3, Gated DeltaNet maintains the same speed as DeltaNet, with only a small performance gap to Mamba2 despite having a more complex and expressive transition matrix.

3.3 Gated Delta Networks and Hybrid Models

Token mixer block. The basic Gated DeltaNet follows Llama's macro architecture, stacking token mixer layers with SwiGLU MLP layers, but replaces self-attention with gated delta rule token mixing. Fig. 1 (right) shows its block design. For the gated delta rule (Eq. 6), queries, keys and values $\{\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}\}$ are generated through linear projection, short convolution and SiLU, with L2 normalization applied to $\boldsymbol{q}, \boldsymbol{k}$ for training stability. α, β use linear projection only.[2] Following Sun et al. (2023a), the output is processed through normalization and gating before applying output projection.

Hybrid models. Linear transformers have limitations in modeling local shifts and comparisons, and their fixed state size makes it hard for retrieval tasks (Arora et al., 2024a). Following recent hybrid architectures like Griffin (De et al., 2024) and Samba (Ren et al., 2024), we combine linear recurrent layers with sliding window attention (SWA), resulting in GatedDeltaNet-H1. We also stack Mamba2, GatedDeltaNet and SWA, resulting in GatedDeltaNet-H2.

[2]We use Mamba2's parameterization for α but omit it for brevity.

Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	SIQA acc ↑	BoolQ acc ↑	Avg.
Recurrent models											
RetNet	19.08	17.27	40.52	70.07	49.16	54.14	67.34	33.78	40.78	60.39	52.02
HGRN2	19.10	17.69	39.54	70.45	49.53	52.80	69.40	35.32	40.63	56.66	51.79
Mamba	17.92	15.06	43.98	71.32	52.91	52.95	69.52	35.40	37.76	61.13	53.12
Mamba2	16.56	12.56	45.66	71.87	55.67	55.24	**72.47**	37.88	40.20	60.13	54.89
DeltaNet	17.71	16.88	42.46	70.72	50.93	53.35	68.47	35.66	40.22	55.29	52.14
Gated DeltaNet	**16.42**	**12.17**	**46.65**	**72.25**	**55.76**	**57.45**	**71.21**	**38.39**	**40.63**	60.24	**55.32**
Attention or hybrid models											
Transformer++	18.53	18.32	42.60	70.02	50.23	53.51	68.83	35.10	40.66	57.09	52.25
Samba	16.13	13.29	44.94	70.94	53.42	55.56	68.81	36.17	39.96	62.11	54.00
Gated DeltaNet-H1	16.07	**12.12**	47.73	**72.57**	56.53	**58.40**	71.75	**40.10**	41.40	**63.21**	**56.40**
Gated DeltaNet-H2	**15.91**	12.55	**48.76**	72.19	**56.88**	57.77	71.33	39.07	**41.91**	61.55	56.18

Table 2: Performance comparison on language modeling and zero-shot common-sense reasoning.

4 EXPERIMENTS

Setup Our experiments encompass a comprehensive comparison of recent state-of-the-art architectures, including pure Transformer models, RNN-based approaches, and hybrid architectures. We evaluate against the following baselines: RetNet (Sun et al., 2023a), HGRN2 (Qin et al., 2024), Mamba (Gu & Dao, 2023), Mamba2 (Dao & Gu, 2024b), Samba (Ren et al., 2024), and DeltaNet (Yang et al., 2024b). For fair comparison, all models are trained under identical conditions with 1.3B parameters on 100B tokens sampled from the FineWeb-Edu dataset (Penedo et al., 2024). We use the AdamW optimizer with a peak learning rate of 4e-4, weight decay of 0.1, and gradient clipping of 1.0. The learning rate follows a cosine annealing schedule with a 1B token warm-up period and batch size of 0.5M tokens. All models employ the LLaMA 2 tokenizer with a vocabulary size of 32,000. For sequence modeling, we set the training length to 4K tokens, with Samba and our hybrid models using a sliding window size of 2K. See appendix for evaluation settings.

Common-sense reasoning In Table 2, we present the language modeling perplexity and **zero-shot** accuracy on commonsense reasoning benchmarks for models with 400M and 1.3B parameters. Gated DeltaNet consistently outperforms other linear models, including RetNet, HGRN2, Mamba, Mamba2, and DeltaNet, at both scales. As expected, the hybrid variant further enhances performance.

	S-NIAH-1 (pass-key retrieval)				S-NIAH-2 (number in haystack)				S-NIAH-3 (word in haystack)		
Model	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K
DeltaNet	97.4	96.8	**99.0**	**98.8**	98.4	45.6	18.6	14.4	85.2	47.0	22.4
Mamba2	**99.2**	**98.8**	65.4	30.4	99.4	98.8	56.2	17.0	64.4	47.6	4.6
Gated DeltaNet	98.4	88.4	91.4	91.8	**100.0**	**99.8**	**92.2**	**29.6**	**86.6**	**84.2**	**27.6**

Table 3: Performance comparison on S-NIAH benchmark suite.

