DIFFERENTIAL ATTENTION FOR STATE SPACE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The Transformer architecture has emerged as the dominant paradigm for sequence modeling; however, it suffers from quadratic complexity in sequence length. In parallel, state space models (SSMs), such as Mamba, have been proposed as efficient alternatives, offering sub-quadratic complexity while maintaining competitive performance. The Differential Transformer demonstrated that transformers tend toward attention over-allocation, which can impair retrieval and long-context reasoning. They address over-allocation by subtracting attention maps, thereby improving signal selectivity and mitigating noise. Existing attempts to transfer this idea to SSMs rely on parallel reduced Mamba blocks that incur limited scalability, and significant computational overhead. We introduce a generalized formulation of differential computation for state space models derived directly from structured state space duality (SSD). Rather than constructing parallel SSM branches, our formulation shows that differential computation naturally emerges as an operation on state-space readout. This perspective enables both an exact single-pass formulation and an efficient single differential circuit implementation without modifying the underlying SSM kernels.

We instantiate the proposed framework in both Mamba-2 and Mamba-3, yielding Diff-Mamba-2 and Diff-Mamba-3. Across language modeling benchmarks, our models consistently outperform their corresponding baselines by up to 1.5 average points, while preserving the computational efficiency of modern SSM architectures. We further demonstrate improved retrieval on real-world long-context benchmarks, and sharper token-level interaction patterns through interpretability analysis. These results establish differential computation as a principled and effective design paradigm for state space models. Our code is publicly available.

1 INTRODUCTION

Modern deep sequence models led by the Transformers architecture (Vaswani et al., 2017) drive the age of Large Language Models (LLMs), demonstrating enormous success in Natural Language Processing (NLP) downstream tasks and effective scaling capability (Achiam et al., 2023; Yang et al., 2025a; Touvron et al., 2023; Grattafiori et al., 2024). However, these models have two major challenges: efficiency, constrained by the quadratic time complexity in sequence length of the Transformer architecture, and robustness, which manifests in hallucinations, weakened retrieval, and sub-optimal performance (Liu et al., 2023). Differential Transformer (Ye et al., 2024) introduces an elegant approach to enhancing robustness by subtracting two attention matrices, thereby mitigating the over-allocation of attention to irrelevant contextual information. This modification promotes sparser and more informative attention distributions, leading to consistent performance improvements across diverse tasks. The architecture has been rapidly adopted and extended, with several subsequent variants and analyses proposed in recent works (Cang et al., 2025b; Kong et al., 2026; Takahashi et al., 2026).

In parallel, state space models (SSMs), such as Mamba (Gu & Dao, 2024; Dao & Gu, 2024; Lahoti et al., 2026), have emerged as efficient alternatives to Transformer-based architectures, offering sub-quadratic complexity while maintaining competitive performance. As a result, they have seen increasing adoption in contemporary LLMs (NVIDIA, 2025; Lenz et al., 2025; Zuo et al., 2024; Glorioso et al., 2024; Dong et al., 2025). Recent work has attempted to extend this principle to SSMs through Differential Mamba (Schneider et al., 2025) and has already inspired subsequent adaptations and extensions (Bae et al., 2025; Cavazos et al., 2026). However, its construction relies on

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.

subtracting the outputs of two independent reduced Mamba-2 blocks, effectively treating differential computation as an architectural duplication rather than an intrinsic property of SSMs. This design increases computational cost, scales less efficiently, and provides limited insight into how differential mechanisms arise from state-space dynamics. Consequently, it remains unclear whether differential computation admits a principled formulation within the SSM framework itself.

In this work, we answer this question by deriving a generalized formulation of differential computation directly from Structured State Space Duality (SSD), which established a unified mathematical framework connecting attention and SSMs (Dao & Gu, 2024). Rather than introducing parallel state-space branches, we show that differential computation can be expressed as an intrinsic operation on state-space representations. This formulation naturally leads to an exact single-pass realization and motivates lightweight differential architectures that retain the computational advantages of modern SSMs. Building on this framework, we instantiate differential computation in both Mamba-2 and Mamba-3, resulting in Diff-Mamba-2 and Diff-Mamba-3. We further introduce efficient architectural variants, including a Single Differential Circuit formulation that preserves the original state dimension and avoids duplicated state-space computations. Unlike previous methods, our approach requires only minimal architectural modifications while maintaining compatibility with existing optimized SSM kernels.

Extensive experiments demonstrate that the proposed formulation consistently improves language modeling, retrieval, and downstream performance across multiple model scales. Across downstream language modeling benchmarks, Diff-Mamba-2 achieves improvements of up to 1.1 average points over Mamba-2, while Diff-Mamba-3 improves upon Mamba-3 by up to 1.5 average points. The proposed models also exhibit stronger retrieval performance on real-world long-context benchmarks while preserving the efficiency characteristics of their underlying architectures. Finally, interpretability analyses reveal sharper token-to-token interaction patterns, providing empirical evidence that differential computation promotes more selective information processing within SSMs.

Our main contributions are summarized as follows:

- We derive the first principled formulation of differential computation for SSMs through SSD, establishing differential computation as an intrinsic operation within the SSM framework rather than an architectural heuristic.
- We show that differential computation can be computed with an exact single-pass and an efficient Single Differential Circuit implementation that preserves the computational efficiency of modern SSMs without changing their optimized kernels.
- The proposed framework was instantiated in both Mamba-2 and Mamba-3, obtaining Diff-Mamba-2 and Diff-Mamba-3, which consistently outperform their corresponding baselines across language modeling downstream tasks, and retrieval benchmarks while maintaining comparable efficiency.

2 Background

2.1 State-Space Layers

State-space layers were first introduced by Gu et al (Gu et al., 2021b) and later substantially improved by the Structured State Space (S4) model (Gu et al., 2021a). The S4 framework discretizes the continuous system with step size Δ and imposes special structure on the state matrix $\mathbf{A}$ (e.g., diagonal or diagonal-plus-low-rank), enabling efficient computation. The resulting discrete recurrence is given by

$$h_t = \bar{\mathbf{A}}\, h_{t-1} + \bar{\mathbf{B}}\, x_t, \qquad y_t = \mathbf{C}_t^\top h_t + D\, x_t. \tag{1}$$

These models implement linear recurrent update rules derived from time-invariant state-space formulations and can be evaluated either as recurrences or as convolutions, enabling efficient parallel computation with sub-quadratic complexity. As a result, structured state-space models have demonstrated strong performance across a wide range of domains, including NLP (Fu et al., 2022; Mehta et al., 2022), audio generation (Goel et al., 2022), image modeling (Baron et al., 2023; Nguyen et al., 2022), reinforcement learning (David et al., 2022; Lu et al., 2023), and speech recognition (Saon et al., 2023).

2.2 MAMBA AND SELECTIVE STATE SPACE

Mamba (Gu & Dao, 2024) introduced selective SSMs, which extend structured state-space layers by making the system parameters input-dependent. In particular, the input and output projections $\mathbf{B}_t$, $\mathbf{C}_t$, as well as the discretization timescale Δ_t, are parameterized as functions of the input token x_t. This input-dependent parameterization enables content-based reasoning, which is essential for processing discrete modalities such as natural language.

The Mamba architecture wraps the selective SSM within a gated block that also incorporates a depthwise causal convolution to provide local context mixing. Formally, the block can be written as

$$\mathrm{Mamba}(u) = \mathrm{OutProj}(\mathrm{Norm}(\mathrm{SSM}(\mathrm{Conv}(x)),\, z)) \tag{2}$$

where

$$[z, x, \mathbf{B}, \mathbf{C}, \Delta] = \mathrm{InProj}(u).$$

Here, $\mathrm{Conv}(\cdot)$ denotes a depthwise causal convolution that mixes local context, z is a multiplicative gating signal, and $\mathrm{OutProj}$ projects the output back to the model dimension. Each Mamba block is therefore primarily parameterized by linear projections, a convolutional layer, and the internal components of the selective state-space module. The core of the architecture is the S6 layer.

