ALTAMBA: Alternating Mamba with Peristaltic Normalization

Learnable Destructive Interference via The Razor Blending Operation

Scott Seto

Independent Researcher

February 2026

Abstract

While modern language models rely on additive residual streams ($\mathbf{y} = \mathbf{x} + f(\mathbf{x})$), we show that *subtractive signal extraction* yields significant gains. **ALTAMBA** (Alternating Mamba) is a dual-path architecture that runs a Transformer and State Space Model (SSM) in parallel at every layer, blending them through learned destructive and constructive interference. Two innovations enable this:

1. **The RAZOR**—a learnable blending operation $(c - W) \cdot \text{LN}(\mathbf{x}_T) + (c' - W') \cdot \mathbf{x}_S$ with per-layer parameters and output scaling—that discovers optimal interference coefficients automatically.
2. **Peristaltic Normalization**—alternating Post-LayerNorm (variance clamping for precise denoising) and Pre-LayerNorm (flexible variance for gradient flow)—with bottleneck compression (0.75×) and phase-alternating output scaling (+1.5×/−1.0×).

On Common Pile (arXiv), ALTAMBA achieves **9.35%** (402M), **9.40%** (1.08B), and **7.66%** (1.78B) validation loss improvement over parameter-matched Jamba baselines, with the 1.78B model reaching a val loss of **2.26**. Through learnable mixing coefficients, the network spontaneously develops interpretable strategies including depth-dependent denoising strength and three-stage gain staging. Our core finding is that *normalization diversity*—using Post-LN and Pre-LN for different computational roles rather than choosing one universally—unlocks denoising capabilities that neither alone can provide.

1 Introduction

1.1 The Abandoned Normalization

In 2018, BERT Devlin et al. [2019] introduced Post-LayerNorm (Post-LN) and achieved breakthrough results in natural language understanding. By 2019, GPT-2 Radford et al. [2019] switched to Pre-LayerNorm (Pre-LN) for improved training stability at scale. Since then, every major language model—GPT-3 Brown et al. [2020], LLaMA Touvron et al. [2023], and their successors—has used exclusively Pre-LN.

Post-LN was abandoned, not refined.

We argue this abandonment was premature. Post-LN possesses unique properties—*unit variance clamping* and *zero mean forcing*—that make it optimal for a task that modern language models do not explicitly perform: precise noise subtraction through destructive interference. The key insight is that Post-LN and Pre-LN are not interchangeable alternatives but complementary tools suited to different computational roles.

1.2 The Razor Blending Operation

We introduce the Razor, a learnable blending operation that enables per-layer destructive or constructive interference between two signal paths:

$$\mathbf{y}_i = s_i \cdot \left[\left(c_1^{(i)} - W_1^{(i)}\right) \cdot \mathrm{LN}(\mathbf{x}_{\text{denoiser}}) + \left(c_2^{(i)} - W_2^{(i)}\right) \cdot \mathbf{x}_{\text{main}} \right] \tag{1}$$

where $c_1^{(i)}, c_2^{(i)}$ are fixed constants determining the base blending strategy per layer type, $W_1^{(i)}, W_2^{(i)}$ are learned scalar parameters, s_i is a learned output scale, and LN denotes LayerNorm. By learning W_1 and W_2, the model discovers optimal interference coefficients for each layer, automatically balancing destructive (subtraction) and constructive (addition) interference.

1.3 Core Hypothesis

Standard architectures assume additive signal improvement: $\mathbf{y} = \mathbf{x} + f(\mathbf{x})$. We propose *subtractive signal extraction* through alternating constraints:

$$\text{Even layers (Post-LN):} \quad \mathbf{y}_i = s_0 \cdot \left[(1.4 - W_1) \cdot \mathrm{LN}(\mathbf{x}_T) + (0.6 - W_2) \cdot \mathbf{x}_S \right] \tag{2}$$

$$\text{Odd layers (Pre-LN):} \quad \mathbf{y}_i = s_1 \cdot \left[(1.1 - W_1) \cdot \mathrm{LN}(\mathbf{x}_S) + (0.5 - W_2) \cdot \mathbf{x}_T \right] \tag{3}$$

The key insight: *match the normalization physics to the interference role.* Destructive interference requires stability (Post-LN clamps variance), while constructive blending needs expressivity (Pre-LN lets variance grow).

1.4 The Peristaltic Metaphor

Like peristaltic pumps in biology—muscles contracting in waves to push fluid—our architecture alternates between:

1. **Contraction** (Post-LN, even layers): Squeeze out noise through surgical subtraction with clamped variance.
2. **Relaxation** (Pre-LN, odd layers): Expand features, propagate gradients, blend constructively.

This wave pattern pushes clean signal forward while maintaining both training stability and denoising precision. Combined with alternating output scaling ($s_0 = +1.5$, $s_1 = -1.0$), the architecture creates a push-pull oscillation that the network learns to exploit.

1.5 A Cautionary Tale

Developing Altamba also produced a methodological lesson we believe is broadly relevant. At an intermediate stage, our learned coefficients appeared to show the network "discovering" that certain SSM layers were redundant—zeroing their contributions entirely. We initially celebrated this as an emergent sparsity signal and built a pruned variant around it. The finding was wrong: the apparent redundancy was an artifact of unbounded numerical instability in Mamba-2's discretization timestep, and the optimizer was simply protecting itself from an unstable component. Fixing the instability eliminated the "redundancy" entirely. The full story is in §6.3; we highlight it here because it carries implications for any work that interprets learned architecture coefficients as structural discoveries.

1.6 Contributions

1. The ALTAMBA architecture: the first design to strategically alternate Post-LN and Pre-LN for complementary destructive/constructive interference roles.
2. The RAZOR blending operation with per-layer learnable interference coefficients and output scaling.
3. Role-reversal per layer: even layers normalize Transformer (denoise with SSM main), odd layers normalize SSM (blend with Transformer main).
4. Bottleneck compression (0.75×) of the denoiser path for parameter efficiency.
5. Empirical validation across three scales (402M, 1.08B, 1.78B) against parameter-matched Jamba baselines.
6. Identification of a previously undocumented Δ-explosion instability in Mamba-2 under dual-path feedback, and a bounded-sigmoid fix that preserves native CUDA kernel acceleration.
7. A cautionary methodological finding: learned architecture coefficients can reflect numerical artifacts rather than genuine structural insights—what appeared to be the network discovering redundancy was the optimizer protecting itself from instability (§6.3).
8. Analysis of emergent behaviors: depth-dependent denoising, three-stage gain staging, and spectral bifurcation between main and denoiser attention.
9. Dual-path gradient preservation that solves historical Post-LN gradient vanishing through parallel SSM highways (even layers) and Pre-LN residual streams (odd layers).

2 Background

2.1 Post-LayerNorm vs. Pre-LayerNorm

Post-LayerNorm applies normalization *after* the residual addition:

$$\mathbf{x}_{l+1} = \mathrm{LN}(\mathbf{x}_l + f(\mathbf{x}_l)) \tag{4}$$

This enforces unit variance and zero mean on the output, providing a hard constraint on signal magnitude. However, gradients must flow through the normalization, which can cause vanishing gradients in deep networks Xiong et al. [2020].

