INTEGRAL DELTANET: IMPROVED LINEAR ATTENTION WITH ADAPTIVE MEMORY INTEGRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear attention replaces softmax attention with a linear formulation, enabling linear-time computation with a fixed-size recurrent state instead of a growing key–value cache. These computational and memory advantages have attracted growing interest in linear attention for long-context language modeling. However, compressing past information into a fixed-size state causes *memory collisions* that hinder long-context understanding. Although DeltaNet mitigates memory collisions through delta-rule updates and Gated DeltaNet extends this approach with gating, these mechanisms may also discard information relevant to future predictions. To address this limitation, we introduce *Integral DeltaNet*, which maintains two complementary recurrent states: a fast-updating *delta state* that captures current key–value associations and an *integral state* that preserves a weighted history of the delta state. At readout, the model adaptively combines these states to access information across different timescales. We further derive a chunkwise parallel formulation that enables hardware-efficient training. We evaluate Integral DeltaNet using 340M and 1.3B parameter models trained from scratch on 15B and 100B FineWeb-Edu tokens, respectively. Comparisons with strong recurrent baselines show overall improvements in language modeling, commonsense reasoning, real-world and synthetic retrieval, long-context understanding, and length extrapolation. Training throughput remains close to that of the backbones.

1 INTRODUCTION

Most modern large language models (LLMs) are built on the Transformer architecture (Vaswani et al., 2017), whose softmax attention enables effective sequence modeling and hardware-efficient parallel training on GPUs. However, its computation scales quadratically with context length, and the key–value cache grows linearly during autoregressive inference. The demand for longer contexts has motivated alternatives with linear computational complexity and constant memory usage in sequence length. Linear attention (Katharopoulos et al., 2020) achieves this through a kernelized dot-product formulation that admits a matrix-valued linear recurrence, supporting parallel training and efficient recurrent inference. Although early linear attention models (Choromanski et al., 2021; Peng et al., 2021) struggled to match Transformers in language modeling, recent variants have become increasingly competitive and have been incorporated into frontier-scale models such as Qwen3-Next (Qwen Team, 2025) and Kimi K3 (Kimi Team, 2026).

Recent linear attention architectures control memory through decay or gating, as in RetNet (Sun et al., 2023), Mamba-2 (Dao & Gu, 2024), and Gated Linear Attention (Yang et al., 2024a), or delta-rule updates, as in DeltaNet (Schlag et al., 2021; Yang et al., 2024b). These mechanisms mitigate *memory collisions* caused by compressing many key–value associations into a fixed-size state (Schlag et al., 2021). Gating reduces previously stored associations, whereas the delta rule performs targeted updates by removing the value currently associated with an incoming key before writing the new association. Gated DeltaNet (GDN) (Yang et al., 2025) combines these mechanisms, while Kimi Delta Attention (KDA) (Kimi Team, 2025) and Gated DeltaNet-2 (GDN-2) (Hatamizadeh et al., 2026) introduce more expressive updates. However, both decay and targeted overwriting can discard information needed over long contexts. In this work, we address the following question:

Can we mitigate memory collisions while better retaining information needed for the future?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

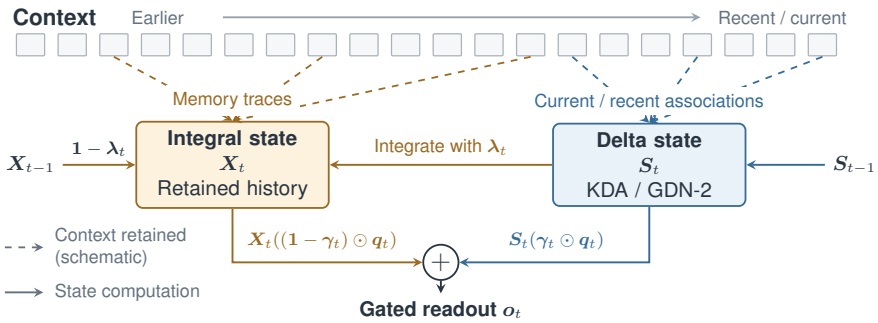


Figure 1: **Integral DeltaNet.** The integral state accumulates a weighted history of delta states. At readout, the model adaptively combines both states to access information across timescales.

We propose *Integral DeltaNet*, which addresses this limitation using two states operating at different timescales: a fast *delta state* S_t, updated using an expressive linear-attention rule such as Kimi Delta Attention or Gated DeltaNet-2, and a slow *integral state* X_t, which maintains a channel-wise weighted history of past delta states. As illustrated in Figure 1, the model adaptively combines the two states at readout, enabling it to access information across multiple timescales.

We derive a chunkwise parallel formulation for Integral DeltaNet that enables hardware-efficient training and use it to train 340M- and 1.3B-parameter models from scratch on 15B and 100B tokens, respectively. We compare these models with strong recurrent baselines and Transformer++ on language modeling, commonsense reasoning, and real-world retrieval at both scales. We further evaluate the 1.3B models on tasks requiring longer contexts: synthetic retrieval, long-context understanding, and length extrapolation. Our results show improvements over the corresponding baselines across multiple tasks, with training throughput close to that of the backbones.

2 Preliminaries

2.1 Linear Attention

Linear attention (Katharopoulos et al., 2020) replaces softmax-normalized query–key similarities with a kernelized dot product, yielding the output at position t as

$$\boldsymbol{o}_t = \sum_{i=1}^{t} \boldsymbol{v}_i(\boldsymbol{k}_i^\top \boldsymbol{q}_t) = \left(\sum_{i=1}^{t} \boldsymbol{v}_i \boldsymbol{k}_i^\top\right) \boldsymbol{q}_t \in \mathbb{R}^{d_v},$$

where $\boldsymbol{k}_t, \boldsymbol{q}_t \in \mathbb{R}^{d_k}$ and $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$ denote the key, query, and value vectors, respectively. By maintaining a prefix state $\boldsymbol{S}_t$, the same computation can be written in recurrent form as

$$\boldsymbol{S}_0 = \boldsymbol{0}, \qquad \boldsymbol{S}_t = \boldsymbol{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\top \in \mathbb{R}^{d_v \times d_k}, \qquad \boldsymbol{o}_t = \boldsymbol{S}_t \boldsymbol{q}_t \in \mathbb{R}^{d_v}. \tag{1}$$

For a sequence of length L, the equivalent masked matrix form is

$$\boldsymbol{O} = \left(\boldsymbol{Q}\boldsymbol{K}^\top \odot \boldsymbol{M}\right) \boldsymbol{V} \in \mathbb{R}^{L \times d_v},$$

where $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{L \times d_k}$ and $\boldsymbol{V}, \boldsymbol{O} \in \mathbb{R}^{L \times d_v}$ are the query, key, value, and output matrices, respectively. The causal mask $\boldsymbol{M} \in \{0,1\}^{L \times L}$ satisfies $M_{t,i} = 1$ for $i \le t$ and $M_{t,i} = 0$ otherwise.

Chunkwise Parallel Training. The recurrent form scales linearly with the sequence length L but processes tokens sequentially, whereas the fully parallel form exploits efficient matrix multiplications at the cost of materializing an $L \times L$ interaction matrix. Chunkwise training (Hua et al., 2022; Sun et al., 2023; Yang et al., 2024b) bridges this gap by partitioning the sequence into chunks of size C, evaluating token interactions in parallel within each chunk, and propagating only a fixed-size state across chunks. Let $N = \lceil L/C \rceil$ be the number of chunks. For chunk $[t]$, where $t \in \{0, \dots, N-1\}$, let $\boldsymbol{Q}_{[t]}, \boldsymbol{K}_{[t]} \in \mathbb{R}^{C \times d_k}$ and $\boldsymbol{V}_{[t]}, \boldsymbol{O}_{[t]} \in \mathbb{R}^{C \times d_v}$ denote the row-stacked query, key, value, and output matrices, and let $\boldsymbol{S}_{[t]} := \boldsymbol{S}_{tC}$ denote the incoming state. Then, chunkwise form can be written as

$$\boldsymbol{O}_{[t]} = \boldsymbol{Q}_{[t]} \boldsymbol{S}_{[t]}^\top + \left(\boldsymbol{Q}_{[t]} \boldsymbol{K}_{[t]}^\top \odot \boldsymbol{M}_C\right) \boldsymbol{V}_{[t]}, \qquad \boldsymbol{S}_{[t+1]} = \boldsymbol{S}_{[t]} + \boldsymbol{V}_{[t]}^\top \boldsymbol{K}_{[t]},$$

where $\boldsymbol{M}_C \in \{0,1\}^{C\times C}$ is the causal mask within each chunk. This reduces the sequential depth from L to approximately L/C without constructing a global attention matrix. The same principle extends to many linear attention variants and underlies FlashLinearAttention (Yang & Zhang, 2024), whose fused chunkwise kernels enable efficient training on modern accelerators.