S6 The S6 layer is a widely used variant of SSMs that employs real-valued, diagonal, and selective state-space dynamics. A standard real diagonal SSM is parameterized by a diagonal transition matrix $\mathbf{A} \in \mathbb{R}^{N' \times N'}$, input and output vectors $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{N' \times 1}$, and a timescale parameter $\Delta \in \mathbb{R}$. Each channel of such a model maps an input scalar sequence x to an output sequence y through the recurrent rule

$$\begin{aligned} h_t &= \bar{\mathbf{A}}\, h_{t-1} + \bar{\mathbf{B}}\, x_t, \qquad y_t = \mathbf{C}^\top h_t, \\ \bar{\mathbf{A}} &= f_A(\mathbf{A}, \Delta), \qquad \bar{\mathbf{B}} = f_B(\mathbf{B}, \Delta) \end{aligned} \tag{3}$$

are obtained through discretization functions f_A and f_B. The recurrence in Eq. equation 3 defines a mapping from $\mathbb{R}^L$ to $\mathbb{R}^L$ and can be computed efficiently in parallel using work-efficient scan algorithms (Smith et al., 2022) or via convolution using FFTs (Gu et al., 2021a). To process D channels, multiple independent SSM instances are applied in parallel.

The S6 layer extends this formulation through a *selective mechanism*, in which the system matrices become input-dependent. Specifically, shared projection matrices

$$S_B, S_C \in \mathbb{R}^{N' \times D}, \qquad A \in \mathbb{R}^{D \times N'}, \qquad S_\Delta \in \mathbb{R}^{1 \times D}$$

are used to generate time-varying parameters:

$$\begin{aligned} B_t = S_B X_t^*, \qquad C_t = S_C X_t^*, \qquad \Delta_t = \mathrm{softplus}(S_\Delta X_t^*), \\ \bar{A}_t = \exp(\Delta_t A), \qquad \bar{B}_t = \Delta_t B_t \end{aligned} \tag{4}$$

which lead to the time-varying recurrence

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t, \qquad y_t = C_t^\top h_t. \tag{5}$$

Mamba-3 The Mamba-3 (Lahoti et al., 2026) architecture was introduced recently, and presents three modifications to the selective SSM core: (i) an *exponential-trapezoidal* discretization, replacing the exponential-Euler scheme of Mamba-2 with a second-order three-term recurrence; (ii) data-dependent rotary embeddings (RoPE) applied to the input and readout matrices $\mathbf{B}$ and $\mathbf{C}$, which is shown to be equivalent to a complex-valued SSM and unlocks state-tracking; and (iii) a multi-input multi-output (MIMO) formulation that replaces the scalar SISO channel with rank-R vector inputs to amplify arithmetic intensity during decoding. The block additionally drops the short causal convolution and adds per-head learnable biases on $\mathbf{B}$ and $\mathbf{C}$ together with their own RMSNorm.

2.3 STRUCTURED STATE SPACE DUALITY

The key insight of Dao and Gu (Dao & Gu, 2024) is that when $\mathbf{A}$ is restricted to be scalar per head (i.e., $\mathbf{A} = a \cdot I$ with $a \in \mathbb{R}$), the SSM becomes equivalent to a form of causal linear attention. Specifically, unrolling the recurrence equation 1 yields the *semiseparable matrix* form:

$$M_{ij} = \mathbf{C}_i^\top \left(\prod_{k=j+1}^{i} \bar{a}_k \right) \bar{\mathbf{B}}_j, \quad i \geq j, \tag{6}$$

where $\bar{a}_t = \exp(\Delta_t \cdot a)$ is the scalar decay. Defining the lower-triangular causal mask $\mathbf{L}$ with entries $L_{ij} = \prod_{k=j+1}^{i} \bar{a}_k$ for $i \geq j$ (and $L_{ii} = 1$), the full output is

$$\mathbf{Y} = (\mathbf{L} \odot \mathbf{C}\mathbf{B}^\top)\,\mathbf{X}. \tag{7}$$

This is precisely causal linear attention with $Q \leftrightarrow \mathbf{C}$, $K \leftrightarrow \mathbf{B}$, $V \leftrightarrow \mathbf{X}$, and causal mask $\mathbf{L}$ replacing the softmax normalization. The SSD framework exploits this duality to design hardware-efficient chunked algorithms that combine the quadratic form (within chunks) with the linear recurrence (between chunks), achieving 2–8$\times$ speedups over Mamba's selective scan (Dao & Gu, 2024).

2.4 Self Attention

Self-attention constitutes a core element of Transformer architectures (Vaswani et al., 2017) and has been pivotal in driving recent progress in both natural language processing and computer vision. This mechanism allows a model to dynamically weigh the influence of different tokens within a sequence by modeling pairwise interactions, effectively determining which elements of the input should be emphasized.

Formally, the self-attention operation is defined as:

$$\mathrm{Attn}(Q, K, V) = \alpha V, \quad \alpha = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \tag{8}$$

where Q, K, and V denote the queries, keys, and values, respectively, and d_k is the dimensionality of the keys. To capture a wider range of dependencies, Transformers employ H parallel attention heads. This multi-head structure allows the model to attend to different subspaces of the input simultaneously, thereby enhancing its representational power.

2.5 Differential Transformer

The Differential Transformer (Ye et al., 2024) modifies standard multi-head attention by splitting each head's queries and keys into two groups and computing the output as the *difference* of the resulting attention maps:

$$\mathrm{DiffAttn}(Q_1, K_1, Q_2, K_2, V) = \left(\mathrm{softmax}\left(\frac{Q_1 K_1^\top}{\sqrt{d}}\right) - \lambda\,\mathrm{softmax}\left(\frac{Q_2 K_2^\top}{\sqrt{d}}\right)\right)V, \tag{9}$$

where Q_1, Q_2 are the paired query projections, K_1, K_2 are the paired keys projections, λ is a learnable scalar and V is shared across both branches. The intuition is that shared noise patterns in the two attention maps cancel upon subtraction, yielding sparser, more informative attention. This is analogous to *differential amplifiers* in electrical engineering, where common-mode noise is rejected by taking the difference of two correlated signals (Ye et al., 2024).

2.6 Differential Mamba

Inspired by the effectiveness of differential constructions in attention-based architectures, Differential Mamba (Schneider et al., 2025) transfers this principle to the selective state-space model Mamba. Prior work has shown that the Mamba block admits an interpretation within a unified attention framework (Zimerman et al., 2024). Building on this viewpoint, Differential Mamba is formulated as the difference between two parallel Mamba branches operating on the same input:

$$\text{Diff-Mamba}(X) = \mathrm{Mamba}_1(X) - \sigma(\lambda)\,\mathrm{Mamba}_2(X), \tag{10}$$

where λ is a learnable scalar parameter constrained to remain positive, and $\sigma(\cdot)$ denotes a sigmoid function introduced to promote numerical stability during optimization.

To further improve training stability and alleviate potential scale discrepancies between the two branches, a normalization operator is applied following the subtraction. In order to maintain a fair comparison under a comparable parameter budget, the dimensionality of the linear projections in each branch is reduced by a factor of two, while keeping the overall number of SSM parameters unchanged.

3 METHOD

We derive a general formulation of differential computation for SSMs through SSD. Our formulation reveals differential computation as an intrinsic operation on state-space representations that can be interpreted as a selective readout with per-state soft gating. This perspective naturally leads to an exact single-pass realization and an efficient Single Differential Circuit (SDC) implementation. Then, we instantiate this framework in both Mamba-2 and Mamba-3 and present the new blocks.

3.1 FROM DIFFERENTIAL ATTENTION TO DIFFERENTIAL SSM

Structured state space duality establishes an equivalence between scalar-transition SSMs and structured masked linear attention. This correspondence motivates an SSM analogue of differential attention equation 9: we introduce two input and readout projection pairs while sharing the input signal and temporal dynamics. Unlike softmax attention, the resulting interaction kernels are signed and unnormalized. Our construction therefore transfers the subtractive structure of differential attention to the SSD setting without assuming an exact equivalence to its softmax formulation.