In-context retrieval on synthetic data Table 3 shows the results on Single Needle-In-A-Haystack (S-NIAH) benchmark suite from RULER (Hsieh et al., 2024). In the simplest S-NIAH-1 setting with synthetic inputs, DeltaNet achieves near-perfect performance across all sequence lengths, benefiting from its delta update rule which is specifically advantageous for in-context recall (§3.1). In contrast, Gated DeltaNet shows slightly lower retrieval accuracy since its gating mechanism discards information, compromising perfect memory retention, while Mamba2's performance degrades significantly beyond 2K sequences. However, retrieval from memory depends on not only retention but also the ability to "forget": given fixed state size, lack of memory clearance leads to memory

Models	SWDE	SQD	FDA	TQA	NQ	Drop	Avg
Recurrent models							
RetNet	14.0	28.5	7.0	54.4	16.2	17.3	22.9
HGRN2	8.3	25.3	4.8	51.2	14.2	16.9	20.1
Mamba	9.8	25.8	3.7	54.3	14.9	17.4	21.0
Mamba2	19.1	33.6	**25.3**	**61.0**	**20.8**	19.2	29.8
DeltaNet	17.9	30.9	18.4	53.9	17.3	18.6	26.2
Gated DeltaNet	**25.4**	**34.8**	23.7	60.0	20.0	**19.8**	**30.6**
Attention or hybrid models							
Transformer++	29.5	38.0	**52.2**	58.3	22.5	21.6	37.0
Samba	33.0	39.2	50.5	57.7	23.5	20.2	37.3
Gated DeltaNet-H1	35.6	39.7	52.0	60.1	24.6	22.2	39.0
Gated DeltaNet-H2	**38.2**	**40.4**	50.7	**63.3**	**24.8**	**23.3**	**40.1**

Table 4: Accuracy on recall-world retrieval tasks with input truncated to 2K tokens. SQD: SQUADE. TQA: Trivial QA.

collision when the state becomes saturated - multiple pieces of information become superimposed, making them indistinguishable. This becomes evident in NIAH-2 and NIAH-3 where needles are grounded in real-world text data: DeltaNet's performance degrades significantly, while Gated DeltaNet's adaptive memory management demonstrates clear advantages over both Mamba2 and DeltaNet.

In-context retrieval on real-world data Table 4 presents results on real-world recall-intensive tasks used by Arora et al. (2024b). As expected, linear recurrent models show a significant performance gap compared to Transformers, while hybrid models combining linear recurrence and attention outperform pure attention models in retrieval tasks.

For pure recurrent models, despite DeltaNet's superior performance on synthetic in-context retrieval tasks (Yang et al., 2024b), its real-world retrieval performance lags behind Mamba2, consistent with our observations in S-NIAH-2 and S-NIAH-3 (Table 3). Gated DeltaNet outperforms both DeltaNet and Mamba2 thanks to its gated delta rule, though the improvement margin is smaller than in Table 3. We attribute this reduced performance gap to instruction-unaligned small language models being prone to repetition errors, which are the primary source of errors in these tasks (cf. Arora et al. (2024b, Appendix E)). Since this issue is largely independent of the update rule choice, the performance differences between models are less pronounced compared to Table 3.

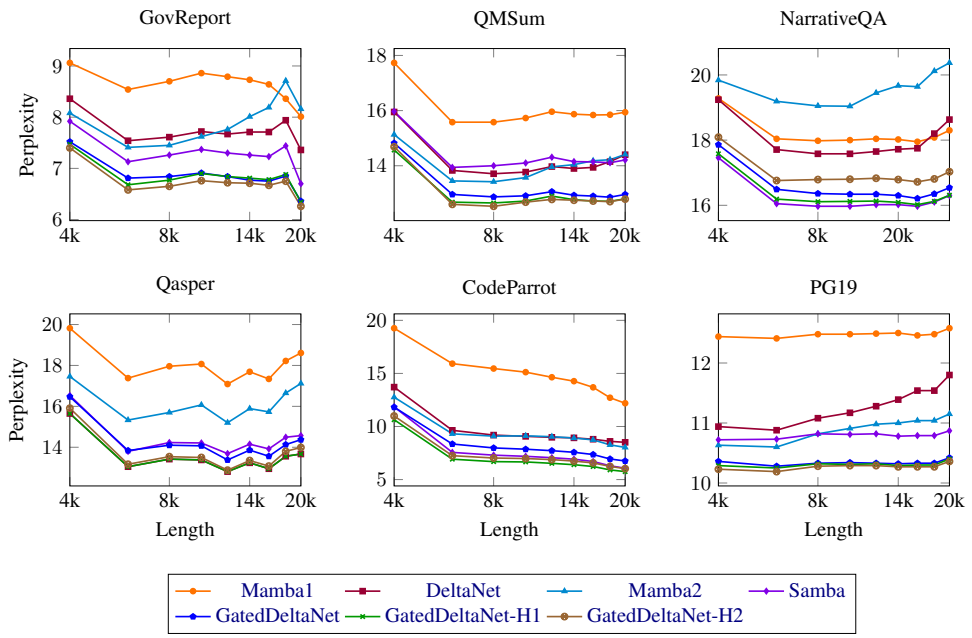

Figure 2: Length extrapolation on six long benchmarks.