2.2 Memory Collision

Linear attention compresses all past key–value associations into the fixed-size state $\boldsymbol{S}_t$. Assuming unit-normalized keys, retrieving the value associated with $\boldsymbol{k}_j$ gives

$$\boldsymbol{S}_t\boldsymbol{k}_j = \boldsymbol{v}_j + \sum_{i=1,i\neq j}^{t} (\boldsymbol{k}_i^\top \boldsymbol{k}_j)\boldsymbol{v}_i.$$

The second term captures interference from other associations and vanishes when all other keys are orthogonal to $\boldsymbol{k}_j$; otherwise, it is generally nonzero. Therefore, storing a growing number of associations in the same state inevitably causes *memory collision*, particularly over long contexts.

DeltaNet mitigates this interference using a delta-rule update with input-dependent strength $\beta_t = \mathrm{sigmoid}(\boldsymbol{w}_\beta^\top \boldsymbol{h}_t)$, where $\boldsymbol{h}_t$ is the input hidden state. This gate controls both erasure of the previously stored value and writing of the new key–value association:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\big(\boldsymbol{I} - \underbrace{\beta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top}_{\text{erase old}}\big) + \underbrace{\beta_t\boldsymbol{v}_t\boldsymbol{k}_t^\top}_{\text{write new}} = \boldsymbol{S}_{t-1} + \beta_t\left(\boldsymbol{v}_t - \boldsymbol{S}_{t-1}\boldsymbol{k}_t\right)\boldsymbol{k}_t^\top. \tag{2}$$

Mamba-2 takes a complementary approach, using a data-dependent decay to suppress old information, parameterized as $\alpha_t = \exp\big(-\mathrm{softplus}(\boldsymbol{w}_\Delta^\top \boldsymbol{h}_t + b_\Delta)\exp(a)\big)$ with a learnable scalar a per head:

$$\boldsymbol{S}_t = \alpha_t\boldsymbol{S}_{t-1} + \boldsymbol{v}_t\boldsymbol{k}_t^\top. \tag{3}$$

Gated DeltaNet combines these complementary approaches, coupling data-dependent decay with the targeted overwriting of the delta rule:

$$\boldsymbol{S}_t = \alpha_t\boldsymbol{S}_{t-1}\left(\boldsymbol{I} - \beta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top\right) + \beta_t\boldsymbol{v}_t\boldsymbol{k}_t^\top. \tag{4}$$

Kimi Delta Attention and Gated DeltaNet-2 further increase the expressivity of this framework through channel-wise update mechanisms: Kimi Delta Attention replaces scalar decay with per-channel gating, and Gated DeltaNet-2 introduces separate channel-wise erase and write gates.

These mechanisms reduce memory collisions, but information removed by decay or overwriting may later become important and cannot be recovered from the recurrent state. In this work, we address this trade-off between reducing interference and preserving potentially useful memory.

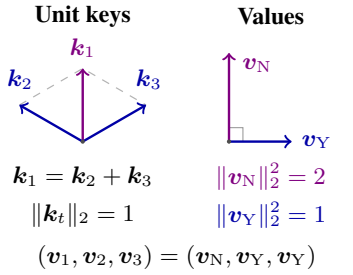
Illustrative Example. To illustrate this trade-off, we consider three sequential key–value pairs $(\boldsymbol{k}_t, \boldsymbol{v}_t)_{t=1}^3$ with keys $\boldsymbol{k}_t \in \mathbb{R}^2$. Let $\boldsymbol{v}_\mathrm{Y}, \boldsymbol{v}_\mathrm{N} \in \mathbb{R}^{d_v}$, $d_v \geq 2$, represent Yes and No, respectively. We choose the keys and values to satisfy

$$\|\boldsymbol{k}_t\|_2 = 1, \qquad \boldsymbol{k}_1 = \boldsymbol{k}_2 + \boldsymbol{k}_3,$$
$$\boldsymbol{v}_\mathrm{Y} \perp \boldsymbol{v}_\mathrm{N}, \quad \|\boldsymbol{v}_\mathrm{Y}\|_2^2 = 1, \quad \|\boldsymbol{v}_\mathrm{N}\|_2^2 = 2,$$
$$(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) = (\boldsymbol{v}_\mathrm{N}, \boldsymbol{v}_\mathrm{Y}, \boldsymbol{v}_\mathrm{Y}).$$

We compare the state updates for linear attention (1), Mamba-2 (3), DeltaNet (2), and Gated DeltaNet (4). Each α_t, β_t may vary across writes within $(0, 1]$. Given the final memory state $\boldsymbol{S}_3$, we sample a query uniformly from the stored keys:

$$\boldsymbol{q}_3 \sim \mathrm{Unif}\{\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3\}.$$

Each model $\mathcal{M}$ (Linear Attention, Mamba-2, DeltaNet, Gated DeltaNet) reads $\boldsymbol{S}_3\boldsymbol{q}_3$ and predicts $\widehat{y}_3^{\mathcal{M}} = \arg\max_{y\in\{\mathrm{Y},\mathrm{N}\}} \boldsymbol{v}_y^\top \boldsymbol{S}_3\boldsymbol{q}_3$. The correct answers are $y(\boldsymbol{k}_1) = \mathrm{N}$ and $y(\boldsymbol{k}_2) = y(\boldsymbol{k}_3) = \mathrm{Y}$. We define the retrieval risk as $\mathcal{R}_\mathcal{M} = \Pr_{\boldsymbol{q}_3}[\widehat{y}_3^{\mathcal{M}} \neq y(\boldsymbol{q}_3)]$, where the probability is over the uniformly sampled query.



Query	$\boldsymbol{k}_1$	$\boldsymbol{k}_2$	$\boldsymbol{k}_3$	$\mathcal{R}$
Correct	N	Y	Y	
Linear Attention	N	N	N	$\frac{2}{3}$
Mamba-2 / DeltaNet / Gated DeltaNet	Y	Y	Y	$\frac{1}{3}$

Figure 2: Illustrative example and retrieval performance.

Below in Proposition 1, we show that there exists a trade-off between memory collision and forgetting.

Proposition 1. *Under the setup above, the retrieval risks satisfy*

$$\mathcal{R}_{\textit{Linear Attention}} = \frac{2}{3}, \qquad \min_{\{\alpha_t\}} \mathcal{R}_{\textit{Mamba-2}} = \min_{\{\beta_t\}} \mathcal{R}_{\text{DeltaNet}} = \min_{\{\alpha_t,\beta_t\}} \mathcal{R}_{\textit{Gated DeltaNet}} = \frac{1}{3}. \tag{5}$$

In Linear Attention, the initial No association dominates the readouts for both later Yes keys, causing two retrieval errors. Mamba-2 and DeltaNet can reduce this interference through decay and targeted overwriting, respectively, and recover both later Yes answers. However, both models forget the first No answer, illustrating how suppressing interference can weaken earlier information still needed for retrieval.

Gated DeltaNet combines decay and overwriting, but it still cannot recover both later Yes answers without forgetting the first No answer. This limitation is inherent to the structure of our example: since $\boldsymbol{k}_1 = \boldsymbol{k}_2 + \boldsymbol{k}_3$, any single-state linear readout satisfies $\boldsymbol{S}_3\boldsymbol{k}_1 = \boldsymbol{S}_3\boldsymbol{k}_2 + \boldsymbol{S}_3\boldsymbol{k}_3$. Hence, two readouts favoring Yes also make the first favor Yes. Thus, even with optimal gates, Gated DeltaNet cannot achieve zero retrieval error in this example; its minimum risk remains $1/3$. Although this limitation follows from the chosen key structure, the example provides a concrete illustration of the trade-off discussed above: the same updates that reduce memory collisions can also cause forgetting of earlier associations that remain relevant to subsequent queries. The full proof is in Appendix A.1.

3 Integral DeltaNet

In this work, we propose *Integral DeltaNet* to reduce memory interference while preserving useful information from distant past. Figure 1 illustrates the central idea: maintain an adaptive *delta state* for the current context alongside an *integral state* that retains a weighted history of earlier delta states. Reading both states gives the model access to information that subsequent updates may have weakened. The two-state recurrence also supports hardware-efficient chunkwise parallel training.