For clarity, we consider a scalar input channel with states $\mathbf{h}_t^{(r)} \in \mathbb{R}^N$ and projection vectors $\bar{\mathbf{B}}_{r,t}, \mathbf{C}_{r,t} \in \mathbb{R}^N$, where $r \in \{1,2\}$. Let $g_t = \sigma(\lambda_t) \in (0,1)$ denote an input-dependent mixing coefficient. The two branches are

$$\mathbf{h}_t^{(1)} = \bar{a}_t\, \mathbf{h}_{t-1}^{(1)} + \bar{\mathbf{B}}_{1,t}\, x_t, \qquad y_t^{(1)} = \mathbf{C}_{1,t}^\top \mathbf{h}_t^{(1)}, \tag{11}$$

$$\mathbf{h}_t^{(2)} = \bar{a}_t\, \mathbf{h}_{t-1}^{(2)} + \bar{\mathbf{B}}_{2,t}\, x_t, \qquad y_t^{(2)} = \mathbf{C}_{2,t}^\top \mathbf{h}_t^{(2)}, \tag{12}$$

$$y_t = y_t^{(1)} - g_t\, y_t^{(2)}. \tag{13}$$

Both branches share the scalar decay $\bar{a}_t$, determined by the transition parameter and timestep Δ_t, and receive the same input x_t. Their distinct projection pairs provide two learned representations of the input history under a common temporal weighting.

State Representations. Assuming zero initial states, unrolling the recurrences gives

$$\mathbf{h}_t^{(r)} = \sum_{j=1}^{t} L_{tj}\, \bar{\mathbf{B}}_{r,j}\, x_j, \qquad r \in \{1,2\}, \tag{14}$$

where $L_{tj} = \prod_{k=j+1}^{t} \bar{a}_k$ for $j \le t$, with $L_{tt} = 1$, and $L_{tj} = 0$ for $j > t$. The shared temporal kernel L determines how past inputs persist, while $\bar{\mathbf{B}}_{1,j}$ and $\bar{\mathbf{B}}_{2,j}$ determine how those inputs are represented in the two state banks.

Differential Interaction Kernel. Substituting equation 14 into the readout, each branch induces a token-interaction kernel

$$M_{tj}^{(r)} = L_{tj}\, \mathbf{C}_{r,t}^\top \bar{\mathbf{B}}_{r,j}, \qquad y_t^{(r)} = \sum_{j=1}^{t} M_{tj}^{(r)}\, x_j. \tag{15}$$

The differential output is therefore

$$y_t = \sum_{j=1}^{t} L_{tj} \left(\mathbf{C}_{1,t}^\top \bar{\mathbf{B}}_{1,j} - g_t\, \mathbf{C}_{2,t}^\top \bar{\mathbf{B}}_{2,j} \right) x_j, \tag{16}$$

This formulation exposes differential computation as an adaptive combination of two interaction kernels over the same token pairs. An interaction is attenuated by cancellation when $M_{tj}^{(1)}$ and $g_t M_{tj}^{(2)}$ have the same sign and similar magnitudes. Depending on their relative signs and magnitudes, subtraction can also reinforce or reverse an interaction. Accordingly, the mechanism provides signed modulation of historical input contributions.

3.1.1 SINGLE-PASS FORMULATION

To improve computational efficiency, we propose an equivalent formulation that avoids redundant computation. Leveraging the linearity of SSMs, the differential mechanism is embedded directly into the state-space dynamics rather than applied post hoc. This reformulation requires only a single SSM forward pass, instead of two, thereby significantly reducing computational overhead. The following proposition formalizes this equivalence by showing that the two-branch differential construction can be expressed as a single SSM operating over an augmented state space (proof in Appendix F).

Proposition 1 (Single-Pass Differential SSM)**.** *Consider the two-branch system equation 11–equation 13 with scalar state decay $\bar{a}_t \in \mathbb{R}$, input matrices $\mathbf{B}_{1,t}, \mathbf{B}_{2,t} \in \mathbb{R}^N$, readout matrices $\mathbf{C}_{1,t}, \mathbf{C}_{2,t} \in \mathbb{R}^N$, and differential gate $\sigma(\lambda_t) \in (0,1)$. Define the augmented matrices*

$$\tilde{\mathbf{B}}_t = \left[\bar{\mathbf{B}}_{1,t}\ \bar{\mathbf{B}}_{2,t}\right] \in \mathbb{R}^{2N}, \qquad \tilde{\mathbf{C}}_t = \left[\mathbf{C}_{1,t}\ -\sigma(\lambda_t)\mathbf{C}_{2,t}\right] \in \mathbb{R}^{2N}. \tag{17}$$

Then the differential output $\mathbf{y}_t = \mathbf{y}_t^{(1)} - \sigma(\lambda_t)\mathbf{y}_t^{(2)}$ is identically computed by the single SSM

$$\tilde{\mathbf{h}}_t = \bar{a}_t \tilde{\mathbf{h}}_{t-1} + \tilde{\mathbf{B}}_t x_t, \qquad \mathbf{y}_t = \tilde{\mathbf{C}}_t^\top \tilde{\mathbf{h}}_t, \tag{18}$$

with augmented state $\tilde{\mathbf{h}}_t \in \mathbb{R}^{2N}$. In SSD form, the full output sequence satisfies

$$\mathbf{Y} = \left(\mathbf{L} \odot \tilde{\mathbf{C}}\tilde{\mathbf{B}}^\top\right)\mathbf{X}, \tag{19}$$

where $\tilde{\mathbf{C}}\tilde{\mathbf{B}}^\top = \mathbf{C}_1\mathbf{B}_1^\top - \sigma(\Lambda) \odot (\mathbf{C}_2\mathbf{B}_2^\top)$ and $\Lambda \in \mathbb{R}^{L\times 1}$ is broadcast over the key dimension.

3.1.2 SINGLE DIFFERENTIAL CIRCUIT FORMULATION

We introduce a Single Differential Circuit (SDC) variant that applies differential computation to either the input or readout projections, reducing parameter count and computational cost relative to the full construction (Appendix E). These variants modulate state writes or readouts, respectively. Our ablation study (Section 4.4) shows that both variants benefit from RMSNorm after subtraction, with the normalized $\mathbf{C}$-only variant outperforming its $\mathbf{B}$-only counterpart and achieving performance comparable to the full differential model.

For the $\mathbf{C}$-only variant, we share the input projections, $\bar{\mathbf{B}}_{1,t} = \bar{\mathbf{B}}_{2,t} = \bar{\mathbf{B}}_t$, and initialize both states identically. Under the shared decay $\bar{a}_t$, the two states then coincide, yielding

$$\mathbf{h}_t = \bar{a}_t\, \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t\, x_t. \tag{20}$$

The unnormalized differential output can therefore be expressed exactly as

$$y_t = \left(\mathbf{C}_{1,t} - g_t\mathbf{C}_{2,t}\right)^\top \mathbf{h}_t, \qquad g_t = \sigma(\lambda_t). \tag{21}$$

For the implemented variant, we additionally normalize the differential readout:

$$\tilde{\mathbf{C}}_t = \mathrm{RMSNorm}(\mathbf{C}_{1,t} - g_t\mathbf{C}_{2,t}), \qquad y_t = \tilde{\mathbf{C}}_t^\top \mathbf{h}_t. \tag{22}$$

Normalization is an architectural addition to the exact unnormalized formulation. The resulting input-dependent readout modulates contributions from a shared recurrent state. It enables a single-pass SSM while preserving the original state dimension N (Figure 1).

3.2 DIFF-MAMBA-2 BLOCK

We instantiate the proposed differential SSM formulation within the Mamba-2 architecture by applying the effective readout matrix (Eq. 22). Unlike prior Differential Mamba (Schneider et al., 2025), which subtracts the outputs of two independent Mamba blocks, our construction embeds the differential mechanism directly within the block. We further parameterize the scaling factor $\sigma(\lambda_t)$ as an input-dependent projection, which we find substantially improves optimization stability. The SDC variant preserves the original state dimension while retaining most of the empirical benefits of the full differential model. Unless otherwise specified, all references to Diff-Mamba-2 throughout this work correspond to the $\mathbf{C}$-only variant due to its favorable efficiency-performance trade-off.