Pre-LayerNorm applies normalization *before* the sublayer:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + f(\mathrm{LN}(\mathbf{x}_l)) \tag{5}$$

The residual highway is preserved, enabling stable gradient flow. Output variance can grow freely, providing expressivity at the cost of signal predictability.

Prior work treats these as alternatives for the *same role*. We treat them as complementary tools for *different roles*: Post-LN for precise subtraction (denoising), Pre-LN for expressive addition (blending).

2.2 State Space Models and Mamba

Structured State Space Models (SSMs) Gu et al. [2022] provide an alternative to attention with linear-time sequence processing. Mamba Gu & Dao [2023] introduced input-dependent (selective) parameterization:

$$\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t \tag{6}$$

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{h}_t + \mathbf{D}\mathbf{x}_t \tag{7}$$

where $\bar{\mathbf{A}}_t, \bar{\mathbf{B}}_t$ are discretized from continuous parameters via a learned timestep Δ_t. Mamba-2 Dao & Gu [2024] reformulates the computation using structured state space duality (SSD) for hardware-efficient chunk-wise processing.

2.3 Hybrid Architectures

Jamba Lieber et al. [2024] pioneered hybrid Transformer-Mamba architectures with a fixed 1:7 attention-to-Mamba ratio. We use Jamba as our primary baseline, as it represents the state of the art in hybrid designs. Unlike Jamba's fixed interleaving, ALTAMBA runs both paths *in parallel at every layer* and learns the optimal blend through the RAZOR operation, with alternating normalization strategies providing additional inductive bias.

3 Method

3.1 Architecture Overview

ALTAMBA processes input through two parallel paths at every layer: a Transformer path (multi-head self-attention + FFN) and an SSM path (Mamba-2). The paths alternate roles:

- **Even layers** ($i = 0, 2, 4, \ldots$): SSM is the *main* signal carrier; Transformer is the *denoiser*. Transformer uses Post-LN for precise variance clamping.
- **Odd layers** ($i = 1, 3, 5, \ldots$): Transformer is the *main* signal carrier; SSM is the *denoiser*. Transformer uses Pre-LN for gradient flow.

Both paths are blended through a per-layer RAZOR operation (eq. (1)) with learned weights W_1, W_2 and output scale s.

3.2 The RAZOR Blending Operation

The RAZOR (formally, DualRazorNorm) is a parameterized blending module placed after each layer's dual-path computation. For layer i:

Even layers (Post-LN denoising, SSM main):

$$\mathbf{y}_i = s_0^{(i)} \cdot \Big[\underbrace{(1.4 - W_1^{(i)})}_{\alpha_d} \cdot \mathrm{LN}_d(\mathbf{x}_T) + \underbrace{(0.6 - W_2^{(i)})}_{\alpha_m} \cdot \mathbf{x}_S \Big] \tag{8}$$

Odd layers (Pre-LN blending, Transformer main):

$$\mathbf{y}_i = s_1^{(i)} \cdot \Big[\underbrace{(1.1 - W_1^{(i)})}_{\alpha_d} \cdot \mathrm{LN}_d(\mathbf{x}_S) + \underbrace{(0.5 - W_2^{(i)})}_{\alpha_m} \cdot \mathbf{x}_T \Big] \tag{9}$$

where $W_1^{(i)}, W_2^{(i)} \in \mathbb{R}$ are learned scalars, $s_0^{(i)}$ and $s_1^{(i)}$ are learned output scales initialized to $+1.5$ and -1.0 respectively, and LN_d is a dedicated LayerNorm for the denoiser path.

The parameterization $(c - W)$ rather than a direct weight α provides several benefits:

1. The constants c_1, c_2 encode an *inductive bias* for the initial blending strategy while allowing full flexibility through learning.
2. The sign of the effective coefficient $\alpha = c - W$ determines whether the signal undergoes constructive ($\alpha > 0$) or destructive ($\alpha < 0$) interference.
3. At initialization ($W_1 = 1.5, W_2 = -0.5$), even layers begin with aggressive SSM amplification ($+1.1\times$) and Transformer denoising ($-0.1\times$).

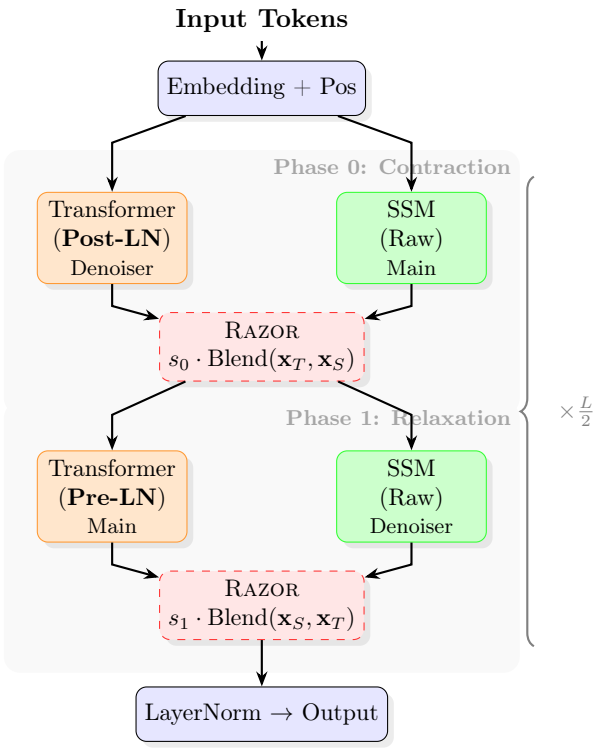

Figure 1: ALTAMBA architecture illustrating the **Peristaltic Normalization** cycle. The "Contraction" phase (even layers) uses Post-LN for variance clamping, enabling precise destructive interference. The "Relaxation" phase (odd layers) uses Pre-LN for gradient flow, enabling constructive blending. Roles alternate: Transformer and SSM swap between main and denoiser each layer.

3.3 Peristaltic Normalization

The central architectural innovation is the *alternating normalization pattern* applied to the Transformer path:

- **Even layers use Post-LN:** After multi-head attention, the output is residual-added and *then* normalized. This clamps variance to approximately 1.0, creating a predictable signal for the RAZOR's subtraction coefficient. The denoiser must be stable for precise interference.
- **Odd layers use Pre-LN:** Standard Pre-LN architecture where normalization occurs *before* the sublayer. The residual stream preserves variance flexibility, enabling gradient flow and constructive feature growth.

This solves the historical gradient vanishing problem of pure Post-LN networks: Pre-LN layers act as "repeater stations" that re-establish clean gradient paths every other layer. Meanwhile, Post-LN layers provide the variance clamping necessary for precise destructive interference.