3.1 Delta States and Integral States

Our Integral DeltaNet consists of a fast delta state $\boldsymbol{S}_t$ and a slower integral state $\boldsymbol{X}_t$, defined below. We update $\boldsymbol{S}_t$ using the expressive channel-wise rules of Kimi Delta Attention and Gated DeltaNet-2:

$$\begin{aligned} \text{KDA:} \quad & \boldsymbol{S}_t = \boldsymbol{S}_{t-1}\text{Diag}\left(\boldsymbol{\alpha}_t\right)\left(\boldsymbol{I} - \beta_t \boldsymbol{k}_t\boldsymbol{k}_t^\top\right) + \beta_t \boldsymbol{v}_t\boldsymbol{k}_t^\top \\ \text{GDN-2:} \quad & \boldsymbol{S}_t = \boldsymbol{S}_{t-1}\text{Diag}\left(\boldsymbol{\alpha}_t\right)\left(\boldsymbol{I} - \left(\boldsymbol{b}_t \odot \boldsymbol{k}_t\right)\boldsymbol{k}_t^\top\right) + \left(\boldsymbol{w}_t \odot \boldsymbol{v}_t\right)\boldsymbol{k}_t^\top \end{aligned}$$

We define the *integral state* $\boldsymbol{X}_t$ by

$$\boldsymbol{X}_t = \boldsymbol{S}_t\text{Diag}\left(\boldsymbol{\lambda}_t\right) + \boldsymbol{X}_{t-1}\text{Diag}\left(\mathbf{1} - \boldsymbol{\lambda}_t\right), \tag{6}$$

where $\boldsymbol{X}_0 = \mathbf{0}$ and $\boldsymbol{\lambda}_t \in [0,1]^{d_k}$ specifies the integration rate in each key channel. Given a query $\boldsymbol{q}_t$ and a channel-wise readout gate $\boldsymbol{\gamma}_t \in [0,1]^{d_k}$, we compute the output as

$$\boldsymbol{o}_t = \boldsymbol{S}_t\left(\boldsymbol{\gamma}_t \odot \boldsymbol{q}_t\right) + \boldsymbol{X}_t\left(\left(\mathbf{1} - \boldsymbol{\gamma}_t\right) \odot \boldsymbol{q}_t\right). \tag{7}$$

Figure 1 illustrates how the two states complement each other. The delta state is updated with incoming key–value pairs, while the integral state accumulates a weighted history of delta states, preserving contributions that subsequent updates may weaken. Smaller integration rates retain this history for longer, while larger rates make $\boldsymbol{X}_t$ follow $\boldsymbol{S}_t$ more closely. At readout, larger entries of $\boldsymbol{\gamma}_t$ place more weight on the current delta state in the corresponding key channels, favoring current information over accumulated history. These gates allow the model to balance information from different timescales according to the current context. Note that the two states are combined only at readout, leaving the delta-state updates unchanged.

Benefits of the Integral State. We revisit the illustrative example analyzed in Proposition 1 to show how Integral DeltaNet improves retrieval performance. We use the same key–value pairs and query distribution, and retain the Gated DeltaNet update (4) for the delta state. For simplicity, we use scalar gates for integration and readout:

$$\boldsymbol{X}_t = \lambda_t \boldsymbol{S}_t + (1-\lambda_t)\boldsymbol{X}_{t-1}, \qquad \boldsymbol{o}_3 = \left[\gamma_3 \boldsymbol{S}_3 + (1-\gamma_3)\boldsymbol{X}_3\right]\boldsymbol{q}_3, \tag{8}$$

where $\boldsymbol{X}_0 = \mathbf{0}$, $\lambda_t \in (0, 1)$ is the integration rate at step t, and $\gamma_3 \in [0, 1]$ can depend on the query $\boldsymbol{q}_3$. The model selects the answer maximizing $\boldsymbol{v}_y^\top \boldsymbol{o}_3$. Its retrieval risk is $\mathcal{R}_{\text{Integral}}$.

The integral state can preserve the first answer even when the delta state loses it, allowing adaptive readout to recover all three answers.

Proposition 2. *In the illustrative example shown in Figure 2, Integral DeltaNet (8) can achieve zero retrieval risk, i.e.,* $\min_{\{\alpha_t, \beta_t, \lambda_t\},\, \gamma_3(\cdot)} \mathcal{R}_{\text{Integral}} = 0$.

One construction uses $\beta_t = 1$, $(\alpha_1, \alpha_2, \alpha_3) = (1, 1/4, 1)$, and $\lambda_t = 1/8$ for $t = 1, 2, 3$. With these updates, the delta state $\boldsymbol{S}_3$ suppresses interference from the initial No association enough to recover both later Yes answers, but loses the first answer. However, the integral state $\boldsymbol{X}_3$ retains a weighted contribution from earlier delta states, including $\boldsymbol{S}_1$, before the initial association was weakened. This retained history allows it to recover the first No answer. This result also holds in other settings; we chose this simple case for clarity. The full proof is in Appendix A.2.

Parametrization. We compute both gates introduced in (6) and (7) from the normalized hidden state $\boldsymbol{h}_t$ as

$$\boldsymbol{\gamma}_t = \exp\left[-\text{softplus}\left(\boldsymbol{W}_\gamma^\uparrow \boldsymbol{W}_\gamma^\downarrow \boldsymbol{h}_t + \boldsymbol{b}_\gamma\right)\right], \quad \boldsymbol{\lambda}_t = \mathbf{1} - \exp\left[-\text{softplus}\left(\boldsymbol{W}_\lambda^\uparrow \boldsymbol{W}_\lambda^\downarrow \boldsymbol{h}_t + \boldsymbol{b}_\lambda\right)\right].$$

Following the low-rank parametrization of KDA, each projection uses a bottleneck rank small relative to the hidden dimension. We initialize the gates so that $\mathbf{1} - \boldsymbol{\gamma}_t$ and $\boldsymbol{\lambda}_t$ are both small. This initialization places nearly all readout weight on the delta state $\boldsymbol{S}_t$, while $\boldsymbol{X}_t$ gradually accumulates a smoothed history of $\boldsymbol{S}_t$. The gates learn its retention time and readout weight.

3.2 Hardware Efficient Training

Chunkwise form supports hardware-efficient training by parallelizing matrix operations within chunks while propagating recurrent states across chunks (Yang et al., 2024a; Yang & Zhang, 2024). We derive this formulation for Integral DeltaNet. For simplicity, assume $L = NC$ and index the N chunks of size C by $t = 0, \ldots, N-1$. A superscript $r \in \{1, \ldots, C\}$ denotes a position within chunk $[t]$. We write $\square_{[t]}^r := \square_{tC+r}$ for $\square \in \{\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}, \boldsymbol{o}, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{S}, \boldsymbol{X}\}$. For the recurrent states, we define $\square_{[t]} := \square_{[t]}^0 := \square_{tC}$ for $\square \in \{\boldsymbol{S}, \boldsymbol{X}\}$, where $r = 0$ marks the chunk start; hence $\square_{[t+1]} = \square_{[t]}^C$.

The delta state evolves according to a linear RNN. Within each chunk, the state at position r is

$$\boldsymbol{S}_{[t]}^r = \boldsymbol{S}_{[t]} \boldsymbol{P}_{[t]}^r + \boldsymbol{H}_{[t]}^r, \qquad \boldsymbol{P}_{[t]}^0 = \boldsymbol{I}, \quad \boldsymbol{H}_{[t]}^0 = \mathbf{0}, \qquad 0 \le r \le C,$$

where $\boldsymbol{P}_{[t]}^r \in \mathbb{R}^{d_k \times d_k}$ propagates the incoming state and $\boldsymbol{H}_{[t]}^r \in \mathbb{R}^{d_v \times d_k}$ collects the writes within the first r positions. Both operators depend only on the inputs and gates in the current chunk. A line of work (Yang et al., 2024b; 2025; Kimi Team, 2025; Hatamizadeh et al., 2026) has developed efficient algorithms based on WY representations (Bischof & Van Loan, 1987) and UT transforms (Joffrain et al., 2006) to compute state updates and outputs without materializing all prefix matrices.