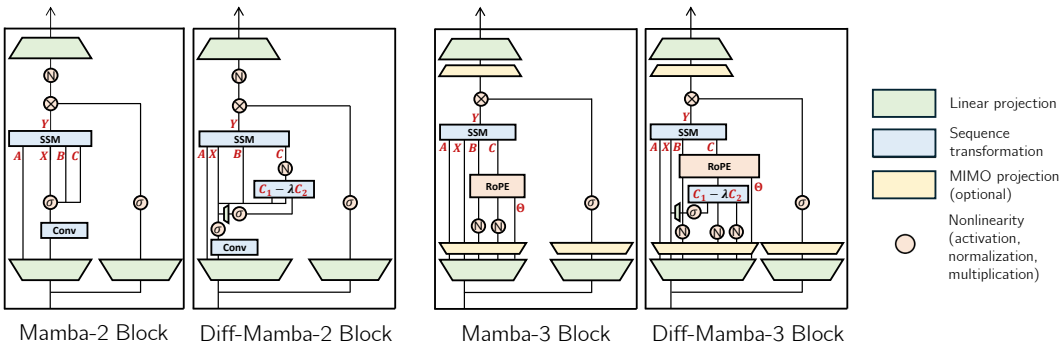


Figure 1: Our Diff-Mamba-2 and Diff-Mamba-3 blocks introduce a lightweight architectural modification that expands only the readout matrix $\mathbf{C}$, while applying a scaling by the parameter λ. This design preserves the structure of the original Mamba block while enabling the incorporation of an explicit differential mechanism with minimal overhead.

3.3 Diff-Mamba-3 Block

In a similar way, we use the proposed SDC variant, which naturally extends to Mamba-3. However, we adapt the construction to account for the structural differences in Mamba-3, where input and readout projections are already preceded by normalization layers with learned biases that effectively replace the convolutional component. In this setting, we apply the differential mechanism *after* the normalization stage, ensuring compatibility with the existing parameterization while preserving the stability benefits of normalized features. As in the Mamba-2 case, we retain the input-dependent gating factor $\sigma(\lambda_t)$ to modulate the differential interaction. Diff-Mamba-3 block is presented in Figure 1.

4 Experimental Results

In this section, we first pretrain Diff-Mamba-2 and Diff-Mamba-3 on the *FineWeb-Edu* dataset at multiple model scales (Section 4.1) and evaluate them across a diverse set of downstream benchmarks, where the differential variants consistently outperform their baselines and achieve state-of-the-art performance. We then assess their retrieval capabilities in Section 4.2, followed by an interpretability analysis based on token-to-token interaction heatmaps in Section 4.3. Finally, Section 4.4 presents architectural design ablations to justify our proposed architecture.

4.1 Language Modeling

All models are pretrained with 100B tokens of the FineWeb-Edu dataset (Penedo et al., 2024) with the Llama-3.1 tokenizer (Grattafiori et al., 2024) at a 2K context length with the same standard training protocol. Training and evaluation details can be found in Appendix C.2. Across our model scales, differential variants consistently outperform their Mamba baseline and other popular baselines on various downstream tasks, including Transformers, Gated Delta Networks (Yang et al., 2025b). Improvements are up to 1.5 average points (Table 1).

4.2 Retrieval Capabilities

Retrieval from long contexts remains a fundamental challenge for linear sequence models, whose compressed state representation must simultaneously preserve historical information and incorporate new inputs. Consequently, retrieval performance provides a direct measure of the quality of the learned memory representation (Arora et al., 2024b; 2025). We hypothesize that the differential state dynamics improve memory utilization, leading to stronger retrieval capabilities than existing linear architectures.

We evaluate this hypothesis on both real-world retrieval benchmarks (Table 2) and synthetic Needle-in-a-Haystack (NIAH) tasks (Table 3) (Hsieh et al., 2024) using the pretrained 1.5B-parameter models introduced in Section 4.1. Following prior work (Arora et al., 2025; 2024c), all real-world retrieval benchmarks are reformulated as cloze-style completion tasks to align with the autoregressive next-token prediction objective, using a maximum context length of 2K tokens. The results show that differential state dynamics consistently improve performance across most real-world retrieval benchmarks for both Mamba-2 and Mamba-3, with the largest gains observed on tasks involving semi-structured and unstructured documents, such as SWDE and FDA. In contrast, improvements do

Table 1: Downstream language modeling evaluations on models trained with 100B FineWeb-Edu tokens. All models are trained with the same procedure. Best results are bolded, and second best are underlined. Green indicates an improvement over the baseline, while gray indicates a decrease.

Model	LAMB. ppl ↓	LAMB. acc ↑	HellaS. acc_n ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc_n ↑	WinoGr. acc ↑	OBQA acc ↑	Average acc ↑	Δ
Transformer-180M	45.0	32.5	39.0	67.1	59.8	27.9	51.2	21.8	42.8	-
GDN-180M	40.8	31.3	40.2	66.3	62.3	28.2	51.7	22.0	43.2	-
Mamba-2-180M	41.8	30.9	40.1	66.8	60.1	27.3	52.0	23.2	42.9	-
Diff-Mamba-2-180M (Ours)	37.4	32.4	41.1	**67.7**	60.8	29.0	52.0	**24.2**	43.9	+1.0
Mamba-3-SISO-180M	37.7	32.5	40.8	66.1	61.5	27.9	52.0	22.8	43.4	-
Diff-Mamba-3-SISO-180M (Ours)	33.0	34.3	41.7	67.5	59.9	27.7	51.0	**24.2**	43.8	+0.4
Mamba-3-MIMO-180M	32.1	34.0	41.0	66.7	60.6	27.7	52.9	22.0	43.5	-
Diff-Mamba-3-MIMO-180M (Ours)	**29.3**	**36.2**	**42.4**	67.2	**62.8**	**29.6**	**54.9**	21.8	**45.0**	+1.5
Transformer-440M	21.2	41.7	50.5	69.9	67.6	34.6	**56.7**	26.0	49.6	-
GDN-440M	18.0	41.9	50.9	70.0	67.0	34.6	56.1	27.6	49.7	-
Mamba-2-440M	19.6	40.8	51.7	70.6	68.8	35.0	54.1	26.0	49.6	-
Diff-Mamba-2-440M (Ours)	18.2	41.2	51.8	71.7	**70.1**	35.2	54.8	26.2	50.1	+0.5
Mamba-3-SISO-440M	19.6	40.2	51.7	**71.9**	68.9	34.4	55.8	26.0	49.8	-
Diff-Mamba-3-SISO-440M (Ours)	17.7	42.9	52.3	70.7	69.7	**36.1**	55.0	27.0	50.5	+0.7
Mamba-3-MIMO-440M	**17.1**	**43.4**	**52.8**	70.8	69.6	35.6	56.3	**28.4**	**51.0**	-
Transformer-880M	15.0	44.7	57.2	72.6	71.6	39.2	57.7	26.8	52.8	-
GDN-880M	12.9	47.6	57.3	73.3	71.4	38.7	58.8	28.6	53.7	-
Mamba-2-880M	13.8	45.0	58.1	72.5	72.3	38.7	56.8	30.2	53.4	-
Diff-Mamba-2-880M (Ours)	12.8	48.0	58.3	73.5	72.8	38.7	58.8	**31.4**	54.5	+1.1
Mamba-3-SISO-880M	12.9	47.2	58.8	73.6	72.7	40.2	58.4	30.0	54.4	-
Diff-Mamba-3-SISO-880M (Ours)	12.0	49.2	59.1	73.4	73.1	40.1	58.9	29.8	54.8	+0.4
Mamba-3-MIMO-880M	**11.8**	**49.5**	**59.2**	**73.7**	**74.7**	**41.2**	**59.9**	28.6	**55.3**	-
Transformer-1.5B	11.1	50.3	60.6	73.8	74.0	40.4	58.7	29.6	55.4	-
GDN-1.5B	10.9	49.2	61.3	74.3	75.3	41.2	58.0	31.6	55.8	-
Mamba-2-1.5B	12.0	47.8	61.4	73.6	75.3	41.8	57.5	32.6	55.7	-
Diff-Mamba-2-1.5B (Ours)	11.7	48.2	61.2	74.6	75.5	43.6	60.9	31.4	56.5	+0.8
Mamba-3-SISO-1.5B	10.9	49.4	61.9	73.6	75.9	42.7	59.4	32.0	56.4	-
Diff-Mamba-3-SISO-1.5B (Ours)	10.6	50.9	62.3	75.0	**76.6**	43.9	**61.3**	32.4	57.5	+1.1
Mamba-3-MIMO-1.5B	10.2	51.7	62.3	**75.3**	76.5	**44.5**	60.6	32.6	57.6	-
Diff-Mamba-3-MIMO-1.5B (Ours)	**9.3**	**53.9**	**63.1**	74.0	76.3	43.4	61.2	**34.0**	**58.0**	+0.4

Table 2: Real-world retrieval capabilities. Tasks utilize cloze variants of the original datasets and are truncated to 2K length. Differential variants demonstrate strong associative recall, question-answering, and information extraction of semi-structured and unstructured data.