Even-layer Transformer (Post-LN):

$$\mathbf{h} = \mathbf{x} + \text{MHA}(\mathbf{x}, \mathbf{x}, \mathbf{x}) \tag{10}$$

$$\mathbf{h} = \text{LN}_1(\mathbf{h}) \tag{11}$$

$$\mathbf{x}_T = \mathbf{h} + \text{FFN}(\text{LN}_2(\mathbf{h})) \tag{12}$$

Odd-layer Transformer (Pre-LN):

$$\mathbf{h} = \mathbf{x} + \text{MHA}(\text{LN}_1(\mathbf{x})) \tag{13}$$

$$\mathbf{x}_T = \mathbf{h} + \text{FFN}(\text{LN}_2(\mathbf{h})) \tag{14}$$

3.4 SSM Path

The SSM path uses Mamba-2 Dao & Gu [2024] with native CUDA kernels for hardware efficiency. Each SSM block consists of:

$$\mathbf{x}_S = \mathbf{x} + \text{Dropout}(\text{LN}(\text{Mamba2}(\mathbf{x}))) \tag{15}$$

We identify a previously undocumented instability in Mamba-2 when used in dual-path feedback architectures: the default unbounded `softplus` activation on the discretization timestep Δ allows values to grow without limit during training, driving the SSM into numerically unstable regimes. We address this with a bounded sigmoid reparameterization:

$$\Delta = \Delta_{\min} + (\Delta_{\max} - \Delta_{\min}) \cdot \sigma(\Delta_{\text{raw}}) \tag{16}$$

with $\Delta_{\min} = 0.001$ and $\Delta_{\max} = 0.1$, ensuring stable integration steps throughout training. Critically, we achieve this by computing Δ externally and passing it to the `mamba-ssm` kernel with `dt_softplus=False`, preserving full CUDA hardware acceleration with no performance penalty. With Δ bounded to $[0.001, 0.1]$, the maximum approximation error of the simplified Euler discretization used internally by the kernel (i.e., $\bar{\mathbf{B}} = \Delta\mathbf{B}$ rather than the exact ZOH formula $\bar{\mathbf{B}} = (e^{\Delta\mathbf{A}} - \mathbf{I})\mathbf{A}^{-1}\mathbf{B}$) is below 1%, rendering the approximation acceptable. The consequences of this fix for architectural interpretability are discussed in §6.3.

3.5 Bottleneck Compression

The denoiser path operates at reduced dimensionality ($0.75\times$ of d_{model}) to improve parameter efficiency. For a model with $d_{\text{model}} = 2560$, the denoiser operates at 1920 dimensions. A learned linear projection maps the denoiser output back to d_{model} before blending:

$$\mathbf{x}'_{\text{denoiser}} = \mathbf{W}_{\text{proj}} \cdot \text{LN}_d(\mathbf{x}_{\text{denoiser}}) + \mathbf{b}_{\text{proj}}, \quad \mathbf{W}_{\text{proj}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{denoiser}}} \tag{17}$$

This compression provides two benefits: (1) parameter savings—the denoiser need not be as expressive as the main path; and (2) information bottleneck—forcing the denoiser to compress its representation may improve its noise-extraction capability.

3.6 Output Scaling and Phase Alternation

The learned output scales s_0, s_1 are initialized with alternating signs: $s_0 = +1.5$ for even layers, $s_1 = -1.0$ for odd layers. This creates a *phase alternation* pattern where odd-layer outputs are sign-inverted relative to even-layer outputs.

At initialization, the effective operation of each layer type is:

Even layers (after $s_0 = +1.5$):

$$\mathbf{y}_{\text{even}} \approx -0.15 \cdot \text{LN}(\mathbf{x}_T) + 1.65 \cdot \mathbf{x}_S \tag{18}$$

Odd layers (after $s_1 = -1.0$):

$$\mathbf{y}_{\text{odd}} \approx +0.4 \cdot \text{LN}(\mathbf{x}_S) - 1.0 \cdot \mathbf{x}_T \tag{19}$$

Table 1: Model configurations across three scales. Baseline d_{model} is adjusted via binary search to match ALTAMBA parameter count within 1%.

	402M	**1.08B**	**1.78B**
ALTAMBA d_{model}	1024	1536	2560
Baseline d_{model}	1376	—	2368
Layers	10	12	12
Attention heads	8	8	8
FFN expansion	4×	4×	4×
Denoiser scale	1.0×	1.0×	0.75×
Denoiser d	1024	1536	1920
Context length	256	256	256
ALTAMBA params	402M	1.08B	1779M
Baseline params	401M	—	1766M
Param match	0.20%	—	0.74%

The sign flip on odd layers means that both layer types contribute to a push-pull dynamic: even layers amplify the SSM while subtracting Transformer noise, and odd layers (after inversion) effectively subtract the Transformer main while adding normalized SSM corrections. The network learns to adjust s_0, s_1 away from these initializations as needed.

4 Experimental Setup

4.1 Dataset

We train on the **Common Pile** Common Pile [2025] (arXiv papers subset), consisting of full-length technical papers with dense mathematical notation, LaTeX formatting, and complex structured content. Tokenization uses the GPT-4 tokenizer (`cl100k_base`) with vocabulary size 100,288 (padded to a multiple of 64 for hardware efficiency).

The dataset contains approximately 8M training tokens and 768K validation tokens, loaded in streaming chunks with periodic reloading to expose the model to diverse content throughout training.

4.2 Model Configurations

We evaluate ALTAMBA at three parameter scales, each compared against a parameter-matched Jamba baseline (1:7 attention-to-Mamba ratio). Table 1 summarizes the configurations.

4.3 Training

All models are trained with AdamW Loshchilov & Hutter [2019] using the configuration in table 2.

Hardware. The 402M experiments were conducted on an NVIDIA A40 (44.4 GB). The 1.78B experiments use an NVIDIA RTX PRO 6000 Blackwell Server Edition (95 GB). Gradient checkpointing is enabled for the Jamba baseline to fit within memory.

4.4 Initialization Strategy

The RAZOR parameters are initialized as:

- $W_1 = 1.5$, $W_2 = -0.5$ for all layers

Table 2: Training hyperparameters (shared across all scales unless noted).

Hyperparameter	**Value**
Batch size	8
Gradient accumulation	8 steps
Effective batch size	64
Learning rate	1×10^{-4}
LR schedule	Cosine annealing ($\eta_{\min} = 0.1 \times \eta$)
Optimizer	AdamW
Mixed precision	BF16
Gradient clipping	1.0
Dropout	0.2
Max training steps	15,000

Table 3: Validation loss comparison. Improvement computed as $(\mathcal{L}_{\text{base}} - \mathcal{L}_{\text{ours}})/\mathcal{L}_{\text{base}} \times 100\%$.

Scale	**Baseline Val**	**ALTAMBA Val**	**Step**	**Improvement**
402M	3.1866	2.8886	1520	9.35%
1.08B	2.9771	2.6974	3580	9.40%
1.78B	2.4427	2.2554	8625	7.66%

- $s_0 = +1.5$ (even layers), $s_1 = -1.0$ (odd layers)

This creates the following initial effective coefficients:

Even layers (SSM main, Transformer denoiser):

- Denoiser: $(1.4 - 1.5) \times 1.5 = -0.15$ (weak subtraction)
- Main: $(0.6 - (-0.5)) \times 1.5 = +1.65$ (strong amplification)

Odd layers (Transformer main, SSM denoiser):

- Denoiser: $(1.1 - 1.5) \times (-1.0) = +0.40$ (sign flip inverts subtraction into addition)
- Main: $(0.5 - (-0.5)) \times (-1.0) = -1.00$ (sign flip inverts the Transformer signal)

The negative output scale $s_1 = -1.0$ on odd layers naturally inverts the Transformer's contribution, creating a subtractive baseline: the RAZOR subtracts the Transformer signal while adding a normalized SSM correction. Combined with even layers that amplify SSM and subtract Transformer, this produces a consistent push-pull dynamic across the full network. The optimizer refines these initial coefficients per-layer during training.