We express the integral state similarly using within-chunk retention weights for $0 \le r \le C$:

$$\boldsymbol{\omega}_{[t]}^{0 \to r} := \prod_{s=1}^{r} \left(\mathbf{1} - \boldsymbol{\lambda}_{[t]}^s\right), \qquad \boldsymbol{\omega}_{[t]}^{i \to r} := \boldsymbol{\lambda}_{[t]}^i \odot \prod_{s=i+1}^{r} \left(\mathbf{1} - \boldsymbol{\lambda}_{[t]}^s\right), \quad 1 \le i \le r,$$

where products are elementwise and an empty product is $\mathbf{1}$. Define the retention-weighted operators:

$$\bar{\boldsymbol{P}}_{[t]}^r := \sum_{i=1}^{r} \boldsymbol{P}_{[t]}^i \text{Diag}\left(\boldsymbol{\omega}_{[t]}^{i \to r}\right), \qquad \bar{\boldsymbol{H}}_{[t]}^r := \sum_{i=1}^{r} \boldsymbol{H}_{[t]}^i \text{Diag}\left(\boldsymbol{\omega}_{[t]}^{i \to r}\right),$$

with empty sums equal to zero matrices of the corresponding dimensions. Substituting the delta-state representation into the integral-state recurrence gives

$$\boldsymbol{X}_{[t]}^r = \boldsymbol{S}_{[t]} \bar{\boldsymbol{P}}_{[t]}^r + \boldsymbol{X}_{[t]} \text{Diag}\left(\boldsymbol{\omega}_{[t]}^{0 \to r}\right) + \bar{\boldsymbol{H}}_{[t]}^r.$$

Thus, the integral state combines retention-weighted prefix operators with the retained incoming integral state. For the full-chunk operators, we omit the superscript C and write $\square_{[t]} := \square_{[t]}^C$ for $\square \in \{\boldsymbol{P}, \boldsymbol{H}, \bar{\boldsymbol{P}}, \bar{\boldsymbol{H}}\}$. The two states then propagate to the next chunk as

$$\boldsymbol{S}_{[t+1]} = \boldsymbol{S}_{[t]} \boldsymbol{P}_{[t]} + \boldsymbol{H}_{[t]}, \quad \boldsymbol{X}_{[t+1]} = \boldsymbol{S}_{[t]} \bar{\boldsymbol{P}}_{[t]} + \boldsymbol{X}_{[t]} \text{Diag}\left(\boldsymbol{\omega}_{[t]}^{0 \to C}\right) + \bar{\boldsymbol{H}}_{[t]}.$$

The outputs similarly separate into contributions from the two incoming states and the writes within the chunk. For $r = 1, \dots, C$, define the effective queries and the within-chunk output contribution as

$$\widetilde{\boldsymbol{q}}^r_{\mathrm{S},[t]} := \boldsymbol{P}^r_{[t]} \left(\boldsymbol{\gamma}^r_{[t]} \odot \boldsymbol{q}^r_{[t]} \right) + \bar{\boldsymbol{P}}^r_{[t]} \left(\left(\mathbf{1} - \boldsymbol{\gamma}^r_{[t]} \right) \odot \boldsymbol{q}^r_{[t]} \right), \quad \widetilde{\boldsymbol{q}}^r_{\mathrm{X},[t]} := \boldsymbol{\omega}^{0 \to r}_{[t]} \odot \left(\mathbf{1} - \boldsymbol{\gamma}^r_{[t]} \right) \odot \boldsymbol{q}^r_{[t]},$$

$$\boldsymbol{z}^r_{[t]} := \boldsymbol{H}^r_{[t]} \left(\boldsymbol{\gamma}^r_{[t]} \odot \boldsymbol{q}^r_{[t]} \right) + \bar{\boldsymbol{H}}^r_{[t]} \left(\left(\mathbf{1} - \boldsymbol{\gamma}^r_{[t]} \right) \odot \boldsymbol{q}^r_{[t]} \right).$$

We stack $(\widetilde{\boldsymbol{q}}^r_{a,[t]})^\top$, $(\boldsymbol{z}^r_{[t]})^\top$, and $(\boldsymbol{o}^r_{[t]})^\top$ as the r-th rows of $\widetilde{\boldsymbol{Q}}_{a,[t]} \in \mathbb{R}^{C \times d_k}$ and $\boldsymbol{Z}_{[t]}, \boldsymbol{O}_{[t]} \in \mathbb{R}^{C \times d_v}$, respectively, for $a \in \{\mathrm{S}, \mathrm{X}\}$. The chunk output is then

$$\boldsymbol{O}_{[t]} = \widetilde{\boldsymbol{Q}}_{\mathrm{S},[t]} \boldsymbol{S}^\top_{[t]} + \widetilde{\boldsymbol{Q}}_{\mathrm{X},[t]} \boldsymbol{X}^\top_{[t]} + \boldsymbol{Z}_{[t]}.$$

Using this chunkwise form and implementations in FlashLinearAttention (Yang & Zhang, 2024), we implement kernels for Integral DeltaNet. We report throughput comparisons in Section 4.

4 Experiments

We apply Integral DeltaNet to KDA and GDN-2, yielding *Integral KDA* and *Integral GDN-2*, respectively. We compare them with strong recurrent baselines Mamba-2, DeltaNet, GDN, KDA, and GDN-2, and with Transformer++ (Touvron et al., 2023). We pretrain all models from scratch with 340M/1.3B non-embedding parameters on 15B/100B FineWeb-Edu (Penedo et al., 2024) tokens, using Llama-2 tokenization and 4K sequences.

Both scales use AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$), 0.5M-token global batches, cosine learning-rate decay from 4×10^{-4} to 4×10^{-5}, weight decay 0.1, and gradient clipping at 1.0. We use 0.5B/1B tokens of linear warmup at 340M/1.3B. We initialize rank-128 gate projections so that $\lambda_{t,j} \approx 1 - \gamma_{t,j} \approx 1/1024$ for each key channel j.

We evaluate these trained models on language modeling, commonsense reasoning, and real-world retrieval at both scales. For tasks requiring longer contexts, we additionally evaluate the 1.3B models on Needle-in-a-Haystack (Hsieh et al., 2024), LongBench (Bai et al., 2024), and length extrapolation. Finally, we measure training throughput at the 1.3B scale. In the tables, we bold the better result in each baseline/Integral pair and mark the top two distinct scores among recurrent models with $\star$/$\dagger$ at each scale.

Language Modeling and Commonsense Reasoning. We use the LM Evaluation Harness (Gao et al., 2023) for evaluation. We report perplexity (ppl) on WikiText-2 (Merity et al., 2017) and LAMBADA (Paperno et al., 2016) and zero-shot accuracy (acc) on nine tasks (Table 1). We use length-normalized accuracy (acc_n) on selected tasks to reduce answer-length bias. Both Integral variants show overall gains at both scales.

Table 1: Performance on language modeling and zero-shot commonsense reasoning.

Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	OBQA acc_n ↑	SIQA acc ↑	BoolQ acc ↑	Avg. acc ↑
340M parameters / 15B tokens												
Transformer++	26.37	36.20	33.1	65.8	39.3	51.1	59.0	27.6	33.6	38.1	59.5	45.2
Mamba-2	27.95	36.89	30.1	66.4	39.5	50.8	58.2†	27.0	33.8	38.9†	55.3	44.4
DeltaNet	27.52	38.86	30.9	66.1	38.0	50.5	56.2	26.9	31.4	37.7	57.0	43.8
GDN	26.32	31.37	33.4	66.8	40.4	52.6	57.7	27.0	34.2†	38.4	56.7	45.3
KDA	**24.89***	26.16†	36.4†	66.9†	**41.2***	**53.4***	**59.4***	27.3†	33.6	**39.0***	57.9†	46.1†
Integral KDA	24.97†	**24.99***	**36.9***	**68.4***	40.7†	52.9†	58.2†	**28.0***	**34.8***	38.8	**59.5***	**46.5***
GDN-2	**26.76**	30.13	34.1	**66.2**	38.5	49.3	56.0	**26.9**	32.4	38.6	**59.5***	44.6
Integral GDN-2	27.59	**28.82**	**35.8**	65.4	**38.7**	**52.8**	**57.4**	26.2	**33.0**	**38.9**†	57.2	**45.0**
1.3B parameters / 100B tokens												
Transformer++	15.90	11.04	49.4	72.0	58.4	58.6	72.1	37.9	40.2	41.0	62.2	54.7
Mamba-2	16.55	12.22	45.6	72.0	58.2	56.4	71.6†	38.7	41.6*	39.7	63.0*	54.1
DeltaNet	15.73	11.51	47.3	73.2†	58.1	56.7	71.0	40.0†	41.0	40.4	60.7	54.3
GDN	16.08	10.98	48.5	73.2†	58.7	59.4†	71.3	39.3	40.8	41.0	59.9	54.7†
KDA	15.33†	9.63†	50.4	**73.9***	60.2†	57.6	71.2	39.0	**41.6***	40.7	57.2	54.6
Integral KDA	**15.00***	**9.16***	**52.7***	73.2†	**60.7***	**61.1***	**72.3***	**40.6***	41.2	**42.5***	**60.7**	**56.1***
GDN-2	16.34	10.56	49.3	**71.9**	**57.9**	57.6	70.0	**39.3**	39.8	**41.5**†	**60.9**†	54.2
Integral GDN-2	**15.92**	**10.31**	**50.6**†	71.7	57.8	**59.2**	**71.5**	38.1	**41.4**†	41.1	59.6	**54.6**

