UNDERSTANDING AND ACCELERATING LONG-CONTEXT EXTRAPOLATION IN HYBRID MODELS VIA STATE-EQUILIBRATION CURRICULUM

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context adaptation of hybrid state-space model (SSM)–Transformer language models remains poorly understood: while multi-stage context-length training is effective in practice, it is unclear how long a model should remain at each intermediate context length before advancing to a longer sequence length. We find that long-context adaptation exhibits a two-phase behavior. When first exposed to a longer context, recurrent state dynamics rapidly stabilize: distance-dependent state drift sharply decreases and then plateaus, which we call state equilibration. Continued training beyond equilibration can still improve performance at the current context length, but these gains increasingly specialize to that length and transfer poorly to longer sequences. Based on this observation, we introduce State-Equilibration Curriculum (SEC), which advances to the next context length once recurrent-state dynamics stabilize. To support SEC efficiently, we develop a context-parallel implementation for hybrid SSM–Transformer models that removes serial cross-device state propagation while exposing chunk-boundary states as a low-overhead training signal. On Nemotron-H-8B and Bamba-9B-v2, SEC matches or improves fully trained multi-stage schedules while using about 30% fewer training tokens and reducing wall-clock training time by up to 42%. Our results suggest that intermediate context stages need not be trained to task convergence: advancing once state dynamics equilibrate provides a more efficient mechanism for multi-stage long-context adaptation.

1 INTRODUCTION

Hybrid SSM–Transformer architectures combine recurrent state-space layers with selected full-attention layers to reduce the cost of long-context language modeling (NVIDIA et al., 2025b; Ganti et al.). By compressing part of the history into a recurrent state, these models can substantially reduce the cost of processing long sequences while retaining the expressive capacity of attention. Despite showing promising results, the native context lengths of hybrid SSM-Transformer models remain limited. Existing methods extending pretrained hybrid models beyond their native context broadly follow two routes. Training-free methods modify positional or recurrent dynamics at inference, which has so far shown limited effectiveness at very long context lengths, such as 100K tokens and beyond (Ben-Kish et al., 2025; Shen et al., 2026; Ye et al., 2025). In contrast, continued-training methods explicitly adapt the model to longer sequences and can support substantially larger context extensions, at the cost of additional training (NVIDIA et al., 2025b).

A common continued-training strategy is *multi-stage long-context training*, where the model is exposed to a sequence of increasingly longer contexts before reaching the final target length (Gao et al., 2025; Xiong et al., 2024; Ding et al., 2024; Xu et al., 2025; NVIDIA et al., 2025b). Such staging can improve training stability and final long-context performance while avoiding the cost of training the entire adaptation budget at the maximum sequence length. However, it introduces a largely unanswered question: *when should an intermediate stage end?* Existing pipelines typically allocate a fixed number of steps or tokens to each stage, or train until loss or downstream performance converges (NVIDIA et al., 2025b; Ganti et al.). Yet without understanding what an intermediate stage must accomplish, there is no principled basis for deciding how much training is sufficient. In

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.

particular, continued improvement at the current context length need to imply that additional training remains useful for adaptation to still longer contexts.

In this work, we show that long-context adaptation exhibits a distinct *two-phase* behavior. In the first phase, which we call *state equilibration*, recurrent-state dynamics rapidly stabilize after the model is exposed to a longer context. In the second phase, *context-length specialization*, continued training still improves performance at the current context length, but these additional gains increasingly specialize to that length and transfer poorly to longer sequences. This two-phase behavior suggests a different view of intermediate stages in multi-stage long-context training: instead of fully optimizing for the current context length, an intermediate stage may only need to provide the adaptation necessary for further context scaling. Once recurrent state dynamics have equilibrated, further optimization at the intermediate length may provide little benefit for subsequent context scaling.