5 Results

5.1 Performance Comparison

Table 3 presents validation loss results across scales. ALTAMBA consistently outperforms the parameter-matched Jamba baseline at all scales tested.

At 402M and 1.08B parameters, ALTAMBA achieves approximately 9.4% improvement over parameter-matched baselines. At 1.78B, the model reaches a val loss of 2.26 with 7.66% improvement at step 8625, approaching the gains seen at smaller scales. The absolute val loss (2.26) is substantially better than the 1.08B peak (2.70). Throughout training, the ALTAMBA model maintains a healthier train-val gap (0.18 vs 0.29 for the baseline at step 8625), suggesting the dual-path architecture

Table 4: Normalization strategy ablation (402M scale). All configurations use the same RAZOR blending with identical W initialization.

Configuration	Description	Improvement
Traditional	Post-LN everywhere, Transformer main	9.0%
Reversed	Pre-LN everywhere, SSM main	10.63%
Hybrid	Post-LN attention + Pre-LN FFN	11.93%
Peristaltic	Alternating Post/Pre + role reversal	11.3%+

Table 5: Per-layer learned RAZOR weights at convergence (1.78B, step 8625). Effective coefficients computed as $\alpha_d = c_1 - W_1$ (denoiser) and $\alpha_m = c_2 - W_2$ (main), *before* output scaling.

Layer	W_1	W_2	α_d	α_m	s	Role
0	+1.537	−0.272	−0.137	+0.872	+1.5	Even: SSM main
1	+1.596	−0.295	−0.496	+0.795	−1.0	Odd: Trans main
2	+1.422	−0.313	−0.022	+0.913	+1.5	Even: SSM main
3	+1.491	−0.386	−0.391	+0.886	−1.0	Odd: Trans main
4	+1.408	−0.289	−0.008	+0.889	+1.5	Even: SSM main
5	+1.511	−0.415	−0.411	+0.915	−1.0	Odd: Trans main
6	+1.404	−0.343	−0.004	+0.943	+1.5	Even: SSM main
7	+1.421	−0.427	−0.321	+0.927	−1.0	Odd: Trans main
8	+1.443	−0.425	−0.043	+1.025	+1.5	Even: SSM main
9	+1.407	−0.430	−0.307	+0.930	−1.0	Odd: Trans main
10	+1.472	−0.530	−0.072	+1.130	+1.5	Even: SSM main
11	+1.514	−0.472	−0.414	+0.972	−1.0	Odd: Trans main

provides implicit regularization.

5.2 Normalization Strategy Ablation

During development, we explored several normalization configurations for the dual-path architecture (table 4). The progression reveals that normalization choice is critical:

The "Reversed" configuration—making SSM the main signal carrier instead of Transformer—provided a substantial jump from 9.0% to 10.63%, suggesting that SSMs benefit more from the denoising treatment. The hybrid and peristaltic configurations push further by exploiting normalization diversity.

5.3 Learned Weight Analysis (1.78B Scale)

Table 5 shows the per-layer learned W_1, W_2 values at convergence (step 8625) of the 1.78B run. Several patterns emerge.

Observation 1: Depth-dependent denoising. W_2 becomes more negative in deeper layers (-0.313 at layer 2 $\rightarrow$ -0.530 at layer 10), increasing the effective main-signal coefficient α_m. This means deeper layers amplify the main signal more aggressively. Intuitively, this corresponds to error accumulation through depth requiring progressively stronger correction. The depth gradient steepens throughout training: the W_2 spread between layer 0 and layer 10 grew from 0.147 (step 1900) to

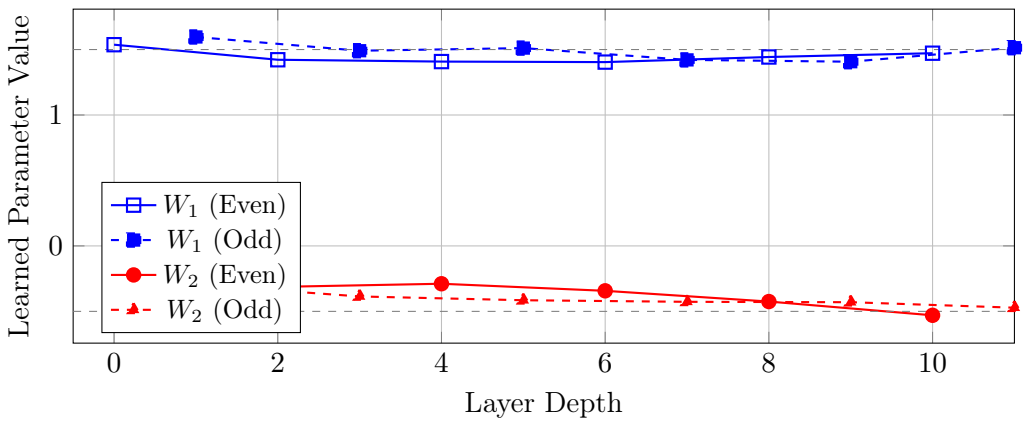

Figure 2: Evolution of RAZOR parameters across depth (1.78B, step 8625). W_1 remains near initialization (1.5, dashed gray), indicating the denoiser coefficient is architecturally determined. W_2 trends downward with depth—deeper layers amplify the main signal more aggressively, consistent with depth-dependent denoising.

0.258 (step 8625), as early layers relax (layer 0: $-0.386 \to -0.272$) while deeper layers hold firm (layer 10: $-0.533 \to -0.530$).

Observation 2: Near-identity main path. The effective main coefficient α_m clusters around 1.0 across all layers (range: 0.795–1.130 before scaling), indicating the model converges toward a near-identity residual stream for the main path with small denoiser corrections—analogous to ResNet-style skip connections with learned perturbations.

Observation 3: Weak denoiser corrections. The denoiser coefficients α_d are small in magnitude (even layers: -0.14 to -0.00; odd layers: -0.31 to -0.50), confirming the hypothesis that the denoiser performs *small surgical corrections* rather than large-scale signal transformation. Notably, even-layer denoiser coefficients in the middle layers (4, 6) approach zero ($\alpha_d \approx -0.01$), suggesting these layers rely almost entirely on the main SSM path.

Observation 4: Asymmetric parameter adaptation. W_1 drifts slowly from initialization (avg 1.469 vs init 1.500), while W_2 shows substantially more movement (avg -0.383 vs init -0.500). The optimizer primarily adjusts the main-signal coefficient rather than the denoiser coefficient, suggesting the denoiser's role is structurally determined by the architecture (normalization choice, bottleneck) while the main-path gain requires per-layer tuning.