Real-World Retrieval. We evaluate document retrieval tasks (Arora et al., 2024b) (Table 2). SWDE (Hao et al., 2011) and FDA (Arora et al., 2023) test extraction from webpages and medical reports respectively. SQuAD (Rajpurkar et al., 2018) and TriviaQA (Joshi et al., 2017) test question answering alongside NQ (Kwiatkowski et al., 2019) and DROP (Dua et al., 2019). Both Integral variants improve average retrieval accuracy over their baselines at both scales.

Table 2: Accuracy (%, ↑) on six real-world retrieval tasks.

	340M parameters / 15B tokens							1.3B parameters / 100B tokens						
Model	SWDE	SQuAD	FDA	TriviaQA	NQ	DROP	**Avg.**	SWDE	SQuAD	FDA	TriviaQA	NQ	DROP	**Avg.**
Transformer++	31.8	34.2	48.2	48.4	17.3	19.4	33.2	48.2	45.3	65.7	62.8	26.7	25.1	45.6
Mamba-2	6.6	24.0	3.7	42.9	10.5	14.9	17.1	20.7	36.5	13.6	59.4	20.1	21.1†	28.6
DeltaNet	7.0	23.7	7.7	45.7	12.4	16.1	18.8	29.6	36.7	27.4	60.3	21.9	21.0	32.8
GDN	11.2	27.6	7.2	46.3	14.2	17.5	20.7	34.0†	39.6†	27.3	62.3	24.1†	20.7	34.7
KDA	10.2	28.6†	8.1	47.8†	14.4†	**17.7**†	21.1	32.7	**41.1***	30.2†	**62.9***	24.1†	**21.6***	35.4†
Integral KDA	**18.1***	**30.6***	**9.6***	**48.9***	**15.0***	17.3	**23.3***	**40.4***	39.5	**35.1***	62.4	**25.3***	20.9	**37.3***
GDN-2	10.2	26.9	4.5	**45.9**	13.3	**18.3***	19.8	24.5	37.5	27.9	60.9	23.3	19.9	32.3
Integral GDN-2	**15.7**†	**27.4**	**8.9**†	45.6	**14.0**	16.2	**21.3**†	**31.7**	**39.4**	**29.3**	**62.6**†	**24.0**	**20.2**	**34.5**

Needle-in-a-Haystack (NIAH). We evaluate long-context retrieval using four synthetic Needle-in-a-Haystack tasks from RULER (Hsieh et al., 2024). We evaluate the 1.3B models on the same 500 examples per task and context length (Table 3). S-NIAH-1/2 retrieve a number from repetitive/natural text, respectively; S-NIAH-3 retrieves a UUID from natural text, and MK-NIAH-1 queries one of multiple key–value pairs. Both Integral variants improve overall performance: Integral KDA matches or outperforms KDA in nearly all settings, with substantial gains on the challenging S-NIAH-3 at 4K and MK-NIAH-1 tasks. Integral GDN-2 underperforms GDN-2 in some settings but improves S-NIAH-3 at 4K and achieves particularly large gains on MK-NIAH-1 at every tested length.

Table 3: Accuracy (%, ↑) on Needle-in-a-Haystack tasks.

	S-NIAH-1				S-NIAH-2				S-NIAH-3			MK-NIAH-1		
Model	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K	1K	2K	4K
Transformer++	100.0	100.0	98.4	0.0	100.0	100.0	86.2	0.0	100.0	99.4	69.8	98.4	94.4	77.0
Mamba-2	100.0*	99.8†	69.0	21.4	100.0*	98.6	65.8	23.8	63.4	46.6	13.2	29.2	21.2	18.4
DeltaNet	100.0*	100.0*	100.0*	100.0*	100.0*	97.4	80.6	47.6†	95.2*	65.2	21.0	52.8	33.4	30.4
GDN	100.0*	99.2	91.6	63.6	88.8	99.6*	89.8†	28.0	90.8	63.2	20.0	48.8	39.0	25.0
KDA	100.0*	100.0*	99.6	78.4	100.0*	79.8	70.0	32.8	90.4	**68.8**	24.8	47.2	37.0	20.8
Integral KDA	100.0*	100.0*	**100.0***	**92.4**	100.0*	**99.2**†	**99.6***	**49.2***	**91.8**	58.6	**41.6***	**54.0**†	**50.4**†	**35.6**†
GDN-2	100.0*	**100.0***	**99.8**†	**97.8**†	99.6†	93.8	**74.2**	**30.8**	**94.0**†	**80.0***	30.0	45.2	32.8	23.8
Integral GDN-2	100.0*	99.8†	98.8	85.8	**100.0***	**94.6**	65.2	29.4	93.4	77.4†	**34.6**†	**76.4***	**67.2***	**39.8***

LongBench. We assess long-context understanding using the 1.3B models on English LongBench tasks (Bai et al., 2024) covering question answering, summarization, few-shot learning, and code completion. We use the LM Evaluation Harness (Gao et al., 2023) with greedy decoding, a 16K-token context budget, and default prompts and truncation, reporting the mean of official task scores. Both Integral variants achieve strong overall performance on LongBench (Table 4).

Table 4: Scores (↑) on LongBench tasks.

	Single-Doc QA			Multi-Doc QA			Summarization			Few-shot			Code		
Model	NQA	QQA	MFQ	HQA	2WM	Mus	GvR	QMS	MNs	TRC	TQA	SSM	LCC	RBP	**Avg.**
Transformer++	0.5	2.3	5.3	1.1	2.2	0.4	5.3	5.8	7.5	11.5	6.6	3.9	17.4	8.4	5.6
Mamba-2	2.0	4.3	12.5	4.1	8.1	3.1†	6.6	16.6	8.7	31.0	22.8	20.5	11.2	17.8	12.1
DeltaNet	2.2	3.9	13.6	5.3	8.1	2.4	4.9	17.5†	10.4	37.0	19.8	27.6†	15.7	18.3	13.3
GDN	2.1	5.1†	13.7†	5.1	8.4	3.3*	6.8	18.3*	12.3	26.0	20.7	23.3	17.7†	19.4	13.0
KDA	2.0	4.3	**13.0**	5.1	**8.4**	**3.1**†	**10.0**†	16.2	9.6	29.5	20.5	27.5	**21.9***	**24.0***	13.9
Integral KDA	**2.6***	**4.4**	9.7	5.1	8.1	2.9	6.6	**16.4**	**11.8**	**52.5***	**27.3***	**28.1***	16.3	18.0	**15.0***
GDN-2	2.2	5.2*	13.1	5.5†	**9.2***	**3.3***	9.8	**16.7**	12.4†	**42.0**†	**25.4**†	24.0	15.4	**19.7**†	14.6
Integral GDN-2	**2.5**†	5.2*	**13.9***	**5.9***	9.1†	2.8	**12.5***	16.1	**13.0***	40.5	24.8	**27.4**	**15.9**	17.1	**14.8**†

Length Extrapolation. We evaluate how well Integral DeltaNet generalizes beyond the 4K training length compared with other recurrent models. We measure perplexity of the 1.3B models up to 20K on six datasets: GovReport (Huang et al., 2021), QMSum (Zhong et al., 2021), NarrativeQA (Kočiský et al., 2018), Qasper (Dasigi et al., 2021), CodeParrot (von Werra, 2021), and PG19 (Rae et al., 2020). Both Integral variants outperform their respective baselines across all six datasets at every evaluated length beyond the 4K training context. Together, they achieve the lowest perplexity among all compared models at every evaluated length on every dataset: Integral KDA leads on GovReport, QMSum, Qasper, and CodeParrot, while Integral GDN-2 leads on NarrativeQA and PG19 (Figure 3).