Length extrapolation on long sequences. As shown in Fig.2, we evaluate the models' capacity to extrapolate to sequences of up to 20K tokens across six long-context benchmarks. Among RNN architectures, Gated DeltaNet demonstrates superior performance with the lowest overall perplexity and exhibits more stable behavior during length extrapolation, indicating enhanced memory management. The hybrid models further improve upon this by leveraging attention for local context modeling, which reduces the memory management burden on their recurrent components. Future work will explore these models' capabilities on even longer sequences.

Long context understanding As demonstrated in Table 5, we evaluated the models' performance on LongBench (Bai et al., 2023). In recurrent models, Gated DeltaNet shows consistent advantages, especially in single-doc QA and Code tasks.

Throughput Comparison. The training throughput comparison across different models is presented in Fig. 3. As our analysis shows, the proposed gated delta rule introduces only marginal overhead

Model	Single-Doc QA			Multi-Doc QA			Summarization			Few-shot			Code		Avg
	NQA	QQA	MFQ	HQA	2WM	Mus	GvR	QMS	MNs	TRC	TQA	SSM	LCC	RBP	
Recurrent models															
RetNet	12.1	10.7	19.1	10.7	**18.0**	5.8	4.8	15.8	7.9	19.0	18.0	12.8	14.1	17.9	13.2
HGRN2	10.7	12.1	19.1	11.3	15.7	6.0	5.2	15.1	**9.2**	16.0	15.8	10.3	18.6	20.8	13.5
Mamba	13.0	10.1	20.4	10.1	16.7	6.0	7.2	15.9	8.4	23.1	21.9	11.2	17.9	19.0	14.6
DeltaNet	12.9	10.8	21.5	10.9	13.2	5.1	6.5	13.5	7.2	15.5	23.3	11.6	17.6	20.3	13.6
Mamba2	11.1	11.3	18.6	11.8	15.1	**6.7**	6.7	14.5	7.4	13.0	**23.6**	8.4	17.9	20.6	13.5
Gated DeltaNet	**14.1**	**14.0**	**23.3**	**13.7**	14.4	5.8	**7.5**	**16.4**	7.9	**30.0**	22.4	**23.0**	**18.7**	**22.1**	**16.6**
Attention or hyrbid models															
Transformer++	11.8	9.3	10.0	10.9	4.2	6.1	7.4	15.8	6.6	16.9	13.5	3.9	17.2	18.7	11.0
Samba	12.5	12.9	25.4	11.2	19.7	6.8	9.1	15.7	11.0	20.0	22.7	22.8	18.1	21.1	15.9
Gated DeltaNet-H1	**14.5**	12.3	26.6	12.6	**23.6**	6.1	9.1	16.1	12.8	33.5	**23.9**	26.8	15.5	19.2	17.8
Gated DeltaNet-H2	12.7	**13.0**	**27.1**	**12.7**	20.6	**7.5**	**10.4**	**16.2**	**13.0**	**40.5**	22.7	**27.9**	**19.9**	**22.1**	**18.4**

Table 5: Accuracy on 14 tasks from LongBench (Bai et al., 2023): Narrative QA, QasperQA, MultiField QA, HotpotQA, 2WikiMulti QA, Musique, GovReport, QMSum, MultiNews, TRec, Trivia QA, SamSum, LCC, and RepoBench-P by order.

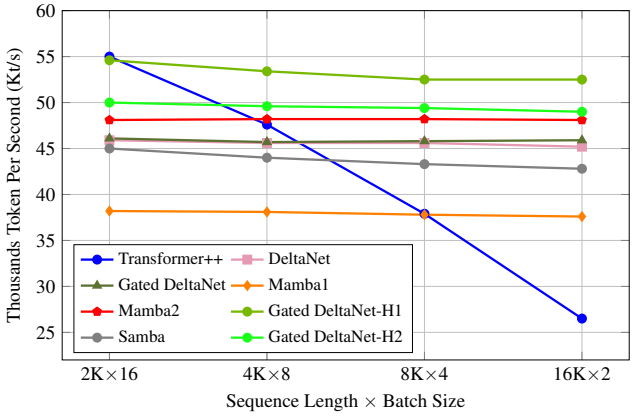

Figure 3: Training throughput comparison of 1.3B models on a single H100 GPU.

compared to the original delta rule, with Gated DeltaNet achieving essentially the same throughput as DeltaNet. Both are slightly slower than Mamba2 (2-3K tokens/sec) due to their more expressive transition matrices.

The Transformer++ achieves the best performance in the 2K context window domain, thanks to the highly optimized Flash-Attention-2 kernel (Dao, 2023). Consequently, hybrid approaches combining 2K window-size SWA attention with other token mixers demonstrate higher throughput than standalone mixers: Samba outperforms Mamba, while Gated DeltaNet-H1 and -H2 outperform Gated DeltaNet. Notably, Gated DeltaNet-H1 maintains compelling training throughput across all sequence lengths, even on short sequences.