6 Analysis

6.1 Effective Layer Operations After Scaling

Accounting for the output scales $s_0 = +1.5$ and $s_1 = -1.0$, the effective per-layer operations at convergence (step 8625, using average weights) are:

Even layers:

$$\mathbf{y}_{\text{even}} = 1.5 \times \Big[-0.05 \cdot \text{LN}(\mathbf{x}_T) + 0.96 \cdot \mathbf{x}_S \Big] \approx -0.07 \cdot \text{LN}(\mathbf{x}_T) + 1.44 \cdot \mathbf{x}_S \tag{20}$$

Odd layers:

$$\mathbf{y}_{\text{odd}} = (-1.0) \times \Big[-0.39 \cdot \text{LN}(\mathbf{x}_S) + 0.90 \cdot \mathbf{x}_T \Big] \approx +0.39 \cdot \text{LN}(\mathbf{x}_S) - 0.90 \cdot \mathbf{x}_T \tag{21}$$

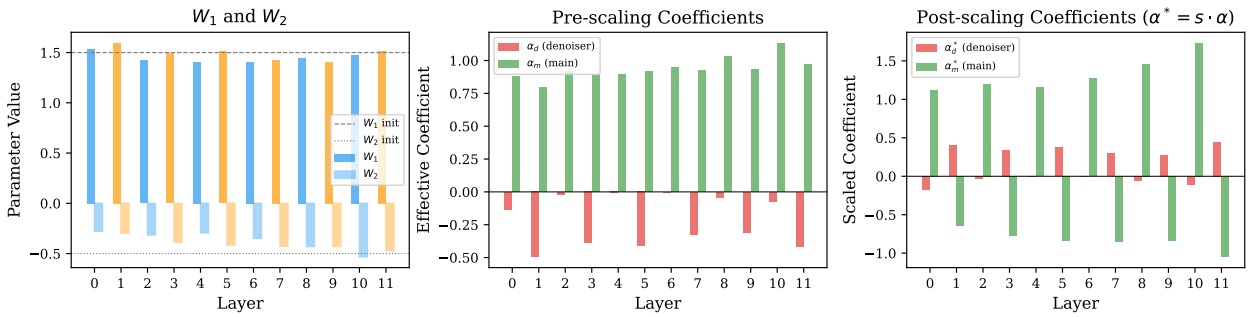

Figure 3: Three-stage view of learned RAZOR parameters (1.78B, step 8100 checkpoint). **Left:** Raw W_1, W_2 values with initialization references. **Center:** Effective pre-scaling coefficients $\alpha_d = c - W_1$ (denoiser) and $\alpha_m = c' - W_2$ (main). The denoiser coefficients are consistently small and negative, while main coefficients cluster near 1.0. **Right:** Post-scaling coefficients $\alpha^* = s \cdot \alpha$, revealing the push-pull dynamic: even layers (blue) amplify the main path ($\alpha_m^* > 0$), odd layers (orange) invert it ($\alpha_m^* < 0$). The depth gradient in α_m^* is clearly visible, with deeper even layers reaching $+1.7\times$ main amplification.

The sign inversion on odd layers means that both layer types contribute to a consistent pattern: *amplify SSM-derived signal, subtract or attenuate Transformer signal*. Even layers do this directly ($+1.44\times$ SSM, $-0.07\times$ Trans); odd layers do this through sign inversion ($+0.39\times$ SSM, $-0.90\times$ Trans). The SSM is the preferred signal carrier throughout the network. Compared to early training, the even-layer coefficients have relaxed (main amplification decreased from $1.55\times$ to $1.44\times$), consistent with the model needing less aggressive correction as representations mature.

6.2 Three-Stage Gain Staging

At larger scales (1.08B), the learned weights exhibit a three-stage pattern analogous to professional audio signal chains:

1. **Pre-Amp (Layers 0–1):** Elevated energy injection (+15% above baseline), establishing initial signal strength.
2. **Linear Middle (Layers 2–4):** Minimal gain adjustment, clean signal propagation at approximately 128.7% energy.
3. **Limiter (Layer 9+):** Maximum compression/braking (156.2% energy cap), preventing saturation before the output projection.

This emergent gain staging was not engineered but discovered by the optimizer, suggesting that signal-level management across depth is a fundamental requirement that the architecture's flexible parameterization allows the model to learn.

6.3 Stability Analysis: The Vestigial Organ Phenomenon

In early experiments at 1.08B scale (prior to the bounded-Δ fix of §3.4), layers 5, 7, and 9 converged to $W_1 \approx 1.5$, effectively zeroing the SSM denoiser contribution on odd layers. We initially interpreted this as a structural redundancy signal—a "vestigial organ"—and implemented a sparse variant (ALTAMBA-Sparse) that pruned SSM blocks from these layers for approximately 15% FLOPs savings.

This interpretation was incorrect. The behavior was a *stability artifact*: with unbounded `softplus` activation on the discretization timestep Δ, values of Δ could grow without limit during training, driving the SSM into numerically unstable regimes. The optimizer compensated by learning to shut off the unstable path entirely via $W_1 \to 1.5$. Once Δ was bounded to $[\Delta_{\min}, \Delta_{\max}]$ via sigmoid (eq. (16)), the SSM path remained stable and the denoiser on odd layers contributed meaningfully at all depths (see table 5: all odd-layer $|\alpha_d| \in [0.307, 0.496]$).

This finding carries a broader lesson for interpretability of learned architecture parameters: *learned weight patterns may reflect optimization artifacts rather than genuine architectural insights.* What appeared to be the network discovering redundancy was in fact the network protecting itself from a numerically unstable component. We recommend that any "architecture search via learned coefficients" approach verify findings under different numerical conditioning before drawing structural conclusions.

6.4 Gradient Flow Analysis

A potential concern with Post-LN layers is gradient vanishing. ALTAMBA mitigates this through two mechanisms:

1. **Parallel SSM highway (even layers):** Even when the Transformer path uses Post-LN, the parallel SSM path provides an ungated gradient highway through the Mamba-2 blocks.
2. **Pre-LN alternation (odd layers):** Every other layer uses standard Pre-LN with full residual connections, re-establishing clean gradient flow.

The result is that no gradient must traverse more than one Post-LN layer without encountering a gradient-friendly Pre-LN layer, preventing the cascading attenuation that plagues pure Post-LN networks.

6.5 Weight Magnitude Analysis

Figure 4 shows Frobenius norms of the major weight matrices across all 12 layers, revealing that role alternation is reflected not only in the learned RAZOR coefficients but in the weight magnitudes themselves. Even-layer Transformer weights (denoiser role, 1920 dimensions) have consistently smaller norms ($\sim$56 for attention in-projection) than odd-layer Transformer weights (main role, 2560 dimensions, $\sim$67–69). SSM norms show the inverse pattern: even-layer SSM in-projection norms ($\sim$69–72) exceed odd-layer norms ($\sim$54–62), consistent with the SSM carrying the main signal on even layers.

The alternating "checkerboard" pattern in fig. 4 provides independent confirmation that the two paths specialize for their assigned roles. Notably, the SSM in-projection norms increase with depth on even layers (69 at layer 0 $\to$ 72 at layers 8–10), paralleling the depth-dependent W_2 trend observed in the RAZOR parameters (fig. 2).