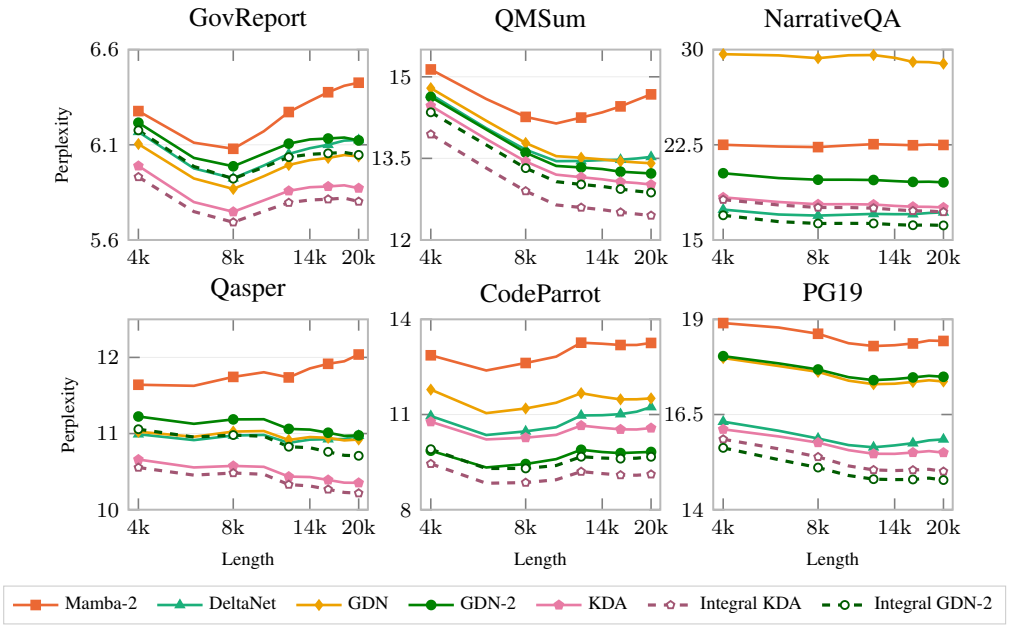


Figure 3: Length extrapolation of 1.3B models trained at 4K (perplexity, ↓).

Training Throughput. We measure end-to-end training throughput of recurrent models and Transformer++ at the 1.3B scale on a single NVIDIA B300 with 32,768 tokens per step (Figure 4). Integral KDA and Integral GDN-2 remain close to their respective backbones, retaining 87–94% of their throughput across the sweep. At 32K × 1, they approach Transformer++ with Blackwell-optimized FlashAttention-4 (Zadouri et al., 2026) (47.3–47.5 vs. 50.3 Kt/s). With the general-purpose FlashAttention-2 (Dao, 2024) implementation, comparable throughput is reached already at 8K × 4 (52.8 vs. 52.9 Kt/s), and both Integral variants are faster at longer contexts. Their peak activation memory at 32K is roughly twice that of the backbones.

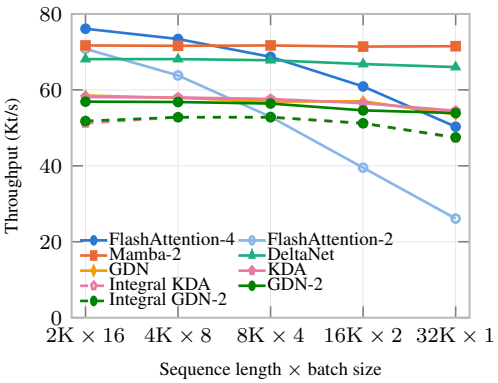


Figure 4: Training throughput on a B300 GPU.

5 Related Work

Linear RNNs. Linear RNNs, including state space models and linear attention, combine parallel training and efficient recurrent inference (Martin & Cundy, 2018; Katharopoulos et al., 2020; Dao & Gu, 2024) with strong sequence modeling performance (Gu et al., 2022; Orvieto et al., 2023; Gu & Dao, 2024), making them promising alternatives to Transformers. Representative models include S4 (Gu et al., 2022), the Linear Recurrent Unit (LRU) (Orvieto et al., 2023), RWKV (Peng et al.,

2023), Mamba (Gu & Dao, 2024), Mamba-2 (Dao & Gu, 2024), and Mamba-3 (Lahoti et al., 2026). They also show promise on several tasks, including in-context learning (Park et al., 2024; Grazzi et al., 2024; Li et al., 2025; Jiang et al., 2025; Oh et al., 2025; Bondaschi et al., 2026; Li et al., 2026). However, constrained state transitions and fixed-size memory can hinder state tracking, associative recall, and copying (Merrill et al., 2024; Grazzi et al., 2025; Arora et al., 2024a; Jelassi et al., 2024).

Linear Attention. Linear attention (Katharopoulos et al., 2020) additively accumulates key–value associations, which can interfere with one another (Schlag et al., 2021). To address this interference, RetNet (Sun et al., 2023) uses constant scalar decay and Mamba-2 (Dao & Gu, 2024) and Gated Linear Attention (GLA) (Yang et al., 2024a) extend this mechanism to time-varying scalar and channel-wise decay, respectively. DeltaNet (Schlag et al., 2021; Yang et al., 2024b) addresses interference through delta-rule updates for targeted overwriting of key–value memory, with later variants adding gating (Yang et al., 2025; Kimi Team, 2025; Hatamizadeh et al., 2026). The DeltaNet state update corresponds to one stochastic gradient descent (SGD) step on squared prediction error for values given keys. This perspective motivates variants of DeltaNet that perform multiple updates per token (Siems et al., 2025), apply diagonal preconditioning (Tumma et al., 2026), or incorporate momentum (Huang et al., 2026). Park et al. (2026) incorporate queries into state updates by combining key- and query-based prediction errors.

Neural Memory. From an online optimization view, DeltaNet can be interpreted as learning key–value associations by updating a linear model to minimize squared prediction error. Sun et al. (2025) extend this view by treating the parameters of a neural network, such as an MLP, as the recurrent state, with each state update corresponding to a gradient descent step on these parameters. For these neural memory updates, Behrouz et al. (2025) add momentum and Behrouz et al. (2026a) use Muon (Jordan et al., 2024). These memories offer greater expressiveness and can improve long-context modeling with memory size independent of sequence length. However, nonlinear updates can increase computation and memory I/O while limiting parallelization relative to linear attention (Sun et al., 2025; Behrouz et al., 2026b).

Complementary Memories. Similar to our work, other approaches have explored combining complementary memories. Behrouz et al. (2025) combine local attention with neural memory and Liu et al. (2026) add a Gated DeltaNet-2 branch to a pretrained Transformer to retain information beyond its bounded attention window. Du et al. (2026) route tokens to independent recurrent states and Bayat et al. (2026) progressively activate memory as context grows. Integral DeltaNet instead preserves information weakened by later updates through a weighted history of delta states.

Iterate Averaging. We form the integral state as an exponential moving average of delta states across tokens, starting from a zero initial state. Averaging iterates across steps is also used in Stochastic Weight Averaging (Izmailov et al., 2018) and Lookahead (Zhang et al., 2019). Other examples include schedule-free methods and their variants (Defazio et al., 2024; Defazio, 2026; Kim et al., 2026; Apte et al., 2026): they combine a fast iterate with its running average and compute gradients at their interpolation, avoiding a prescribed learning-rate decay schedule. Song et al. (2025) show that implicit weight averaging helps schedule-free methods track low-loss trajectories. Despite this similarity, Integral DeltaNet differs from applying a schedule-free optimizer: we use the integral state for adaptive readout, not to update the delta state.

6 Conclusion

In this work, we introduced Integral DeltaNet to address the trade-off between mitigating memory collisions and retaining useful information. To this end, the model pairs a delta state that adapts to incoming key–value associations with an integral state that preserves a weighted history of delta states. Adaptive readout combines these complementary states to access information weakened by later updates, while our chunkwise parallel formulation enables hardware-efficient training. Experiments with Integral KDA and Integral GDN-2 show overall gains in language modeling, commonsense reasoning, retrieval, long-context understanding, and length extrapolation, with training throughput close to that of KDA and GDN-2. These results highlight the importance of integrated memory and adaptive readout. A possible direction for future work is to use the integral state to guide delta-state updates, similar to how schedule-free methods (Defazio et al., 2024) use averaged iterates.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods in code. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, translate text, clean or reformat datasets, or interpret results. Qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for English grammar, clarity, and readability. We have reviewed all AI-assisted work: the authors manually tested the AI-assisted code and reviewed the manuscript sentence by sentence. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Anuj Apte, Pranav Deshpande, Niraj Kumar, Shouvanik Chakrabarti, and Junhyung Lyle Kim. Anytime training with schedule-free spectral optimization. *arXiv preprint arXiv:2605.23061*, 2026. URL https://arxiv.org/abs/2605.23061.