5 Related Work

Gated linear RNN. Large linear recurrent language models have attracted significant attention due to their training and inference efficiency. The field of linear RNNs has rapidly evolved from using data-independent decay mechanisms, as exemplified by models like S4 (Gu et al., 2022), S5 (Smith et al., 2023), LRU (Orvieto et al., 2023), RWKV4/5 (Peng et al., 2023), and RetNet (Sun et al., 2023a), to incorporating data-dependent decay mechanisms in more recent architectures such as HGRN1/2 (Qin et al., 2024; 2023b), Mamba1/2 Gu & Dao (2023); Dao & Gu (2024a), RWKV6 (Peng et al., 2024), and GSA (Zhang et al., 2024). This transition stems from the proven advantages of gating/forgetting mechanisms (termed selective mechanisms in Mamba)—a classical concept originating in the gated RNN literature (Gers et al., 2000) whose significance has been consistently reaffirmed (Greff et al., 2015; van der Westhuizen & Lasenby, 2018; Qin et al., 2024; 2023b; Gu & Dao, 2023).

Modern forget gates differ from traditional designs like those in LSTM by removing the dependency on the previous hidden state, relying solely on input data. This modification enables efficient parallelism across sequence lengths (Martin & Cundy, 2018; Qin et al., 2023b). The absence of a forget gate has been a notable limitation in DeltaNet, and our gated extension addresses this gap in a natural and effective way.

Delta rule. The delta learning rule has been shown to offer superior memory capacity compared to the Hebbian learning rule (Gardner, 1988; Prados & Kak, 1989). While linear transformers rely on a Hebbian-like learning rule, DeltaNet utilizes the delta rule, and this advantage in memory capacity is empirically evident in synthetic in-context learning tasks. Moreover, this superiority extends across various applications, including language modeling (Irie et al., 2021; Yang et al., 2024b), reinforcement learning (Irie et al., 2022b), and image generation (Irie & Schmidhuber, 2023). Yang et al. (2024b) further parallelized delta rule computations across sequence lengths and demonstrated the enhanced expressiveness of DeltaNet's transition matrix. Specifically, DeltaNet's data-dependent identity-plus-low-rank structure ($\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\intercal$) offers greater flexibility compared to Mamba2's data-dependent diagonal matrices ($\alpha_t \mathbf{I}$). This architectural shift from diagonal to structured dense matrices substantially improves the model's capabilities in complex reasoning tasks, including regular language recognition (Fan et al., 2024; Grazzi et al., 2024) and state-tracking tasks beyond the TC^0 complexity class (Merrill et al., 2024)—capabilities that are particularly crucial for applications like coding. Recent work by Grazzi et al. (2024) suggests that allowing negative eigenvalues in DeltaNet could further enhance its state tracking capabilities, which could be directly incorporated into Gated DeltaNet as well.

The delta rule shares an intriguing connection with online (meta) learning via gradient descent (Munkhdalai et al., 2019; Irie et al., 2022a). Recent architectures like Longhorn (Liu et al., 2024) and TTT (Sun et al., 2024a) revisit this relationship by reformulating state space learning as a gradient-based online learning problem (see also §3.1). While Longhorn offers a more theoretically rigorous formulation, its reliance on diagonal approximation significantly compromises expressiveness. TTT presents an interesting case: its linear variant without Layernorm is equivalent to DeltaNet, but adding Layernorm transforms it into a non-linear RNN model. This transformation necessitates a hybrid training approach where a "delta-like-rule" is applied at the chunk level every N tokens (where N is the chunk size).

Despite its advantages, the delta rule has theoretical limitations (Irie et al., 2023) and shows moderate performance on real-world datasets (Yang et al., 2024b). Previous extensions enhance expressiveness through strict *nonlinear* recurrence (Irie et al., 2021; 2022b), but sacrifice training parallelism. Our Gated DeltaNet maintains a linear RNN, enabling efficient training while improving expressiveness through gating, leading to consistent improvement across tasks. Future work could adopt GLA-like diagonal gating (Yang et al., 2024a) to further relax gating restrictions.

6 Conclusion

In this work, we introduced Gated DeltaNet, which enables better key-value association learning compared to Mamba2 and more adaptive memory clearance than DeltaNet, leading to consistently better empirical results across various tasks. We extended the parallel algorithm from Yang et al. (2024b) to enable hardware-efficient training of Gated DeltaNet. Our hybrid Gated DeltaNet model achieves even higher training throughput and overall performance, making it well-suited for practical deployment.

References

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-Context Language Learning: Architectures and Algorithms, 2024. URL https://arxiv.org/abs/2401.12973.

Maksym Andriushchenko, Francesco D'Angelo, Aditya Vardhan Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning? *ArXiv*, abs/2310.04415, 2023. URL https://api.semanticscholar.org/CorpusID:263829417.

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. *ArXiv preprint*, abs/2312.04927, 2023a. URL https://arxiv.org/abs/2312.04927.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes, 2023b. URL https://arxiv.org/abs/2304.09433.

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. *ArXiv preprint*, abs/2402.18668, 2024a. URL https://arxiv.org/abs/2402.18668.

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. Just read twice: closing the recall gap for recurrent language models, 2024b. URL https://arxiv.org/abs/2407.05483.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv preprint*, abs/2308.14508, 2023. URL https://arxiv.org/abs/2308.14508.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *ArXiv preprint*, abs/2405.04517, 2024. URL https://arxiv.org/abs/2405.04517.

Christian H. Bischof and Charles Van Loan. The WY representation for products of householder matrices. In *SIAM Conference on Parallel Processing for Scientific Computing*, 1985. URL https://api.semanticscholar.org/CorpusID:36094006.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6239.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

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

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv preprint*, abs/2307.08691, 2023. URL https://arxiv.org/abs/2307.08691.