Based on this observation, we introduce **State-Equilibration Curriculum (SEC)**, which advances to the next context length once recurrent-state dynamics enter a stable regime, rather than waiting for task or loss convergence at the current stage. One major challenge is that optimized SSM implementations (Dao & Gu, 2024) do not normally expose the recurrent states needed to monitor this process. Long-context training also requires distributing sequence computation across devices (Liu et al., 2023). We therefore develop an efficient context-parallel implementation for hybrid SSM–Transformer models that addresses both problems at once. The key observation is that the Mamba2 recurrence is affine in the incoming recurrent state: each sequence chunk can be summarized by a locally computable transfer function, while only its incoming state depends on preceding chunks. This allows ranks to compute local chunk summaries in parallel, exchange them collectively, reconstruct the correct incoming states, and then complete the local recurrence. Compared with serial cross-rank state passing, this removes the long inter-device dependency chain while naturally materializing chunk-boundary states that SEC can monitor with little additional overhead.

Our contributions are: (i) We identify a two-phase behavior in long-context adaptation: rapid *state equilibration* followed by slower *context-length specialization*, with gains in the latter stage transferring poorly to longer contexts. (ii) We propose State-Equilibration Curriculum (SEC), which advances context length based on stabilization of recurrent-state dynamics. (iii) We develop an efficient context-parallel implementation for hybrid SSM–Transformer models that both removes serial state propagation and exposes the signals used by SEC. (iv) We evaluate SEC on Nemotron-H-8B and Bamba-9B-v2, extending their native contexts to 128K. Across both models, recurrent-state dynamics stabilize far earlier than current-length task performance. Training beyond this point continues to improve the intermediate context length but provides little additional benefit to longer contexts. SEC therefore truncates the intermediate stage aggressively while preserving, and in some settings improving, final long-context performance. Compared with fully trained multi-stage schedules, SEC uses approximately 30% fewer training tokens and reduces wall-clock training time by up to 42%. Additional evaluations on InfiniteBench and short-context language modeling and reasoning tasks show that these savings do not come at the expense of broader model capabilities.

2 Background and Related Work

Hybrid SSM–Transformer models. Recent language models increasingly combine standard self-attention with state-space or linear-attention layers (Katharopoulos et al., 2020; Vyas et al., 2020), such as Mamba (Gu & Dao, 2023; Dao & Gu, 2024) and Gated DeltaNet (Yang et al., 2025; Hatamizadeh et al., 2026), to reduce the cost of processing long sequences while retaining the modeling capacity of attention (Ganti et al.; NVIDIA et al., 2025b; Qwen Team, 2025; Gu et al., 2025; Lenz et al., 2025; 2024; Ren et al., 2024; Dong et al., 2025). Unlike self-attention, these recurrent layers compress prior context into a fixed-size hidden state that evolves throughout the sequence. This design enables more efficient long-sequence processing, but also makes context-length generalization depend on recurrent dynamics learned within the model's training horizon. Empirically, hybrid and recurrent models can degrade substantially when evaluated beyond their native context length (Waleffe et al., 2024; Jelassi et al., 2024; Shen et al., 2026). This motivates us to examine how its dynamics change during long-context adaptation.

Extending context length. One line of work extends context without additional long-context training. For Transformer models, representative approaches rescale positional information or modify positional encoding, including PI (Chen et al., 2023), NTK-based methods (LocalLLaMA, 2023a;b),

YaRN (Peng et al., 2024), and LongRoPE (Ding et al., 2024; Shang et al., 2025). For stateful models, recent methods such as LongMamba (Ye et al., 2025), DeciMamba (Ben-Kish et al., 2025), and UPI (Shen et al., 2026) instead modify effective receptive-field or state dynamics to improve length extrapolation. Such methods are inexpensive, but their effectiveness degrades as the target length moves substantially beyond the training context. An alternative is to directly adapt the pretrained model on longer sequences through continued training (Ding et al., 2024; Shang et al., 2025; Meta AI, 2024; 2025). Recent hybrid models such as Nemotron-H use multi-stage continued training for long-context adaptation, combining supervised fine-tuning with later preference optimization stages (NVIDIA et al., 2025b;a; Sun et al., 2025).

Context parallelism. Long-context training requires partitioning the sequence dimension across devices to reduce per-device activation memory. For Transformer layers, context-parallel methods such as DeepSpeed-Ulysses (Jacobs et al., 2024) and Ring-Attention (Liu et al., 2023) can be used. For recurrent and linear-attention layers, parallel scan and chunked linear attention provide closely related single-device algorithms for parallelizing computation over sequence chunks, while a straightforward distributed Mamba2 implementation passes the final state of one chunk to the next (Dao & Gu, 2024), preserving a serial dependency across ranks. § 4.2 extends this lineage to distributed context parallelism by exchanging the required cross-chunk states with a single all-gather, removing the rank-wise dependency while exposing the chunk-boundary states used by our SEC method.