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. *arXiv preprint arXiv:2304.09433*, 2023. URL https://arxiv.org/abs/2304.09433.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Re. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=LY3ukUANko.

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. *arXiv preprint arXiv:2407.05483*, 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, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Reza Bayat, Ali Behrouz, Vahab Mirrokni, and Aaron Courville. Proteus: Incremental memory activation for long-context sequence modeling. *arXiv preprint arXiv:2608.16844*, 2026.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113506–113543, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4ca07aa108036f80cbb5b82285fd4b1-Abstract-Conference.html.

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. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=HV1R5bMSpa.

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. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=gZyEJ2kMow.

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. URL https://epubs.siam.org/doi/abs/10.1137/0908009.

Marco Bondaschi, Nived Rajaraman, Xiuying Wei, Razvan Pascanu, Caglar Gulcehre, Michael Gastpar, and Ashok Vardhan Makkuva. From Markov to Laplace: How Mamba in-context learns Markov chains. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8fef8a350b54e18758ea546bc984ba28-Abstract-Conference.html.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/98ed250b203d1ac6b24bbcf263e3d4a7-Abstract-Conference.html.

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, 2024. 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021. URL https://aclanthology.org/2021.naacl-main.365/.

Aaron Defazio. ScheduleFree+: Scaling learning-rate-free & schedule-free learning to large language models. *arXiv preprint arXiv:2605.19095*, 2026. URL https://arxiv.org/abs/2605.19095.

Aaron Defazio, Xingyu (Alice) Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/136b9a13861308c8948cd308ccd02658-Abstract-Conference.html.

Jusen Du, Weigao Sun, Disen Lan, Jiaxi Hu, Tao Zhang, and Yu Cheng. MoM: Linear sequence modeling with mixture-of-memories. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3PdOq8Rgue.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Riccardo Grazzi, Julien Niklas Siems, Simon Schrodi, Thomas Brox, and Frank Hutter. Is Mamba capable of in-context learning? In *Proceedings of the Third International Conference on Automated Machine Learning*, volume 256 of *Proceedings of Machine Learning Research*, pp. 1/1–26. PMLR, 2024. URL https://proceedings.mlr.press/v256/grazzi24a.html.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K.H. Franke, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UvTo3tVBk2.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: A unified solution for structured web data extraction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/StructedDataExtraction_SIGIR2011.pdf.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026. URL https://arxiv.org/abs/2605.22791.

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? In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *Proceedings of the 39th International Conference on Machine Learning*, 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, 2021. URL https://aclanthology.org/2021.naacl-main.112/.

Yulong Huang, Xiang Liu, Hongxiang Huang, Xiaopeng Lin, Zunchang Liu, Xiaowen Chu, Zeke Xie, and Bojun Cheng. MDN: Parallelizing stepwise momentum for delta linear attention. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Sh3rKEAxlW.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1803.05407.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21502–21521. PMLR, 2024. URL https://proceedings.mlr.press/v235/jelassi24a.html.

Jiarui Jiang, Wei Huang, Miao Zhang, Taiji Suzuki, and Liqiang Nie. Trained Mamba emulates online gradient descent in in-context linear regression. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0f9e0309d8a947ca44463a9b7e8b6a3f-Abstract-Conference.html.

Thierry Joffrain, Tze Meng Low, Enrique S Quintana-Ortí, Robert van de Geijn, and Field G. Van Zee. Accumulating Householder transformations, revisited. *ACM Transactions on Mathematical Software (TOMS)*, 32(2):169–179, 2006. URL https://www.cs.utexas.edu/~flame/books/ACMTOMS.pdf.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. 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*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Jueun Kim, Baekrok Shin, Jihun Yun, Beomhan Baek, Minhak Song, and Chulhee Yun. AMUSE: Anytime Muon with stable gradient evaluation. *arXiv preprint arXiv:2605.22432*, 2026. URL https://arxiv.org/abs/2605.22432.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025. URL https://arxiv.org/abs/2510.26692.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

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. URL https://aclanthology.org/Q18-1023/.

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. URL https://aclanthology.org/Q19-1026/.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8abd2043b71a074278d5f687947bff9c-Abstract-Conference.html.

Hongkang Li, Songtao Lu, Xiaodong Cui, Pin-Yu Chen, and Meng Wang. How can Mamba learn in context with outliers and generalize provably? In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.00399.

Yingcong Li, Davoud Ataee Tarzanagh, Ankit Singh Rawat, Maryam Fazel, and Samet Oymak. Gating is weighting: Understanding gated linear attention through in-context learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=9AFIz0YzD7.

Zhichen Liu, Ruihan Sun, Hengjie Yang, Zipeng Wu, Zhaohan Chen, Xiaofan Zhang, and Yang Xu. LiveMem: Maintaining memory state continuity in long-running LLM inference. *arXiv preprint arXiv:2608.02515*, 2026. URL https://arxiv.org/abs/2608.02515.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyUNwulC-.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.07843.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

Junsoo Oh, Wei Huang, and Taiji Suzuki. Mamba can learn low-dimensional targets in-context via test-time feature learning. *arXiv preprint arXiv:2510.12026*, 2025. URL https://arxiv.org/abs/2510.12026.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, 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 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016. URL https://aclanthology.org/P16-1144/.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Can Mamba learn how to learn? A comparative study on in-context learning tasks. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39793–39812. PMLR, 2024. URL https://proceedings.mlr.press/v235/park24j.html.

Sumin Park, Seojin Kim, and Noseong Park. Q-Delta: Beyond key–value associative state evolution. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=O6tuJZIv0u.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. Random feature attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QtTKTdVrFBB.

Qwen Team. Qwen3-Next: Towards ultimate training & inference efficiency. Qwen Research Blog, 2025. URL https://qwen.ai/blog?id=4074cca80393150c248e508aa62983f9cb7d27cd.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

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

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/e1ea2520fbf9cd1600b287dde67e0a3c-Abstract-Conference.html.

Minhak Song, Beomhan Baek, Kwangjun Ahn, and Chulhee Yun. Through the river: Understanding the benefit of schedule-free methods for language model training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/b94ab4933fea38629a1308fb78cce2cc-Abstract-Conference.html.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

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. URL https://arxiv.org/abs/2307.08621.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

Neehal Tumma, Noel Loo, and Daniela Rus. Preconditioned DeltaNet: Curvature-aware sequence modeling for linear recurrences. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=UC6YiTOeKb.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Leandro von Werra. Training CodeParrot from scratch. Hugging Face Blog, 2021. URL https://huggingface.co/blog/codeparrot.

Songlin Yang and Yu Zhang. FLA: A Triton-based library for hardware-efficient implementations of linear attention mechanism, January 2024. URL https://github.com/fla-org/flash-linear-attention.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 115491–115522, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d13a3eae72366e61dfdc7eea82eeb685-Abstract-Conference.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/4904fad153f6434a7bcf04465d4be2cc-Abstract-Conference.html.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Vijay Thakkar, and Tri Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. In *Proceedings of Machine Learning and Systems*, volume 8, pp. 912–926, 2026. URL https://proceedings.mlsys.org/paper_files/paper/2026/hash/ae8b0b5838ba510daff1198474e7b984-Abstract-Conference.html.

Michael R. Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1907.08610.

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, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472/.

A Proofs for the Illustrative Example

A.1 Proof of Proposition 1

Note that $\boldsymbol{k}_1^\top \boldsymbol{k}_2 = \boldsymbol{k}_1^\top \boldsymbol{k}_3 = 1/2$ and $\boldsymbol{k}_2^\top \boldsymbol{k}_3 = -1/2$. For each final state $\boldsymbol{S}_3$, define the Yes–No score differences

$$m_i := (\boldsymbol{v}_\mathrm{Y} - \boldsymbol{v}_\mathrm{N})^\top \boldsymbol{S}_3 \boldsymbol{k}_i, \qquad i = 1, 2, 3.$$

A positive score difference predicts Yes, and a negative one predicts No. For linear attention, summing the three writes gives

$$\boldsymbol{S}_3 = \boldsymbol{v}_\mathrm{N} \boldsymbol{k}_1^\top + \boldsymbol{v}_\mathrm{Y} (\boldsymbol{k}_2 + \boldsymbol{k}_3)^\top = (\boldsymbol{v}_\mathrm{N} + \boldsymbol{v}_\mathrm{Y}) \boldsymbol{k}_1^\top.$$

The answer-vector norms and orthogonality give $(m_1, m_2, m_3) = (-1, -1/2, -1/2)$. All three predictions are No, with risk $\mathcal{R}_{\text{Linear Attention}} = 2/3$.