Model (1.5B)	SWDE	SQD.	FDA	TQA	NQ	Drop
GDN	32.7	40.0	28.3	63.5	25.7	24.5
Mamba-2	30.7	39.1	23.7	64.3	25.1	28.5
Diff-Mamba-2 (Ours)	35.5	40.0	**32.3**	64.5	25.0	**29.8**
Mamba-3 SISO	28.5	40.1	23.4	64.5	26.5	27.4
Diff-Mamba-3-SISO (Ours)	32.3	41.3	25.7	64.5	26.6	26.3
Mamba-3 MIMO	**36.3**	41.7	29.3	64.5	26.2	26.3
Diff-Mamba-3-MIMO (Ours)	**36.3**	**43.7**	22.7	**65.9**	**27.5**	26.9

not consistently transfer to the synthetic NIAH benchmarks, where the differential variants exhibit performance degradation on several configurations.

4.3 Model Interpretability

Interpretability is important for understanding and improving model architectures. While prior work has provided insights into Mamba's internal mechanisms, it does not explicitly decompose token-level contributions, limiting understanding of how information is selectively routed across layers. LaTIM (Pitorro & Treviso, 2025) addresses this by enabling token-level decomposition for Mamba, providing fine-grained attribution of token interactions. Using the ℓ_2 formulation, we compute token-to-token interaction heatmaps for both Mamba-2 and Diff-Mamba-2 on the copying task (Jelassi et al., 2024). As shown in Figure 2, Diff-Mamba-2 exhibits sharper interaction patterns, with stronger structure along both the main diagonal and a secondary diagonal corresponding to the copied sequence. This suggests more precise alignment between source and target tokens, indicating improved selective retrieval behavior during copying.

Table 3: Synthetic retrieval capabilities measured by needle-in-a-haystack (NIAH) tasks across increasing context lengths. Differential variants exhibit mixed performance, improving several settings while degrading others.

Model (1.5B)	NIAH-Single-1			NIAH-Single-2			NIAH-Single-3		
Context Length	1024	2048	4096	1024	2048	4096	1024	2048	4096
GDN	**100.0**	**100.0**	**99.8**	**100.0**	93.8	49.8	83.8	68.4	**34.2**
Mamba-2	**100.0**	99.6	62.0	**100.0**	53.8	11.8	**95.8**	**87.4**	13.4
Diff-Mamba-2 (Ours)	**100.0**	**100.0**	24.2	**100.0**	99.0	**52.6**	93.6	71.6	22.2
Mamba-3 SISO	**100.0**	**100.0**	88.2	**100.0**	95.4	50.6	92.4	81.4	**34.2**
Diff-Mamba-3-SISO (ours)	**100.0**	**100.0**	91.2	**100.0**	**100.0**	33.0	77.2	62.2	17.2
Mamba-3 MIMO	**100.0**	**100.0**	93.0	**100.0**	86.0	40.4	**95.8**	84.4	25.6
Diff-Mamba-3-MIMO	**100.0**	**100.0**	87.6	**100.0**	97.0	24.8	61.4	54.6	17.0

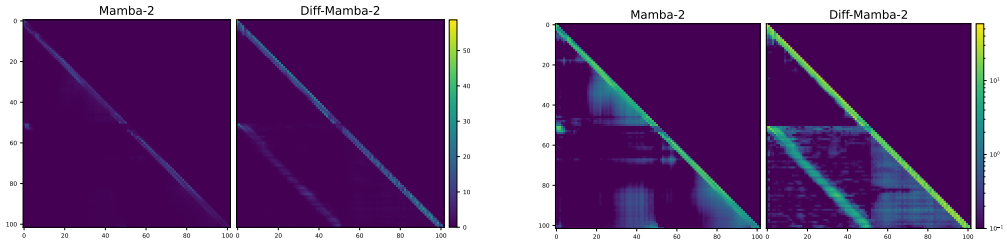


Figure 2: From left to right: we present heatmaps of token-to-token interactions for Mamba-2 and Diff-Mamba-2 on the copying task. The interaction between source and copied tokens (along the diagonal line) is more clearly highlighted in Diff-Mamba-2. The first subfigure visualizes $\mathrm{LaTIM}(\ell_2)$, and the second depicts $\log\left(\mathrm{LaTIM}(\ell_2)\right)$ for improved contrast.

4.4 Architectural Ablation

To validate the proposed architectural design choices, we conduct a systematic comparison of model variants. First, we isolate the subtraction component from projections' expansion or additional gate, demonstrating the gains are from the differential mechanism. Then, we test multiple circuit configurations; both $\mathbf{B}$ and $\mathbf{C}$, $\mathbf{B}$-only or $\mathbf{C}$-only (Table 4). For each configuration, we further examine the effect of applying RMSNorm after the differential subtraction. Both ablations are measured with PPL on the test split of PG19 (Rae et al., 2019) dataset, a benchmark for measuring long-range dependency modeling. Training was conducted with scaling-law settings on The-Pile. The results indicate dual differential circuit outperforms, while $\mathbf{C}$-only with normalization attains comparable results. Despite the marginally superior performance of the dual differential design, we adopt the $\mathbf{C}$-only variant due to its more favorable efficiency-performance trade-off (Appendix E).

(a) Differential components.

Expand	Gate	Subtract	PPL	# Params
×	×	×	39.688	368M
✓	×	×	39.167	381M
✓	✓	×	39.173	381M
✓	✓	✓	**38.870**	381M

(b) Circuit and normalization.

Differential Circuit	w/o Nrm	w. Nrm	# Params
C,B	**38.870**	—	381M
B	39.435	39.130	374M
C	39.364	39.017	374M

Table 4: Ablation studies of the differential architecture on the PG19 test set. (a) examine the subtraction effect while (b) compares circuit configurations. Lower PPL is better.

5 Conclusion

In this paper, we introduced a principled framework for differential modeling in state space architectures through the lens of SSD, yielding two novel architectures: Diff-Mamba-2 and Diff-Mamba-3. Our approach achieves effective noise cancellation with minimal efficiency overhead of the underlying architecture. Through experimental results, we show consistent improvements over popular baselines across downstream tasks, with differential variants achieving gains of up to 1.5 points on average. On real-world retrieval benchmarks, the differential variants consistently improve performance across most tasks. Interpretability analysis further reveals that the differential mechanism produces sharper token-to-token interaction patterns, suggesting more precise selective retrieval behavior. Our work advances research on efficient sequence models by demonstrating that principled noise suppression can be embedded explicitly into SSM dynamics, addressing the over-allocation of representational capacity to irrelevant context.

AI USE STATEMENT

Generative AI tools were used to assist with writing and improving clarity, and to assist with the retrieval and discovery of finding relative work. The scientific ideas, methodology, experimental design, results, and conclusions were developed and validated by the authors. All parts where AI tools were involved were reviewed, edited, and verified by the authors.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Yaroslav Aksenov, Nikita Balagansky, Sofia Lo Cicero Vaina, Boris Shaposhnikov, Alexey Gorbatovski, and Daniil Gavrilov. Linear transformers with learnable kernel functions are better in-context models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9584–9597, 2024.

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2505.15105*, 2025.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International conference on learning representations*, volume 2024, pp. 15664–15730, 2024a.