3 Two-Phase Long-Context Adaptation

We begin by examining what changes inside a hybrid model when it is adapted from its native context length to a substantially longer one. Our goal is to understand which aspects of the model adapt early and which continue to change with prolonged training.

3.1 Measuring Recurrent-Stage Dynamics

Mamba compresses past context into a fixed-size recurrent state whose evolution is controlled by input-dependent forget gates. As the recurrence is applied over increasingly long sequences, a natural question is how the geometry of this hidden state changes with distance. In particular, states that are close together in the sequence may behave differently from states separated by tens of thousands of tokens, especially when the model is evaluated beyond the context lengths encountered during training. Prior work has characterized the effective receptive field of Mamba through output-side metrics that measure the influence of earlier tokens on a particular output position (Ben-Kish et al., 2025; Ye et al., 2025). Our goal is different: instead of asking how much information a particular output can retrieve from the past, we directly characterize how the recurrent state itself changes as the distance between positions increases. This distinction is useful because the state is shared across subsequent token predictions and provides a direct view of the internal recurrent dynamics (see Appendix B for a comparison with effective-receptive-field metrics).

For a recurrent layer, let $\mathbf{h}_t$ denote the hidden state after processing token t. We measure the cosine similarity between two (flattened) states,

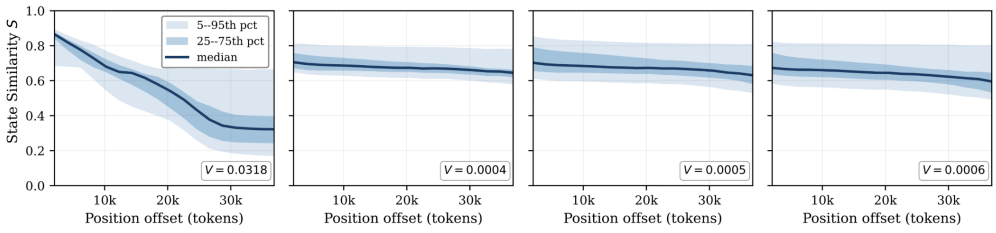
$$\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i^\top \mathbf{h}_j}{\|\mathbf{h}_i\| \, \|\mathbf{h}_j\|}, \tag{1}$$

and define the *state-similarity profile* as the average similarity between all pairs of states at constant relative distance,

$$S(k) := \mathbb{E}_{\substack{i,j \in [1,L] \\ |i-j|=k}} \left[\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j) \right], \tag{2}$$

where L is the sequence length and k is the positional offset between two states. When $S(k)$ is close to 1, the states remain similar across an offset k, indicating that intervening tokens are strongly discounted and earlier information is largely preserved; values near 0 or below indicate substantial state decorrelation. The effect of k on $S(k)$ is of particular interest: as relative distance increases, does state similarity decay, or does it stay roughly constant, indicating that the state wanders within a stable neighborhood? To summarize this distance dependence with a scalar statistic, we use the variance of $S(k)$ across offsets k to characterize the degree of influence relative distance exerts on state similarity:

$$V(t) := \mathrm{Var}_{m \in [1, R-1]} \left[\widetilde{S}^{(t)} \left(m \cdot L_{\mathrm{chunk}} \right) \right] \tag{3}$$



(a) 8k Base (b) 128k Extended (c) 32k, 500 steps (d) 32k, 6k steps

Figure 1: State similarity curves of Nemotron-H 8B across four training checkpoints, measured on 40K-token PG-19 sequences. After extending training to 32K, state similarity remains nearly stationary from 500 to 6000 steps (V: 0.0005 vs. 0.0006).

Large V indicates pronounced variation in state similarity across positional offsets, while small V indicates that state-similarity is stable over the measured distances. Note that V and S can be orthogonal: a slowly evolving state (high S) might patrol a tight neighborhood (low V), or drift gradually but substantially from its starting point (high V), as could a rapidly evolving state (low S). We track S and V together to capture recurrent-state dynamics throughout long-context adaptation.