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

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024b. URL https://proceedings.mlr.press/v235/dao24a.html.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.365. URL https://aclanthology.org/2021.naacl-main.365.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models, 2024. URL https://arxiv.org/abs/2402.19427.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1102. URL https://aclanthology.org/P19-1102.

Ting-Han Fan, Ta-Chung Chi, and Alexander Rudnicky. Advancing regular language reasoning in linear recurrent neural networks. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 45–53, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-short.4.

Daniel Y. Fu, Hermann Kumbong, Eric Nguyen, and Christopher Ré. Flashfftconv: Efficient convolutions for long sequences with tensor cores. *ArXiv preprint*, abs/2311.05908, 2023. URL https://arxiv.org/abs/2311.05908.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2021.

E. Gardner. The space of interactions in neural network models. *Journal of Physics A*, 21:257–270, 1988. URL https://api.semanticscholar.org/CorpusID:15378089.

Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Comput.*, 12(10):2451–2471, 2000.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409.

Riccardo Grazzi, Julien N. Siems, Jorg K. H. Franke, Arber Zela, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear rnns through negative eigenvalues. 2024. URL https://api.semanticscholar.org/CorpusID:274141450.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28: 2222–2232, 2015. URL https://api.semanticscholar.org/CorpusID:3356463.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian J. McAuley. Longcoder: A long-range pre-trained language model for code completion. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12098–12107. PMLR, 2023. URL https://proceedings.mlr.press/v202/guo23j.html.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580.

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

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc V. Le. Transformer quality in linear time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9099–9117. PMLR, 2022. URL https://proceedings.mlr.press/v162/hua22a.html.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.112. URL https://aclanthology.org/2021.naacl-main.112.

Kazuki Irie and Jürgen Schmidhuber. Images as weight matrices: Sequential image generation through synaptic learning rules. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=ddad0PNUvV.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Going beyond linear transformers with recurrent fast weight programmers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 7703–7717, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/3f9e3767ef3b10a0de4c256d7ef9805d-Abstract.html.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9639–9659. PMLR, 2022a. URL https://proceedings.mlr.press/v162/irie22a.html.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. A modern self-referential weight matrix that learns to modify itself. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of*

Machine Learning Research, pp. 9660–9677. PMLR, 2022b. URL https://proceedings.mlr.press/v162/irie22b.html.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. Practical computational power of linear transformers and their recurrent and self-referential extensions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9455–9465, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.588. URL https://aclanthology.org/2023.emnlp-main.588.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat After Me: Transformers are Better than State Space Models at Copying. *ArXiv preprint*, abs/2402.01032, 2024. URL https://arxiv.org/abs/2402.01032.

Thierry Joffrain, Tze Meng Low, Enrique S. Quintana-Ortí, Robert A. van de Geijn, and Field G. Van Zee. Accumulating householder transformations, revisited. *ACM Trans. Math. Softw.*, 32:169–179, 2006. URL https://api.semanticscholar.org/CorpusID:15723171.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017b. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020a. URL http://proceedings.mlr.press/v119/katharopoulos20a.html.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020b. URL http://proceedings.mlr.press/v119/katharopoulos20a.html.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023.

Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *Neural Information Processing Systems*, 1991. URL https://api.semanticscholar.org/CorpusID:10137788.

Brian Kulis and Peter L. Bartlett. Implicit online learning. In Johannes Fürnkranz and Thorsten Joachims (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pp. 575–582. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/429.pdf.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1150.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. @articleDBLP:journals/corr/abs-2407-14207, author = Bo Liu and Rui Wang and Lemeng Wu and Yihao Feng and Peter Stone and Qiang Liu, title = Longhorn: State Space Models are Amortized Online Learners, journal = CoRR, volume = abs/2407.14207, year = 2024, url = https://doi.org/10.48550/arXiv.2407.14207, doi = 10.48550/ARXIV.2407.14207, eprinttype = arXiv, eprint = 2407.14207, timestamp = Fri, 23 Aug 2024 08:12:16 +0200, biburl = https://dblp.org/rec/journals/corr/abs-2407-14207.bib, bibsource = dblp computer science bibliography, https://dblp.org : State space models are amortized online learners. *ArXiv preprint*, abs/2407.14207, 2024. URL https://arxiv.org/abs/2407.14207.

Tianyang Liu, Canwen Xu, and Julian McAuley. RepoBench: Benchmarking repository-level code auto-completion systems. *ArXiv preprint*, abs/2306.03091, 2023. URL https://arxiv.org/abs/2306.03091.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. OpenCeres: When open information extraction meets the semi-structured web. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3047–3056, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1309. URL https://aclanthology.org/N19-1309.

Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HyUNwulC-.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

William Merrill, Jackson Petty, and Ashish Sabharwal. The Illusion of State in State-Space Models, 2024. URL https://arxiv.org/abs/2404.08819.