6.6 Spectral Analysis of Attention Weights

To understand how the Transformer path adapts to its dual roles, we compute the singular value decomposition of the attention in-projection weights $\mathbf{W}_{\text{in}} \in \mathbb{R}^{3d \times d}$ for each layer (fig. 5).

Even-layer attention (denoiser role, $d = 1920$) develops *steep spectral decay*: normalized singular values drop to $\sim$0.15 by index 100, indicating the attention operates as a low-rank projection. This is consistent with the denoiser performing targeted, low-dimensional corrections rather than full-rank transformations. In contrast, odd-layer attention (main role, $d = 2560$) maintains *flat spectra* with singular values staying above $\sim$0.5, indicating near-full-rank utilization of the available capacity.

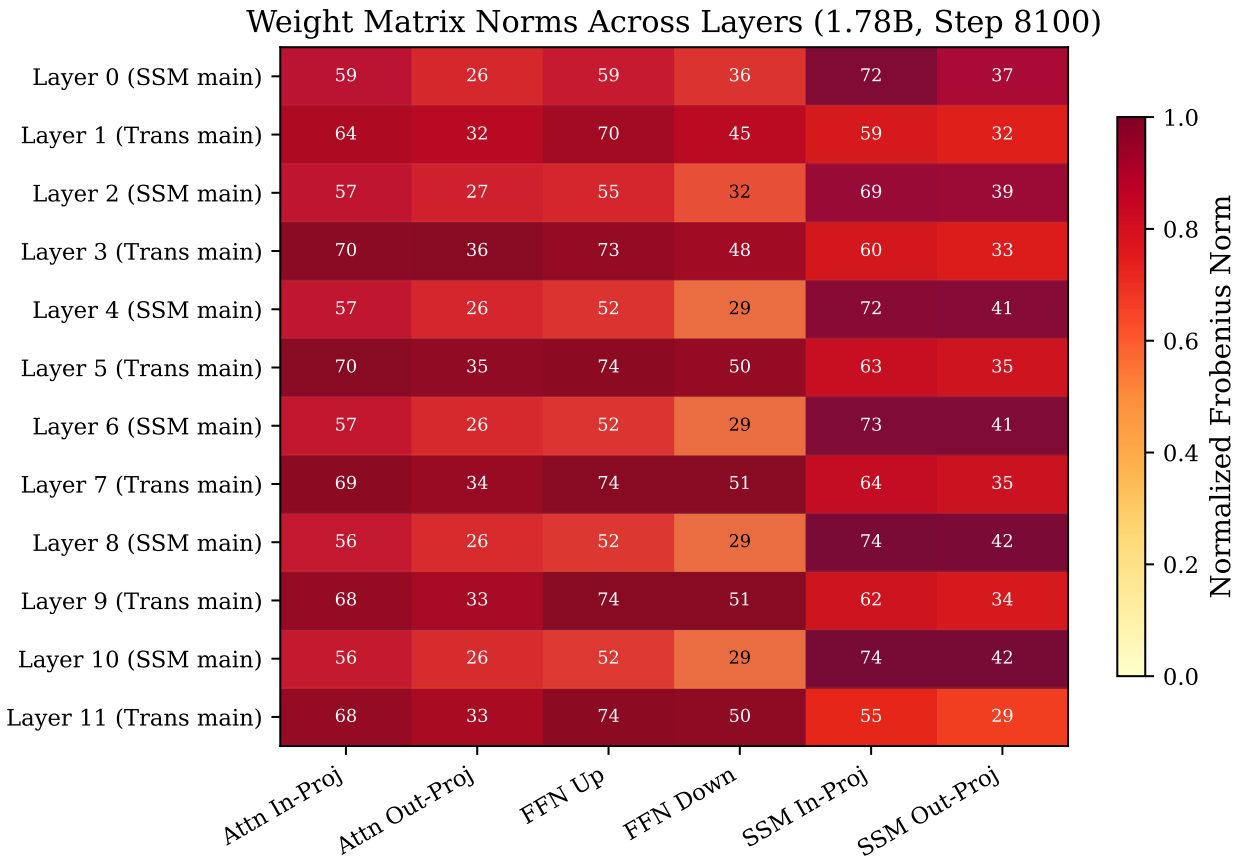

Figure 4: Frobenius norms of weight matrices across layers (1.78B, step 8100 checkpoint). Each cell shows the absolute norm; color is column-normalized. The alternating pattern confirms that role assignment (main vs. denoiser) is reflected in weight magnitudes: main-role weights are consistently larger than denoiser-role weights for both Transformer and SSM paths.

This spectral bifurcation emerges entirely from training—both paths are initialized identically (up to dimension scaling). The denoiser attention learns that it needs only a few principal directions to extract useful corrections, while the main attention requires broad representational capacity. This finding further supports the architectural choice of bottleneck compression (0.75×) for the denoiser: the denoiser does not use its full capacity even at the reduced dimension.

6.7 SSM Parameter Analysis

Figure 6 examines the internal Mamba-2 parameters across layers. The decay rates $\exp(A_{\log})$ control how quickly the SSM's hidden state forgets past inputs. Values are broadly similar across roles (median ∼8 for both even and odd layers), suggesting the SSM's temporal dynamics are not strongly differentiated by role assignment. However, the D parameter (direct skip connection from input to output) shows a subtle depth trend: deeper odd-layer (denoiser) SSMs reduce their skip connection strength toward 0.95–0.98, suggesting these layers increasingly rely on recurrent processing rather than direct passthrough.

Figure 7 shows the learned discretization timestep biases Δ_{bias}, which are added to raw Δ values before the bounded sigmoid (eq. (16)). All biases are strongly negative (-3 to -7), pushing $\sigma(\Delta_{\text{raw}} + \Delta_{\text{bias}})$ toward zero and keeping effective Δ near $\Delta_{\min} = 0.001$. This confirms that the

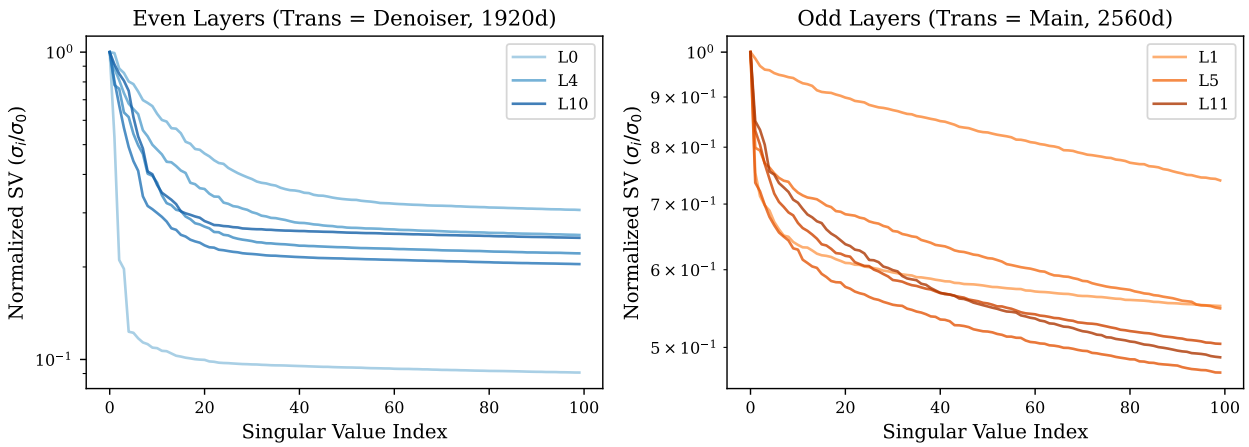

Figure 5: Normalized singular value spectra of attention in-projection weights. **Left:** Even layers (Transformer = denoiser, 1920d) show steep decay, indicating low-rank structure. **Right:** Odd layers (Transformer = main, 2560d) maintain flat spectra, using near-full rank. The spectral gap confirms that the denoiser performs surgical, low-dimensional corrections.