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 arXiv:2402.18668*, 2024b.

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, 2024. *URL https://arxiv. org/abs/2407.05483*, 2024c.

Sangmin Bae, Bilge Acun, Chien-Yu Lin, Haroun Habeeb, Seungyeon Kim, Liang Luo, Junjie Wang, and Carole-Jean Wu. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv preprint arXiv:2510.04800*, 2025.

Ethan Baron, Itamar Zimerman, and Lior Wolf. 2-d ssm: A general spatial layer for visual transformers. *arXiv preprint arXiv:2306.06635*, 2023.

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. *Advances in Neural Information Processing Systems*, 37:107547–107603, 2024.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical intuition by question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Yueyang Cang, Yuhang Liu, Xiaoteng Zhang, Li Shi, and Wenge Que. Shared diff transformer. *arXiv preprint arXiv:2501.17900*, 2025a.

Yueyang Cang, Yuhang Liu, Xiaoteng Zhang, Erlu Zhao, and Li Shi. DINT transformer. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9801–9809, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.495. URL https://aclanthology.org/2025.emnlp-main.495/.

Javier Salazar Cavazos, Maximillian Egan, Krisanne Litinas, Benjamin Hampstead, and Scott Peltier. Behavior score prediction in resting-state functional mri by deep state space modeling. *arXiv preprint arXiv:2602.07131*, 2026.

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

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 arXiv:1803.05457*, 2018.

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

Shmuel Bar David, Itamar Zimerman, Eliya Nachmani, and Lior Wolf. Decision s4: Efficient sequence-based rl via state spaces layers. In *The Eleventh International Conference on Learning Representations*, 2022.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, ZIJIA CHEN, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=A1ztozypga.

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.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*, 2024.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International conference on machine learning*, pp. 7616–7633. PMLR, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg Franke, Frank Hutter, et al. Unlocking state-tracking in linear rnns through negative eigenvalues. In *International Conference on Learning Representations*, volume 2025, pp. 36565–36597, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

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

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated deltanet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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 arXiv:2404.06654*, 2024.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and eran malach. Repeat after me: Transformers are better than state space models at copying. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=duRRoGeoQT.

Mandar Joshi, Eunsol Choi, Daniel S 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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

Chaerin Kong, Jiho Jang, and Nojun Kwak. Understanding differential transformer unchains pretrained self-attentions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=cWlRC064GK.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Aakash Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, J Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HwCvaJOiCj.

Barak Lenz, Opher Lieber, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M. Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, Gal Cohen, Gal Shachaf, Haim Rozenblum, Hofit Bata, Ido Blass, Inbal Magar, Itay Dalmedigos, Jhonathan Osin, Julie Fadlon, Maria Rozman, Matan Danos, Michael Gokhman, Mor Zusman, Naama Gidron, Nir Ratner, Noam Gat, Noam Rozen, Oded Fried, Ohad Leshno, Omer Antverg, Omri Abend, Or Dagan, Orit Cohavi, Raz Alon, Ro'i Belson, Roi Cohen, Rom Gilad, Roman Glozman, Shahar Lev, Shai Shalev-Shwartz, Shaked Haim Meirom, Tal Delbari, Tal Ness, Tomer Asida, Tom Ben Gal, Tom Braude, Uriya Pumerantz, Josh Cohen, Yonatan Belinkov, Yuval Globerson, Yuval Peleg Levy, and Yoav Shoham. Jamba: Hybrid transformer-mamba language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JFPaD7lpBD.

Junghwan Lim, Sungmin Lee, Dongseok Kim, Wai Ting Cheung, Beomgyu Kim, Taehwan Kim, Haesol Lee, Junhyeok Lee, Dongpin Oh, and Eunhwan Park. Grouped differential attention. *arXiv preprint arXiv:2510.06949*, 2025.

Zhixuan Lin, Evgenii Nikishin, Xu He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. In *International Conference on Learning Representations*, volume 2025, pp. 69704–69738, 2025.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. *URL https://arxiv.org/abs/2307.03172*, 2, 2023.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:47016–47031, 2023.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. *arXiv preprint arXiv:2404.08819*, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. URL https://aclanthology.org/D18-1260.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

NVIDIA. Nemotron 3 Nano: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for Agentic reasoning, 2025. URL https://arxiv.org/abs/2512.20848. Technical report.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Razvan Pascanu, Marco Baroni, and Gemma Boleda. 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)*. Association for Computational Linguistics, 2016. URL https://aclanthology.org/P16-1144.

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 *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

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

Hugo Pitorro and Marcos Vinicius Treviso. LaTIM: Measuring latent token-to-token interactions in mamba models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24478–24493, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1194. URL https://aclanthology.org/2025.acl-long.1194/.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

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

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *Advances in Neural Information Processing Systems*, 38:100092–100118, 2026.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

George Saon, Ankit Gupta, and Xiaodong Cui. Diagonal state space augmented transformers for speech recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

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

Nadav Schneider, Itamar Zimerman, and Eliya Nachmani. Differential mamba. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2561–2575, 2025.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Improving state-tracking in linear rnns via householder products. *Advances in Neural Information Processing Systems*, 38:153738–153782, 2026.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

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

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Tsubasa Takahashi, Shojiro Yamabe, Futa Kai Waseda, and Kento Sasaki. Understanding sensitivity of differential attention through the lens of adversarial robustness. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kwy0cKmTm3.

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.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

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

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37: 115491–115522, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=r8H7xhYPwz.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. 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, et al. Gated slot attention for efficient linear-time sequence modeling. *Advances in Neural Information Processing Systems*, 37:116870–116898, 2024.

Itamar Zimerman, Ameen Ali, and Lior Wolf. A unified implicit attention formulation for gated-linear recurrent sequence models. *arXiv preprint arXiv:2405.16504*, 2024.

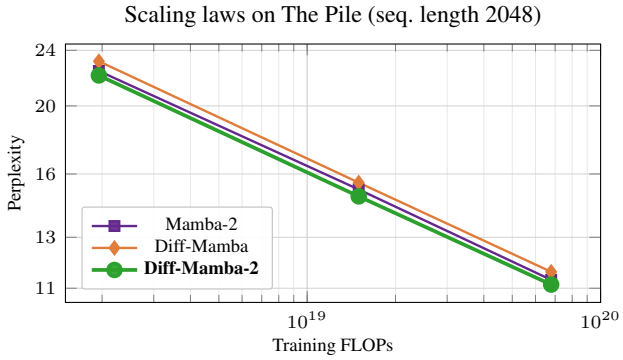
Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355*, 2024.

A Related Work

Our work connects differential attention with efficient recurrent modeling. Related approaches explore subtractive attention and recurrent outputs (Ye et al., 2024; Cang et al., 2025b; Schneider et al., 2025; Cang et al., 2025a; Lim et al., 2025), alongside adaptive attention gating and forgetting (Qiu et al., 2026; Lin et al., 2025). These directions build on kernelized and learnable linear attention (Katharopoulos et al., 2020; Choromanski et al., 2020; Peng et al., 2021; Aksenov et al., 2024) and connections between attention, fast-weight memory, and recurrence (Schlag et al., 2021; Sun et al., 2023; Dao & Gu, 2024). Complementary architectures develop structured state spaces and long convolutions (Gu et al., 2021a; Smith et al., 2022; Poli et al., 2023; Gu & Dao, 2024; Lahoti et al., 2026). Within recurrent models, research investigates gated and slot-based memories (Yang et al., 2023; Beck et al., 2024; Qin et al., 2024; Zhang et al., 2024), delta-rule updates and adaptive forgetting (Yang et al., 2024; 2025b; Hatamizadeh et al., 2026), and learned memory updates at test time (Sun et al., 2024; Behrouz et al., 2026). Related analyses examine state-tracking limitations and more expressive transitions (Merrill et al., 2024; Grazzi et al., 2025; Siems et al., 2026), while retrieval studies characterize associative recall, copying, and efficiency trade-offs (Arora et al., 2024a;b; Jelassi et al., 2024).