Tsendsuren Munkhdalai, Alessandro Sordoni, Tong Wang, and Adam Trischler. Metalearned neural memory. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13310–13321, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/182bd81ea25270b7d1c2fe8353d17fe6-Abstract.html.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Çaglar Gülçehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26670–26698. PMLR, 2023. URL https://proceedings.mlr.press/v202/orvieto23a.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *ArXiv preprint*, abs/2406.17557, 2024. URL https://arxiv.org/abs/2406.17557.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, Przemysław Kazienko, Kranthi Kiran GV, Jan Kocoń, Bartłomiej Koptyra, Satyapriya Krishna, Ronald McClelland Jr., Niklas Muennighoff, Fares Obeid, Atsushi Saito, Guangyu Song, Haoqin Tu, Stanisław Woźniak, Ruichong Zhang, Bingchen Zhao, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. Eagle and Finch: RWKV with Matrix-Valued States and Dynamic Recurrence, 2024. URL https://arxiv.org/abs/2404.05892.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=QtTKTdVrFBB.

DL Prados and SC Kak. Neural network capacity using delta rule. *Electronics Letters*, 3(25):197–199, 1989.

Zhen Qin, Dong Li, Weigao Sun, Weixuan Sun, Xuyang Shen, Xiaodong Han, Yunshen Wei, Baohong Lv, Fei Yuan, Xiao Luo, Y. Qiao, and Yiran Zhong. Transnormerllm: A faster and better large language model with improved transnormer. 2023a. URL https://api.semanticscholar.org/CorpusID:260203124.

Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/694be3548697e9cc8999d45e8d16fe1e-Abstract-Conference.html.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion. *ArXiv preprint*, abs/2404.07904, 2024. URL https://arxiv.org/abs/2404.07904.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *ArXiv preprint*, abs/2406.07522, 2024. URL https://arxiv.org/abs/2406.07522.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4463–4473, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021a. URL http://proceedings.mlr.press/v139/schlag21a.html.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021b. URL http://proceedings.mlr.press/v139/schlag21a.html.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=Ai8Hw3AXqks.

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artif. Intell.*, 46(1-2):159–216, 1990. doi: 10.1016/0004-3702(90)90007-M. URL https://doi.org/10.1016/0004-3702(90)90007-M.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): Rnns with expressive hidden states. *ArXiv preprint*, abs/2407.04620, 2024a. URL https://arxiv.org/abs/2407.04620.

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*, abs/2307.08621, 2023a. URL https://arxiv.org/abs/2307.08621.

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*, abs/2307.08621, 2023b. URL https://arxiv.org/abs/2307.08621.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models. *ArXiv preprint*, abs/2405.05254, 2024b. URL https://arxiv.org/abs/2405.05254.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31.

Jos van der Westhuizen and Joan Lasenby. The unreasonable effectiveness of the forget gate. *ArXiv preprint*, abs/1804.04849, 2018. URL https://arxiv.org/abs/1804.04849.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. RNNs are not Transformers (Yet): The Key Bottleneck on In-context Retrieval. *ArXiv preprint*, abs/2402.18510, 2024. URL https://arxiv.org/abs/2402.18510.

Bernard Widrow, Marcian E Hoff, et al. Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pp. 96–104. New York, 1960.

Songlin Yang and Yu Zhang. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism, 2024.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024a. URL https://proceedings.mlr.press/v235/yang24ab.html.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *NeurIPS*, 2024b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, Peng Zhou, and Guohong Fu. Gated slot attention for efficient linear-time sequence modeling. 2024. URL https://api.semanticscholar.org/CorpusID:272593079.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5905–5921, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472.

A APPENDIX

A.1 EXTENDED WY REPRESENTATION FOR GATED DELTA RULE

To reduce notation clutter, we only consider the first chunk here.

For $\mathbf{S}_t$, the extended WY representation is

$$\mathbf{S}_t = \sum_{i=1}^{t} \frac{\gamma_t}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal, \qquad \mathbf{u}_t = \beta_t \left(\boldsymbol{v}_t - \sum_{i=1}^{t-1} \frac{\gamma_t}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^T \boldsymbol{k}_t \right)$$

We proof this by mathmetical induction.

Proof.

$$\begin{aligned}
\mathbf{S}_{t+1} &= \mathbf{S}_t \left(\alpha_{t+1} (\mathbf{I} - \beta_{t+1} \boldsymbol{k}_{t+1} \boldsymbol{k}_{t+1}^\intercal) \right) + \beta_{t+1} \boldsymbol{v}_{t+1} \boldsymbol{k}_{t+1}^\intercal \\
&= \alpha_{t+1} (\sum_{i=1}^{t} \frac{\gamma_t}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal) - \alpha_{t+1} \beta_{t+1} (\sum_{i=1}^{t} \frac{\gamma_t}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal \boldsymbol{k}_i \boldsymbol{k}_{t+1}^\intercal) + \beta_{t+1} \boldsymbol{v}_{t+1} \boldsymbol{k}_{t+1}^\intercal \\
&= \sum_{i=1}^{t} \frac{\gamma_{t+1}}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal + \underbrace{\beta_{t+1} \left(\boldsymbol{v}_{t+1} - \sum_{i=1}^{t} \frac{\gamma_{t+1}}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^T \boldsymbol{k}_{t+1} \right)}_{\mathbf{u}_{t+1}} \boldsymbol{k}_{t+1}^\intercal \\
&= \sum_{i=1}^{t} \frac{\gamma_{t+1}}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal + \underbrace{\frac{\gamma_{t+1}}{\gamma_{t+1}}}_{1} \mathbf{u}_{t+1} \boldsymbol{k}_{t+1}^\intercal \\
&= \sum_{i=1}^{t+1} \frac{\gamma_{t+1}}{\gamma_i} \mathbf{u}_i \boldsymbol{k}_i^\intercal
\end{aligned}$$