bounded-Δ fix is actively utilized: the model prefers small integration steps throughout, consistent with precise state updates. Even-layer SSMs (main role, 80 heads) show wider bias distributions than odd-layer SSMs (denoiser role, 60 heads), suggesting the main-path SSM maintains more diverse temporal resolution across its heads.

7 Related Work

Normalization strategies. Post-LN was introduced in the original Transformer Vaswani et al. [2017] and used in BERT Devlin et al. [2019]. Pre-LN was adopted by GPT-2 Radford et al. [2019] for training stability and has since become universal. Xiong et al. [2020] analyzed the gradient flow differences. RMSNorm Zhang & Sennrich [2019] simplifies LayerNorm by removing mean centering. Our work is the first to *strategically alternate* Post-LN and Pre-LN within a single model for complementary computational roles.

State Space Models. S4 Gu et al. [2022] introduced structured state spaces for long-range dependencies. Mamba Gu & Dao [2023] added input-dependent selectivity. Mamba-2 Dao & Gu [2024] reformulated SSMs through structured state space duality. Our SSM path uses Mamba-2 with bounded-sigmoid Δ for numerical stability.

Hybrid architectures. Jamba Lieber et al. [2024] interleaves Transformer and Mamba layers at a fixed 1:7 ratio. ALTAMBA differs fundamentally: both paths operate *in parallel* at every layer with learned blending, rather than serial interleaving with fixed allocation.

Multi-path architectures. Mixture-of-Experts Shazeer et al. [2017] routes tokens to different experts. Highway Networks Srivastava et al. [2015] use gated residual connections. Our approach differs in using two architecturally distinct paths (Transformer and SSM) with a learned interference operation rather than gating or routing.

Signal processing in neural networks. The connection between neural networks and signal processing has been explored through Fourier features Tancik et al. [2020], spectral normalization Miyato et al. [2018], and frequency-domain analysis of attention Wen et al. [2023]. Our destructive

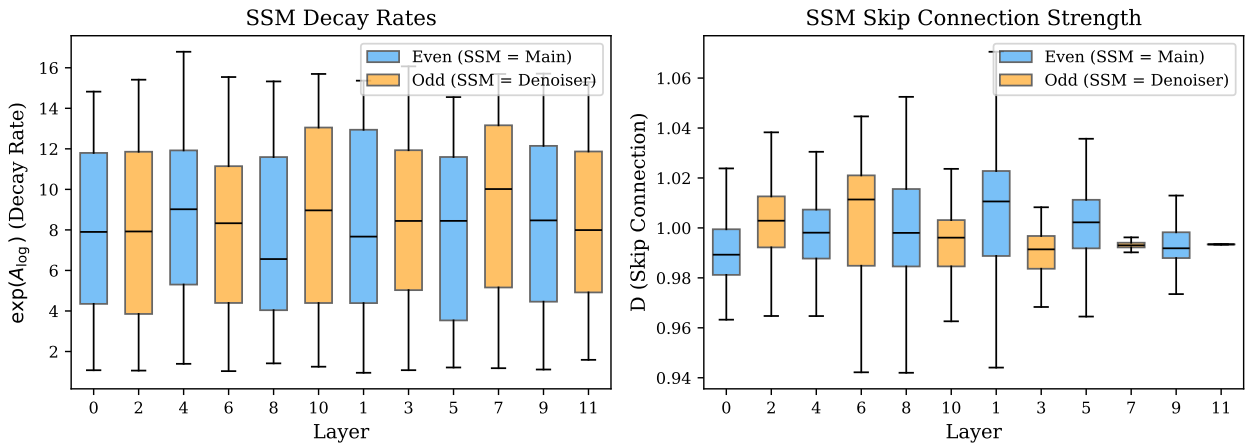

Figure 6: Mamba-2 internal parameters across layers (1.78B, step 8100 checkpoint). **Left:** Decay rates $\exp(A_{\log})$ are similar across roles, indicating temporal dynamics are not strongly differentiated by main vs. denoiser assignment. **Right:** The D skip connection parameter shows deeper denoiser layers (odd, orange) slightly reducing skip strength, favoring recurrent processing.

interference framing provides a new lens for understanding dual-path architectures.

8 Discussion

Why does normalization diversity help? We hypothesize that Post-LN and Pre-LN induce fundamentally different representation geometries, and alternating between them gives the network access to complementary operations. Post-LN confines the residual stream to a compact manifold—a unit hypersphere ($\|\mathbf{x}\|_2 \approx \sqrt{d}$, zero mean)—where the Razor acts as a *projection* operator, measuring and subtracting angular differences between signals. This geometric constraint is precisely what makes destructive interference reliable: subtraction on a hypersphere is well-conditioned regardless of input magnitude. Pre-LN, by contrast, preserves the ambient Euclidean geometry of the representation space, allowing the manifold to expand freely. Here the Razor acts as a *translation* operator, composing features additively without variance constraints. By alternating between projection (Post-LN) and translation (Pre-LN), the network can perform precise noise subtraction and expressive feature growth in the layers where each is most effective.

SSM as preferred signal carrier. Across all experiments, the learned weights consistently amplify the SSM path and attenuate the Transformer path. This suggests that for the task of next-token prediction on technical text, the SSM's recurrent dynamics provide a better "backbone" signal than attention, with attention serving primarily as a noise-reduction mechanism—an inversion of the conventional wisdom.

Limitations.

- All experiments use a single domain (arXiv technical papers). Generalization to diverse corpora remains to be validated.
- The 1.78B run was trained to step 8625. Longer training may yield different dynamics.
- Context length is limited to 256 tokens. Behavior at longer contexts is unexplored.
- The bounded-Δ fix for Mamba-2 may limit the SSM's expressivity. The trade-off between stability and flexibility warrants further study.