B Scaling Laws

While Diff-Mamba demonstrates improved long-context and retrieval capabilities compared to Mamba-2, it struggles when scaling to a large number of layers (Schneider et al., 2025). We illustrate this behavior using scaling laws (Hoffmann et al., 2022; Kaplan et al., 2020) experiments (Figure 3). Specifically, we pretrain Mamba-2, Diff-Mamba, and Diff-Mamba-2 models at three parameter scales; 130M, 370M, and 780M parameters on *The Pile* dataset and evaluate on its test partition. The results reveal that Diff-Mamba underperforms relative to Mamba in terms of perplexity, indicating slower convergence, while Diff-Mamba-2 achieves lower perplexity than both Diff-Mamba and Mamba-2 across scales, demonstrating improved training efficiency and faster convergence. Training details can be found in Appendix C.1.



	Model size		
Model	130M	370M	780M
Mamba-2	22.43	15.21	11.31
Diff-Mamba	23.15	15.55	11.61
Diff-Mamba-2	**22.10**	**14.87**	**11.13**

Figure 3: Scaling laws on The Pile at ≈130M, 370M, and 780M parameters. Diff-Mamba-2 attains the lowest perplexity at every scale, while Diff-Mamba lags behind even vanilla Mamba-2.

C Experimental Details

This section describes the settings for each experimental design through the paper. All the models were trained with B200 GPU.

C.1 Scaling Laws

Scaling law experiments generally followed the GPT3 recipe. All models were trained on the Pile with the GPT2 tokenizer.

C.2 Language Modeling

Our pretraining procedures follow those of Mamba-3 section D. We pretrain our differential variants at each scale following the same procedure and with bfloat16 on B200 GPUs. Baseline results are

Table 5: Model configurations and training hyperparameters.

Params	n_layers	d_model	d_head	Training Steps	Learning Rate	Batch Size	Tokens
130M	24	768	64	5000	6e-4	0.5M tokens	2.5B
370M	48	1024	64	13500	3e-4	0.5M tokens	7B
780M	48	1536	64	29000	2.5e-4	0.5M tokens	15B

from the Mamba-3 paper (Lahoti et al., 2026). The Transformer baselines follow (Dao & Gu, 2024), and the GDN baselines follow (Yang et al., 2025b), where $q, k_{dim} = 128$, $v_{dim} = 256$.

We utilize LM Evaluation Harness (Gao et al., 2024) to test the zero-shot language modeling capabilities of our pretrained model on LAMBADA (OpenAI version) (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), Arc-Easy/Arc-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2019), and OpenBookQA (Mihaylov et al., 2018).

C.3 Retrieval Capabilities

Baseline results are from the Mamba-3 paper. We follow the same evaluation procedure: For our real-world retrieval tasks, we evaluate on the common suite consisting of SWDE (Arora et al., 2024b), SQuAD (Rajpurkar et al., 2018), FDA (Arora et al., 2024b), TriviaQA (Joshi et al., 2017), NQ (Kwiatkowski et al., 2019), and DROP (Dua et al., 2019). We utilize the cloze-formatted version of the aforementioned tasks provided by (Arora et al., 2024b) and (Arora et al., 2025), as the original datasets are in a question-answering format, making it challenging for solely pretrained models. All tasks were truncated to match the training context length. The synthetic NIAH tasks (Hsieh et al., 2024) were also run with LM Evaluation Harness.

C.4 Model Interpretability

We analyze token-level interactions using LaTIM (Pitorro & Treviso, 2025), following the implementation provided by the authors. Specifically, we train Mamba-2 and Diff-Mamba-2 models on the copying task with identical experiment settings as the original LaTIM paper. Then, we use the ℓ_2 formulation to compute token-to-token interaction matrices for both. The input consists of a sequence followed by a delimiter and a repeated copy of the initial sequence. For visualization, we compute LaTIM scores for the third model layer (out of 8). All visualizations use the same input sequence and identical plotting settings for both models to enable direct qualitative comparison.

D Model Analysis

Implicit and Explicit Differential Modeling. The placement of the differential subtraction fundamentally shapes its corrective effect. In Diff-Mamba (Schneider et al., 2025), two independent Mamba blocks with separate state-space parameters process the same input, and their outputs are subtracted. Because the branches undergo entirely distinct transformations, their representations may diverge substantially, yielding only an indirect, coarse-grained correction–an *implicit* differential model. In Diff-Mamba-2, the subtraction occurs within a single pipeline: both branches share the same projections, recurrent state, decay $\bar{a}_t$, and input x_t, differing only in their input and readout matrices. This tight coupling ensures precise common-mode cancellation, constituting an *explicit* differential model.

We hypothesize that these two designs are endpoints of a broader *Differential Spectrum*: the greater the architectural proximity between the subtracted branches, the more fine-grained and effective the resulted noise cancellation. Differential modeling admits a continuum of configurations, with intermediate variants.

E Efficiency Benchmark

It is essential that the proposed differential extensions preserve the computational benefits of the Mamba. To this end, we evaluate both the computational efficiency and memory footprint of the proposed models. Specifically, we benchmark Diff-Mamba-2 SDC (374M params) against alternative differential designs, including Diff-Mamba (382M params), single-pass SSM formulation (381M params), and the dual-branch SSM formulation (381M params). We further compare Diff-Mamba-2

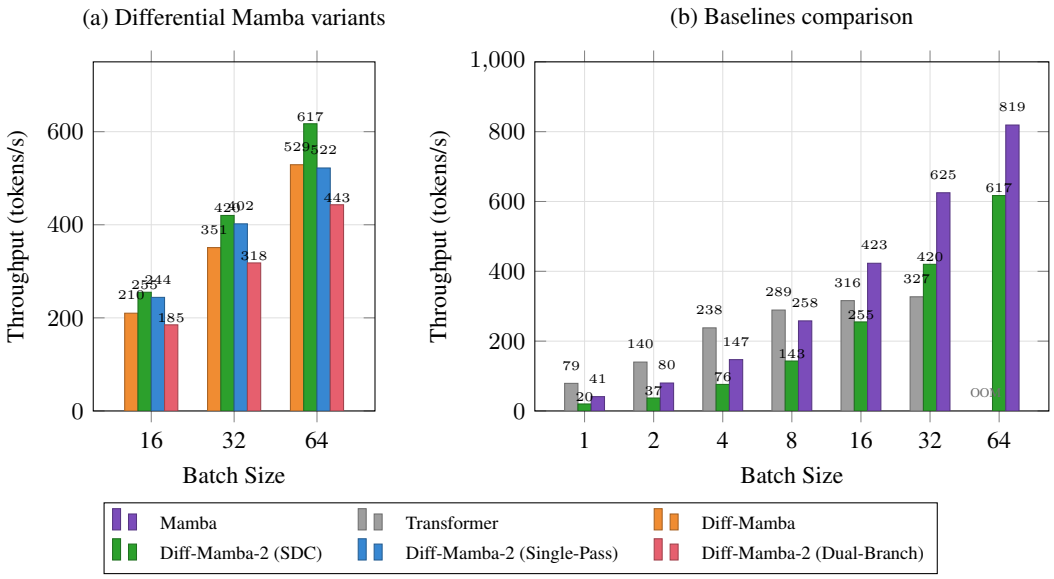


Figure 4: Inference throughput comparison. (a) Throughput scaling across Differential Mamba architectural variants at large batch sizes. (b) Comparison of the best Diff-Mamba-2 variant (SDC) against Mamba and Transformer baselines.

SDC against baseline Mamba (368M params) in terms of inference throughput, memory consumption, and forward-pass latency. These measurements provide a comprehensive characterization of the computational trade-offs introduced by the proposed architectural modifications. Our results demonstrate that Diff-Mamba-2 SDC achieves superior inference throughput compared to other differential variants while largely preserving the efficiency characteristics that make Mamba architectures attractive for long-sequence modeling.

E.1 End-to-End Inference

We evaluate end-to-end inference throughput across varying batch sizes on an A100 GPU with 80GB of memory (Figure 4). The benchmark uses a prompt length of 2048 tokens and generates 128 new tokens with the mentioned untrained models. Throughput, measured in tokens/s, is computed as batch size$\times 128$ / inference time. As shown in Figure 4, Diff-Mamba-2 SDC consistently outperforms other variants. At batch size 64, Diff-Mamba-2 SDC and Mamba-2 reach 617 and 819 tokens/s, respectively.