□

For $\mathbf{P}_t$,

$$\begin{aligned}
\mathbf{P}_t &= \prod_{i=1}^{t} \alpha_t \left(\mathbf{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\intercal \right) \\
&= \underbrace{\left(\prod_{i=1}^{t} \alpha_t \right)}_{\gamma_t} \underbrace{\left(\prod_{i=1}^{t} (\mathbf{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\intercal) \right)}_{\mathbf{I} - \sum_{i=1}^{t} \mathbf{w}_i \boldsymbol{k}_i^\intercal}
\end{aligned}$$

and

$$\prod_{i=1}^{t} (\mathbf{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\intercal) = \mathbf{I} - \sum_{i=1}^{t} \mathbf{w}_i \boldsymbol{k}_i^\intercal, \quad \mathbf{w}_n = \beta_n \boldsymbol{k}_n - \beta_n \sum_{t=1}^{n-1} (\mathbf{w}_t (\boldsymbol{k}_t^\intercal \boldsymbol{k}_n)$$

has already been proved in Yang et al. (2024b).

A.2 ABLATION STUDY AND ADDITIONAL EXPERIMENTS

Table S.1 presents ablation studies on the Gated DeltaNet block's components. Our experiments demonstrate that both the short convolution and output gate are crucial for model performance, while output normalization yields marginal improvements. Consistent with Yang et al. (2024b), we found L2 normalization to be essential for optimal performance, though the choice of feature map was less influential. Nevertheless, SiLU consistently outperformed other activation functions, aligning with observations from Qin et al. (2023a). Through empirical analysis, we determined that a head dimension of 128 provides an optimal trade-off between performance and computational efficiency.

Table S.1: Ablation study on the Gated DeltaNet block. Avg-PPL and Avg-Acc denote average perplexity and zero-shot commonsense reasoning accuracy (as in Table 2), respectively. All models have 400M parameters and are trained for 15B tokens on the same subset of FineWeb-Edu dataset (Penedo et al., 2024).

Gated DeltaNet Ablations (400M)	Avg-PPL (↓)	Avg-Acc (↑)
Gated DeltaNet *w* Head Dim 128,	27.35	47.26
Macro Design		
w. naive Delta Rule	30.87	45.12
w/o. Short Conv	28.95	46.16
w/o. Output Gate	29.12	45.46
w/o. Output Norm	27.55	47.07
Normalization & Feature Map		
w. L_1-norm & ReLU	30.79	45.92
w. L_1-norm & 1+ELU	30.34	46.05
w. L_1-norm & SiLU	30.18	46.09
w. L_2-norm & ReLU	27.67	46.94
w. L_2-norm & 1+ELU	27.58	47.17
Model Dimensions		
w. Head Dim 64	28.31	46.35
w. Head Dim 256	27.13	47.38

Model	**Wiki.** ppl ↓	**LMB.** ppl ↓	**LMB.** acc ↑	**PIQA** acc ↑	**Hella.** acc_n ↑	**Wino.** acc ↑	**ARC-e** acc ↑	**ARC-c** acc_n ↑	**SIQA** acc ↑	**BoolQ** acc ↑	**Avg.**
Hybrid Ablations (500M/15B)											
Gated DeltaNet + SWA + Mamba2	24.02	28.20	34.77	67.08	40.84	50.74	60.35	28.83	38.94	61.49	47.88
Gated DeltaNet + Mamba2 + SWA	23.69	26.83	36.17	67.51	41.51	51.85	61.19	29.77	38.58	53.73	47.54
Mamba2 + SWA + Gated DeltaNet	24.14	25.21	36.79	64.96	41.18	52.01	60.90	30.03	38.07	59.44	47.92
Mamba2 + Gated DeltaNet + SWA	**23.54**	**24.11**	36.92	66.48	41.70	52.72	61.06	30.54	39.91	60.51	**48.73**

Table S.2: Ablation studies of Gated DeltaNet models. All evaluations are performed by using `lm-evaluation-harness` (Gao et al., 2021). All models use the Llama tokenizer and are trained on the same subset of the FineWeb-Edu dataset (Penedo et al., 2024).

Models	**State size**	**SWDE** ↑	**SQuAD** ↑	**FDA** ↑	**TriviaQA** ↑	**NQ** ↑	**Drop** ↑	**Avg**
400M params / 15B tokens								
Transformer++	N/A	<u>22.1</u>	28.3	30.2	43.1	15.6	17.5	26.1
Samba	$2062 \times Ld$	**23.1**	29.9	31.0	45.1	16.3	16.7	27.0
RetNet	$512 \times Ld$	6.0	19.6	1.5	39.4	8.7	14.9	15.0
HGRN2	$128 \times Ld$	6.1	15.3	1.0	36.9	7.6	12.1	13.1
Mamba	$32 \times Ld$	6.8	15.7	1.1	37.8	8.0	12.2	13.6
Mamba2	$256 \times Ld$	12.0	24.9	10.8	43.3	11.8	17.3	20.1
DeltaNet	$128 \times Ld$	7.4	22.4	6.5	41.8	12.3	16.7	17.8
Gated DeltaNet	$128 \times Ld$	11.3	26.0	4.5	42.2	10.2	18.0	18.7
Gated DeltaNet	$256 \times Ld$	13.6	26.5	9.8	48.3	13.7	16.0	21.3
Gated DeltaNet-H2	$1418 \times Ld$	20.1	<u>31.8</u>	**41.0**	<u>48.9</u>	<u>17.5</u>	**19.1**	**29.7**
Gated DeltaNet-H1	$2112 \times Ld$	20.7	**33.2**	<u>33.1</u>	**49.8**	**19.5**	<u>18.9</u>	<u>29.2</u>