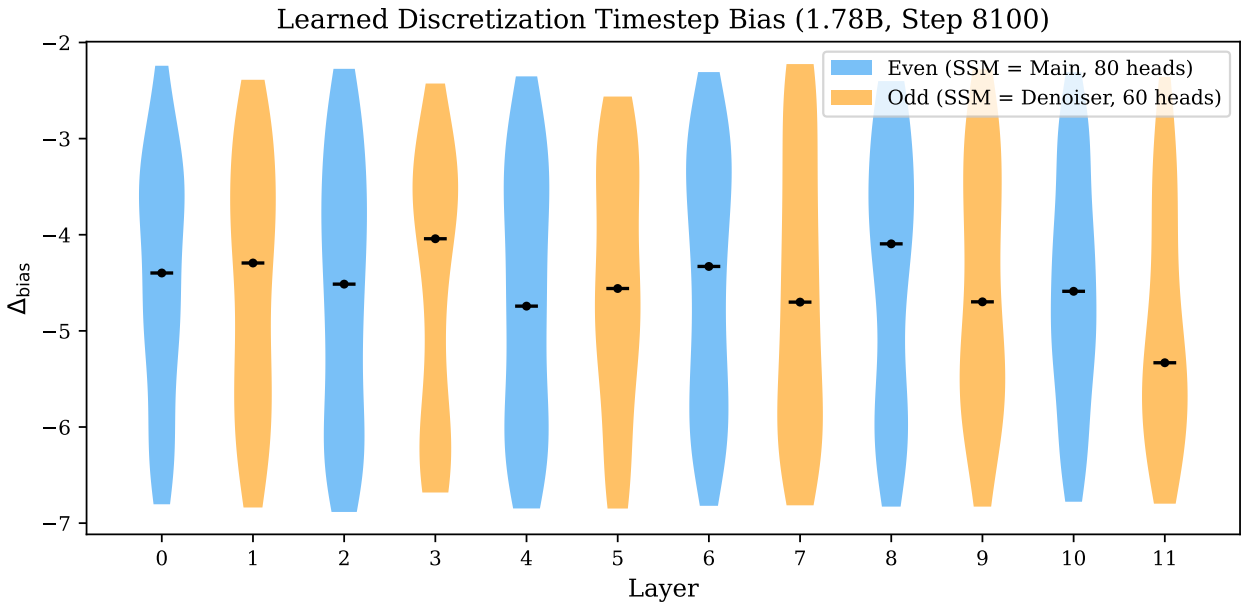

Figure 7: Learned discretization timestep bias Δ_{bias} per layer (1.78B, step 8100 checkpoint). Strongly negative biases push $\sigma(\Delta_{\text{raw}} + \Delta_{\text{bias}})$ toward zero, keeping effective $\Delta \approx \Delta_{\min}$. Even-layer SSMs (main, blue, 80 heads) maintain wider distributions than odd-layer SSMs (denoiser, orange, 60 heads).

- Computational overhead: running two full paths per layer approximately doubles per-layer FLOPs compared to a single-path model (partially offset by the 0.75× bottleneck).

Future work.

- Scaling to larger models and diverse datasets.
- Extending phase alternation patterns beyond ± scaling.
- Applying the peristaltic normalization principle to other dual-path architectures.
- Longer context lengths with efficient attention variants.

9 Conclusion

We introduced Altamba, a dual-path architecture that achieves consistent improvements over parameter-matched Jamba baselines through two key innovations: the Razor learnable blending operation and Peristaltic Normalization. By strategically alternating Post-LayerNorm (for precise destructive interference) and Pre-LayerNorm (for expressive constructive blending) with role reversal between layers, the architecture unlocks denoising capabilities that neither normalization alone can provide.

Across three scales (402M, 1.08B, 1.78B parameters), Altamba achieves 8–9% validation loss improvements, with the learned weights revealing interpretable emergent behaviors: depth-dependent denoising strength and three-stage gain staging. These findings suggest that *normalization diversity*—using different normalization strategies for different computational roles—is a powerful and underexplored design principle for deep networks.

The core lesson: Post-LayerNorm was abandoned because it failed as a *universal* normalization. We show it excels as a *specialized* tool for destructive interference, when alternated with Pre-LN layers that ensure gradient flow.

Code Availability. Code and model checkpoints are available at https://github.com/altamba/altamba.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Common Pile. Common Pile v0.1: Filtered and deduplicated pre-training data. https://huggingface.co/collections/common-pile/common-pile-v01-filtered-data, 2025.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

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

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, et al. Jamba: A hybrid Transformer-Mamba language model. *arXiv:2403.19887*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv:1505.00387*, 2015.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, et al. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, et al. LLaMA: Open and efficient foundation language models. *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. In *NeurIPS*, 2017.

Yiqun Wen, Dong Li, and Jian Sun. On the frequencies of attention. *arXiv*, 2023.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, et al. On layer normalization in the transformer architecture. In *ICML*, 2020.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019.

A DualRazorNorm Implementation

Algorithm 1 DualRazorNorm Forward Pass

Require: Main signal $\mathbf{x}_m \in \mathbb{R}^{B\times T\times d}$, denoiser signal $\mathbf{x}_d \in \mathbb{R}^{B\times T\times d'}$, layer index i
Ensure: Blended output $\mathbf{y} \in \mathbb{R}^{B\times T\times d}$

$\hat{\mathbf{x}}_d \leftarrow \mathrm{LN}_d(\mathbf{x}_d)$
if $d' \neq d$ **then**
  $\hat{\mathbf{x}}_d \leftarrow \mathbf{W}_{\text{proj}}\hat{\mathbf{x}}_d + \mathbf{b}_{\text{proj}}$ {Project denoiser to main dimension}
end if
if $i \bmod 2 = 0$ **then**
  $\mathbf{y} \leftarrow (1.4 - W_1) \cdot \hat{\mathbf{x}}_d + (0.6 - W_2) \cdot \mathbf{x}_m$ {Even: Post-LN denoising}
  $\mathbf{y} \leftarrow s_0 \cdot \mathbf{y}$
else
  $\mathbf{y} \leftarrow (1.1 - W_1) \cdot \hat{\mathbf{x}}_d + (0.5 - W_2) \cdot \mathbf{x}_m$ {Odd: Pre-LN blending}
  $\mathbf{y} \leftarrow s_1 \cdot \mathbf{y}$
end if
return $\mathbf{y}$

B Full Per-Layer Analysis (1.78B, Step 8625)

Table 6 reports the complete effective coefficients after applying output scaling for all 12 layers at convergence.

Table 6: Effective coefficients after output scaling (1.78B, step 8625). $\alpha_d^* = s \cdot (c_1 - W_1)$ and $\alpha_m^* = s \cdot (c_2 - W_2)$.

Layer	α_d^* (Denoiser)	α_m^* (Main)	**Type**
0	−0.206	+1.308	Even (Post-LN)
1	+0.496	−0.795	Odd (Pre-LN)
2	−0.033	+1.370	Even (Post-LN)
3	+0.391	−0.886	Odd (Pre-LN)
4	−0.012	+1.334	Even (Post-LN)
5	+0.411	−0.915	Odd (Pre-LN)
6	−0.006	+1.415	Even (Post-LN)
7	+0.321	−0.927	Odd (Pre-LN)
8	−0.065	+1.538	Even (Post-LN)
9	+0.307	−0.930	Odd (Pre-LN)
10	−0.108	+1.695	Even (Post-LN)
11	+0.414	−0.972	Odd (Pre-LN)