For Mamba-2, applying the update $\boldsymbol{S}_t = \alpha_t \boldsymbol{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\top$ three times gives

$$\boldsymbol{S}_3 = \alpha_2 \alpha_3 \boldsymbol{v}_\mathrm{N} \boldsymbol{k}_1^\top + \alpha_3 \boldsymbol{v}_\mathrm{Y} \boldsymbol{k}_2^\top + \boldsymbol{v}_\mathrm{Y} \boldsymbol{k}_3^\top.$$

With $(\alpha_1, \alpha_2, \alpha_3) = (1, 1/4, 1)$, the final state is $\boldsymbol{S}_3 = (\frac{1}{4}\boldsymbol{v}_\mathrm{N} + \boldsymbol{v}_\mathrm{Y}) \boldsymbol{k}_1^\top$. The corresponding score differences are

$$(m_1, m_2, m_3) = \left(\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{4}\right).$$

All three predictions are Yes. Only the first query is incorrect, and $\mathcal{R}_{\text{Mamba-2}} = 1/3$.

For DeltaNet, $\alpha_t = 1$ in (4). With $(\beta_1, \beta_2, \beta_3) = (1/4, 1, 1)$, the first two states are

$$\boldsymbol{S}_1 = \tfrac{1}{4} \boldsymbol{v}_\mathrm{N} \boldsymbol{k}_1^\top, \qquad \boldsymbol{S}_2 = \tfrac{1}{4} \boldsymbol{v}_\mathrm{N} (\boldsymbol{k}_1 - \tfrac{1}{2}\boldsymbol{k}_2)^\top + \boldsymbol{v}_\mathrm{Y} \boldsymbol{k}_2^\top.$$

These states satisfy $\boldsymbol{S}_2 \boldsymbol{k}_2 = \boldsymbol{v}_\mathrm{Y}$ and $\boldsymbol{S}_2 \boldsymbol{k}_3 = \frac{3}{16}\boldsymbol{v}_\mathrm{N} - \frac{1}{2}\boldsymbol{v}_\mathrm{Y}$. After the third write,

$$\begin{aligned} \boldsymbol{S}_3 \boldsymbol{k}_2 &= \boldsymbol{S}_2 (\boldsymbol{k}_2 + \tfrac{1}{2}\boldsymbol{k}_3) - \tfrac{1}{2}\boldsymbol{v}_\mathrm{Y} = \tfrac{3}{32}\boldsymbol{v}_\mathrm{N} + \tfrac{1}{4}\boldsymbol{v}_\mathrm{Y}, \\ \boldsymbol{S}_3 \boldsymbol{k}_3 &= \boldsymbol{v}_\mathrm{Y}, \qquad \boldsymbol{S}_3 \boldsymbol{k}_1 = \tfrac{3}{32}\boldsymbol{v}_\mathrm{N} + \tfrac{5}{4}\boldsymbol{v}_\mathrm{Y}. \end{aligned}$$

The final score differences are

$$(m_1, m_2, m_3) = \left(\tfrac{17}{16}, \tfrac{1}{16}, 1\right).$$

Only the first query is incorrect, and $\mathcal{R}_{\text{DeltaNet}} = 1/3$.

Gated DeltaNet combines Mamba-2's decay with the delta rule and includes DeltaNet as the case $\alpha_t = 1$. It therefore also attains risk $1/3$ with the DeltaNet gates above.

To show that Mamba-2, DeltaNet, and Gated DeltaNet cannot achieve risk below $1/3$, consider any final state $\boldsymbol{S}_3$. Since $\boldsymbol{k}_1 = \boldsymbol{k}_2 + \boldsymbol{k}_3$, the score differences satisfy $m_1 = m_2 + m_3$. Correctly answering the second and third queries requires $m_2, m_3 > 0$. This forces $m_1 > 0$, giving an incorrect Yes answer to the first query. Thus at least one of the three equally likely queries must be incorrect. Together with the constructions above, this proves that all three models have minimum risk $1/3$. □

A.2 Proof of Proposition 2

Consider Gated DeltaNet with $\beta_t = 1$ and $(\alpha_1, \alpha_2, \alpha_3) = (1, 1/4, 1)$. The three states are

$$\begin{aligned} \boldsymbol{S}_1 &= \boldsymbol{v}_\mathrm{N} \boldsymbol{k}_1^\top, \\ \boldsymbol{S}_2 &= \tfrac{1}{4} \boldsymbol{v}_\mathrm{N} (\boldsymbol{k}_1 - \tfrac{1}{2}\boldsymbol{k}_2)^\top + \boldsymbol{v}_\mathrm{Y} \boldsymbol{k}_2^\top, \\ \boldsymbol{S}_3 &= \tfrac{1}{4} \boldsymbol{v}_\mathrm{N} (\boldsymbol{k}_1 - \tfrac{1}{2}\boldsymbol{k}_2 - \tfrac{3}{4}\boldsymbol{k}_3)^\top + \boldsymbol{v}_\mathrm{Y} (\boldsymbol{k}_2 + \tfrac{3}{2}\boldsymbol{k}_3)^\top. \end{aligned}$$

The Yes–No score differences for the first query at the three write times are

$$(\boldsymbol{v}_\mathrm{Y} - \boldsymbol{v}_\mathrm{N})^\top \boldsymbol{S}_1 \boldsymbol{k}_1 = -2, \qquad (\boldsymbol{v}_\mathrm{Y} - \boldsymbol{v}_\mathrm{N})^\top \boldsymbol{S}_2 \boldsymbol{k}_1 = \tfrac{1}{8}, \qquad (\boldsymbol{v}_\mathrm{Y} - \boldsymbol{v}_\mathrm{N})^\top \boldsymbol{S}_3 \boldsymbol{k}_1 = \tfrac{17}{16}.$$

With $\lambda_t = 1/8$ for $t = 1, 2, 3$, applying $\boldsymbol{X}_t = \lambda_t \boldsymbol{S}_t + (1 - \lambda_t) \boldsymbol{X}_{t-1}$ three times from $\boldsymbol{X}_0 = \boldsymbol{0}$ gives

$$\begin{aligned} \boldsymbol{X}_3 &= \lambda_1 (1 - \lambda_2)(1 - \lambda_3) \boldsymbol{S}_1 + \lambda_2 (1 - \lambda_3) \boldsymbol{S}_2 + \lambda_3 \boldsymbol{S}_3 \\ &= \tfrac{1}{8} \left(\tfrac{49}{64} \boldsymbol{S}_1 + \tfrac{7}{8} \boldsymbol{S}_2 + \boldsymbol{S}_3 \right). \end{aligned}$$

The score difference for the first query is

$$(\boldsymbol{v}_{\mathrm{Y}} - \boldsymbol{v}_{\mathrm{N}})^{\top} \boldsymbol{X}_3 \boldsymbol{k}_1 = \tfrac{1}{8}\left(-\tfrac{49}{64} \cdot 2 + \tfrac{7}{8} \cdot \tfrac{1}{8} + \tfrac{17}{16}\right) = -\tfrac{23}{512} < 0.$$

Read $\boldsymbol{X}_3$ for $\boldsymbol{q}_3 = \boldsymbol{k}_1$ and $\boldsymbol{S}_3$ for $\boldsymbol{q}_3 \in \{\boldsymbol{k}_2, \boldsymbol{k}_3\}$. For the latter two queries,

$$(\boldsymbol{v}_{\mathrm{Y}} - \boldsymbol{v}_{\mathrm{N}})^{\top} \boldsymbol{S}_3 \boldsymbol{k}_2 = \tfrac{1}{16} > 0, \qquad (\boldsymbol{v}_{\mathrm{Y}} - \boldsymbol{v}_{\mathrm{N}})^{\top} \boldsymbol{S}_3 \boldsymbol{k}_3 = 1 > 0.$$

The Yes–No score difference is negative for the first query and positive for the other two. These signs give the correct answers No, Yes, and Yes, respectively, proving $\mathcal{R}_{\mathrm{Integral}} = 0$. □