Table S.3: Results on real-world in-context retrieval tasks for 0.4B models trained for 15B tokens.

Additionally, Table S.2 demonstrates that among various hybrid architectures, the combination of Mamba2, Gated DeltaNet, and SWA in this specific order produces superior results.

Tables S.4 and S.3 present zero-shot performance on common-sense reasoning and retrieval tasks for smaller-scale models. The results in Table S.3 demonstrate that Gated DeltaNet with half the state size of Mamba2 shows lower performance than Mamba2, while outperforming Mamba2 when using the same state size. This highlights the critical role of state size in RNN model performance in retrieval task.

Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	SIQA acc ↑	BoolQ acc ↑	Avg.
Transformer++	30.63	37.37	29.64	64.27	37.72	51.53	54.95	27.36	38.07	61.59	45.64
RetNet	29.92	46.83	29.16	65.23	36.97	51.85	56.01	27.55	37.30	59.66	45.47
HGRN2	32.33	47.14	26.12	64.52	35.45	52.24	55.97	25.51	37.35	59.02	44.52
Mamba	29.22	39.88	29.82	65.72	37.93	50.11	58.37	26.70	37.76	61.13	45.94
Mamba2	26.34	33.19	32.03	65.77	39.73	**52.48**	59.00	27.64	37.92	60.72	46.91
DeltaNet	27.69	44.04	29.96	64.52	37.03	50.82	56.77	27.13	38.22	60.09	45.57
Gated DeltaNet	25.47	29.24	34.40	65.94	40.46	51.46	59.80	28.58	37.43	60.03	47.26
Gated DeltaNet-H2	24.19	**28.09**	**36.77**	**66.43**	**40.79**	52.17	59.55	**29.09**	**39.04**	58.56	47.69
Gated DeltaNet-H1	**24.06**	28.72	36.00	65.50	40.73	51.30	**60.69**	28.49	37.71	**61.77**	**47.88**

Table S.4: Zero-shot common sense reasoning results on 0.4B models trainsed for 15B tokens.

B EXPERIMENTAL SETTINGS

B.1 EVALUATION

Commonsense reasoning Following Gu & Dao (2023), we evaluate our model on multiple commonsense reasoning benchmarks: PIQA (Bisk et al., 2020), HellaSwag (Hella.; Zellers et al., 2019), WinoGrande (Wino.; Sakaguchi et al., 2020), ARC-easy (ARC-e) and ARC-challenge (ARC-c) (Clark et al., 2018), SIQA (Sap et al., 2019), BoolQ (Clark et al., 2019), Wikitext (Wiki.; Merity et al., 2017), and LAMBADA (LMB.; Paperno et al., 2016).

In-context retrieval Our evaluation comprises both synthetic and real-world tasks. For synthetic tasks, we utilize the Needle-In-A-Haystack Single (NIAH-S) benchmark suite from RULER (Hsieh et al., 2024), which includes three increasingly complex tasks: S-NIAH-1 (passkey retrieval), S-NIAH-2 (numerical needle in haystack), and S-NIAH-3 (word-based needle in haystack). For real-world tasks, following Arora et al. (2024b), we evaluate on diverse datasets: SWDE (Lockard et al., 2019) for structured HTML relation extraction, FDA (Arora et al., 2023b) for PDF key-value retrieval, and several question-answering datasets including SQuAD (Rajpurkar et al., 2018), TriviaQA (Joshi et al., 2017a), Drop (Dua et al., 2019), and NQ (Kwiatkowski et al., 2019). Since our pretrained models lack instruction tuning, we employ the Cloze Completion Formatting prompts provided by Arora et al. (2024b), which better align with our models' next-word-prediction training objective.

Long context understanding We evaluate on 14 tasks from Longbench (Bai et al., 2023), encompassing: narrative comprehension (Narrative QA (Kočiský et al., 2018)), scientific understanding (QasperQA (Dasigi et al., 2021)), multi-hop reasoning (MultiField QA, HotpotQA (Yang et al., 2018), 2WikiMulti QA (Ho et al., 2020), Musique (Trivedi et al., 2022)), document summarization (GovReport (Huang et al., 2021), QMSum (Zhong et al., 2021), MultiNews (Fabbri et al., 2019)), and various specialized tasks (TRec (Li & Roth, 2002), Trivia QA (Joshi et al., 2017b), SamSum (Gliwa et al., 2019), LCC (Guo et al., 2023), and RepoBench-P (Liu et al., 2023)).