Materializing the recurrent state at every token is not supported by standard Mamba kernels (Gu & Dao, 2023; Dao & Gu, 2024), which are optimized to avoid exposing intermediate states. Instead, we estimate the state-similarity profile from states sampled at sequence-chunk boundaries. For a sequence of length $T = RS$, with R chunks of length S, we estimate the profile at offsets mS as

$$\widetilde{S}(mS) := \mathbb{E}_{1 \le i \le R-m} \left[\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_{i+m}) \right], \qquad m = 1, \ldots, R-1. \tag{4}$$

This provides a coarse-grained estimate of the distance-wise profile for characterizing its overall shape while avoiding fine-grained state materialization.

3.2 Phase I: State Equilibration

We first examine how recurrent-state behavior changes during long-context adaptation. Fig. 1 compares Nemotron-H-8B at four checkpoints, all evaluated on the same 40K-token sequences from the PG-19 validation set: the 8K pretrained base model, checkpoints after 500 and 6000 steps of continued training at 32K context, and the fully adapted 128K checkpoint. The pretrained model exhibits pronounced distance dependence in its recurrent states: state similarity decreases substantially as positional offset grows, yielding a distance-impact statistic of $V = 0.0318$. This behavior changes rapidly once long-context adaptation begins. After only 500 steps at 32K context, V falls to 0.0005 and the state-similarity profile becomes nearly flat over the measured offsets. Different Mamba heads still stratify into horizontal bands representing different time-horizons (mean S), but notably these horizons remain stable over increasing distance.

More importantly, continued training changes this profile very little. After $12\times$ as much training (6000 steps), $V = 0.0006$; the median profile and percentile bands are nearly indistinguishable from those at 500 steps. The fully adapted 128K checkpoint reaches $V = 0.0004$, placing it in essentially the same regime. We call this rapid transition *state equilibration*: the phase during which recurrent-state dynamics adapt to the enlarged sequence horizon and enter a stable distance- *and training*-independent regime. We define equilibration as the point at which the state-similarity profile ceases to change materially with continued training, and note that this correlates empirically to small V, since V drops rapidly in this initial phase.

3.3 Phase II: Context-Length Specialization

State equilibration does not imply that optimization at the current context length has converged. After recurrent-state dynamics enter a stable regime, continued training can still improve performance at the current context length. However, as we show empirically in § 5.2, these later gains transfer poorly to longer sequences. We refer to this second regime as *context-length specialization*. During this phase, the model continues adapting to the current training horizon even though its distance-dependent recurrent-state dynamics have already stabilized. The resulting improvements are therefore increasingly specific to the current context length instead of contributing to further context scaling. This distinction is important for multi-stage long-context adaptation: *convergence at the current context length is not the same as completion of transferable long-context adaptation*. We



Figure 2: (**a**) The serial P2P algorithm of Dao & Gu (2024) propagates recurrent states across device sequentially, introducing inter-device idle time that grows with the degree of parallelism. (**b**) Our all-gather-based algorithm exchanges local summary collectively, allowing prefix-state materialization to proceed in parallel across devices and reduces pipeline stalls.

further observe similar position-dependent state drift in a Gated DeltaNet hybrid and pure Mamba-2, suggesting that the phenomenon extends beyond the two hybrid models studied here, while exhibiting architecture-dependent differences in its magnitude and shape. Detailed results are in Appendix **??**. If an intermediate stage primarily serves as a transition that prepares the model for subsequent, longer contexts, then training deep into the specialization phase may provide little additional benefit for further context scaling. This observation motivates the State-Equilibration Curriculum.

4 State-Equilibration Curriculum for Long-Context Adaptation

The two-phase behavior in § 3 suggests a different role for intermediate context stages: they need only adapt the model sufficiently for further context scaling, rather than fully optimize performance at the current length. This motivates **State-Equilibration Curriculum (SEC)**, which advances context length once the recurrent-state dynamics stabilize. Concretely, SEC monitors the distance-impact statistic $V(t)$ at each stage and moves to the next context length when the recurrent-state profile has equilibrated. Each stage starts from the previous checkpoint, while training at the final target length continues until the desired end-task criterion is met.