E.2 Memory and Latency Benchmark

We evaluate the memory consumption and forward-pass latency of Diff-Mamba-2 across varying batch sizes and sequence lengths, as summarized in Table 6. Memory usage was measured after a forward pass using sequences of length 2048, while latency was evaluated across progressively longer contexts. Diff-Mamba-2 preserves the efficiency characteristics of Mamba remarkably well. Across all evaluated batch sizes, both models exhibit identical GPU memory consumption. Similarly, forward-pass latency remains highly comparable across all tested sequence lengths, with only negligible differences even at long contexts.

Table 6: Memory usage and forward-pass latency comparisons.

(a) Memory Benchmark (GB)

Batch Size	Mamba	Diff-Mamba-2
1	2.20	2.20
2	3.74	3.74
4	6.06	6.06
8	10.69	10.69
16	19.96	19.96
32	38.49	38.49
64	75.57	75.57

(b) Forward-Pass Latency (s)

Seq Length	Mamba	Diff-Mamba-2
512	0.079	0.058
1024	0.089	0.066
2048	0.145	0.118
4096	0.249	0.240
8192	0.469	0.447
16384	0.841	0.858
32768	1.704	1.726
65536	3.384	3.408

F PROOFS

The following proposition shows that the two-branch differential computation equation 11–equation 13 is algebraically equivalent to a single SSM operating over an augmented state space of dimension $2N$.

Proposition 1. [Single-Pass Differential SSM] Consider the two-branch system equation 11–equation 13 with scalar state decay $\bar{a}_t \in \mathbb{R}$, input matrices $\mathbf{B}_{1,t}, \mathbf{B}_{2,t} \in \mathbb{R}^N$, readout matrices $\mathbf{C}_{1,t}, \mathbf{C}_{2,t} \in \mathbb{R}^N$, and differential gate $\sigma(\lambda_t) \in (0,1)$. Define the augmented matrices

$$\tilde{\mathbf{B}}_t = \begin{bmatrix} \bar{\mathbf{B}}_{1,t} \\ \bar{\mathbf{B}}_{2,t} \end{bmatrix} \in \mathbb{R}^{2N}, \qquad \tilde{\mathbf{C}}_t = \begin{bmatrix} \mathbf{C}_{1,t} \\ -\sigma(\lambda_t)\,\mathbf{C}_{2,t} \end{bmatrix} \in \mathbb{R}^{2N}. \tag{23}$$

Then the differential output $\mathbf{y}_t = \mathbf{y}_t^{(1)} - \sigma(\lambda_t)\,\mathbf{y}_t^{(2)}$ is identically computed by the single SSM

$$\tilde{\mathbf{h}}_t = \bar{a}_t\,\tilde{\mathbf{h}}_{t-1} + \tilde{\mathbf{B}}_t\,x_t, \qquad \mathbf{y}_t = \tilde{\mathbf{C}}_t^\top \tilde{\mathbf{h}}_t, \tag{24}$$

with augmented state $\tilde{\mathbf{h}}_t \in \mathbb{R}^{2N}$ and shared scalar decay $\bar{a}_t$. In SSD form, the full output sequence satisfies

$$\mathbf{Y} = \left(\mathbf{L} \odot \tilde{\mathbf{C}}\,\tilde{\mathbf{B}}^\top\right)\mathbf{X}, \tag{25}$$

where $\tilde{\mathbf{C}}\,\tilde{\mathbf{B}}^\top = \mathbf{C}_1\mathbf{B}_1^\top - \sigma(\Lambda) \odot (\mathbf{C}_2\mathbf{B}_2^\top)$ and $\Lambda \in \mathbb{R}^{L\times 1}$ is broadcast over the key dimension.

Proof. **Recurrence equivalence.** Partition the augmented state as $\tilde{\mathbf{h}}_t = [\mathbf{h}_t^{(1)};\, \mathbf{h}_t^{(2)}]$ with $\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)} \in \mathbb{R}^N$. Since $\bar{a}_t$ is scalar, it acts identically on both partitions:

$$\tilde{\mathbf{h}}_t = \bar{a}_t\tilde{\mathbf{h}}_{t-1} + \tilde{\mathbf{B}}_t x_t = \begin{bmatrix} \bar{a}_t\mathbf{h}_{t-1}^{(1)} + \bar{\mathbf{B}}_{1,t}x_t \\ \bar{a}_t\mathbf{h}_{t-1}^{(2)} + \bar{\mathbf{B}}_{2,t}x_t \end{bmatrix}, \tag{26}$$

so $\mathbf{h}_t^{(1)}$ and $\mathbf{h}_t^{(2)}$ evolve exactly as in equation 11–equation 12. The readout then recovers the differential output:

$$Y_t = \tilde{\mathbf{C}}_t^\top\tilde{\mathbf{h}}_t = [\mathbf{C}_{1,t};\, -\sigma(\lambda_t)\mathbf{C}_{2,t}]^\top[\mathbf{h}_t^{(1)};\, \mathbf{h}_t^{(2)}] = \mathbf{C}_{1,t}^\top\mathbf{h}_t^{(1)} - \sigma(\lambda_t)\,\mathbf{C}_{2,t}^\top\mathbf{h}_t^{(2)} = \mathbf{y}_t^{(1)} - \sigma(\lambda_t)\,\mathbf{y}_t^{(2)}. \tag{27}$$

Preserving structured state space duality. Unrolling the augmented recurrence gives

$$\tilde{\mathbf{h}}_t = \sum_{j=1}^{t}\left(\prod_{k=j+1}^{t}\bar{a}_k\right)\tilde{\mathbf{B}}_j\,x_j = \sum_{j=1}^{t} L_{tj}\,\tilde{\mathbf{B}}_j\,x_j. \tag{28}$$

The output at position i is therefore

$$y_i = \tilde{\mathbf{C}}_i^\top\tilde{\mathbf{h}}_i = \sum_{j=1}^{i} L_{ij}\,\tilde{\mathbf{C}}_i^\top\tilde{\mathbf{B}}_j\,x_j. \tag{29}$$

Expanding the inner product using the definition of $\tilde{\mathbf{B}}_t$ and $\tilde{\mathbf{C}}_t$ in equation 23:

$$\tilde{\mathbf{C}}_i^\top\tilde{\mathbf{B}}_j = \mathbf{C}_{1,i}^\top\bar{\mathbf{B}}_{1,j} + (-\sigma(\lambda_i))\,\mathbf{C}_{2,i}^\top\bar{\mathbf{B}}_{2,j} = \mathbf{C}_{1,i}^\top\bar{\mathbf{B}}_{1,j} - \sigma(\lambda_i)\,\mathbf{C}_{2,i}^\top\bar{\mathbf{B}}_{2,j}, \tag{30}$$

so

$$y_i = \sum_{j=1}^{i} L_{ij}\left(\mathbf{C}_{1,i}^\top\bar{\mathbf{B}}_{1,j} - \sigma(\lambda_i)\,\mathbf{C}_{2,i}^\top\bar{\mathbf{B}}_{2,j}\right)x_j = y_i^{(1)} - \sigma(\lambda_i)\,y_i^{(2)}. \tag{31}$$

Writing this in matrix form with $\mathbf{L}$ the lower-triangular causal mask yields $\mathbf{Y} = (\mathbf{L} \odot \tilde{\mathbf{C}}\tilde{\mathbf{B}}^\top)\mathbf{X}$, where the (i,j)-th entry of $\tilde{\mathbf{C}}\tilde{\mathbf{B}}^\top$ decomposes as $\mathbf{C}_{1,i}^\top\mathbf{B}_{1,j} - \sigma(\lambda_i)\mathbf{C}_{2,i}^\top\mathbf{B}_{2,j}$, or equivalently $\tilde{\mathbf{C}}\tilde{\mathbf{B}}^\top = \mathbf{C}_1\mathbf{B}_1^\top - \sigma(\Lambda) \odot (\mathbf{C}_2\mathbf{B}_2^\top)$. □