4.1 Detecting State Equilibration

Conceptually, state equilibration corresponds to stabilization of the state-similarity profile instead of attainment of a particular absolute value of V. Empirically, this transition is exhibited as a rapid drop in $V(t)$ into a stable low-V regime, often by nearly two orders of magnitude (e.g., to the order of 5×10^{-4} in our tested setting), within a few hundred training steps. Operationally, SEC uses a distance-impact threshold τ to detect this transition: an intermediate stage advances once $V(t) \leq \tau$. The threshold is then used as a coarse indicator of entry into the stable regime. We evaluate the sensitivity of SEC to τ and report the exact values used in § 5. Loss and downstream performance at the current context length continue to improve after state equilibration, even though these later gains transfer poorly to longer sequences. Waiting for such metrics to converge therefore trains well into the context-length specialization phase. A fixed step budget can avoid this cost when the appropriate duration is known in advance, but does not adapt when equilibration occurs at different rates. SEC instead ties the stage transition directly to the model's recurrent-state dynamics. The required statistic is inexpensive to collect: our context-parallel implementation already materializes recurrent states at chunk boundaries, so monitoring $V(t)$ requires no additional state recomputation (§ 4.2).

4.2 Efficient Context Parallelism for SEC

SEC requires both scalable long-context continued training and efficient observation of recurrent states. For Transformer layers, existing context-parallel attention already distributes sequence computation across devices (Liu et al., 2023; Jacobs et al., 2024). The recurrent component is more challenging because its hidden state introduces an explicit dependency across sequence shards. A straightforward Mamba2 context-parallel implementation preserves this dependency: each rank processes its local sequence chunk only after receiving the final recurrent state from the previous rank (Dao & Gu, 2024). Thus, for R sequence shards, recurrent execution forms an R-stage serialized critical path, with later ranks unable to make progress until all preceding chunks have been processed. Although the sequence is partitioned across devices for memory scalability, the recurrent computation itself exposes little cross-rank parallelism. We remove this serial dependency by exploiting the affine structure of the recurrence.

Let rank r own a local chunk of length S, indexed by $s = 0, \ldots, S-1$, and let $\mathbf{h}_{\text{in}}^{(r)}$ denote the recurrent state entering the chunk. The local recurrence is

$$\mathbf{h}_s^{(r)} = \overline{a}_s^{(r)} \mathbf{h}_{s-1}^{(r)} + \mathbf{z}_s^{(r)}, \qquad \mathbf{h}_{-1}^{(r)} = \mathbf{h}_{\text{in}}^{(r)}. \tag{5}$$

Unrolling the recurrence gives

$$\mathbf{h}_s^{(r)} = \overline{a}_{0:s}^{(r)} \mathbf{h}_{\text{in}}^{(r)} + \mathbf{z}_s^{(r)} + \overline{a}_s^{(r)} \mathbf{z}_{s-1}^{(r)} + \overline{a}_{s-1:s}^{(r)} \mathbf{z}_{s-2}^{(r)} + \cdots + \overline{a}_{1:s}^{(r)} \mathbf{z}_0^{(r)}, \tag{6}$$

where $\overline{a}_{i:j}^{(r)} := \prod_{i \le k \le j} \overline{a}_k^{(r)}$. The first term contains the entire dependence on the state entering the chunk, while all remaining terms depend only on the local inputs.

Therefore, the complete transformation performed by rank r can be written compactly as

$$\mathbf{h}_{\text{out}}^{(r)} = \mathbf{A}^{(r)} \mathbf{h}_{\text{in}}^{(r)} + \mathbf{b}^{(r)}, \tag{7}$$

where $\mathbf{A}^{(r)}$ summarizes propagation of the incoming state through the chunk and $\mathbf{b}^{(r)}$ summarizes the contribution of the local inputs. Both quantities can be computed using only the sequence shard assigned to rank r, without knowing the true incoming recurrent state.

This decomposition allows every rank to make progress before the cross-rank state dependency is resolved. In the first pass, each rank independently computes its chunk-local transfer summary $(\mathbf{A}^{(r)}, \mathbf{b}^{(r)})$. Concretely,

$$\mathbf{A}^{(r)} = \overline{a}_{0:S-1}^{(r)}, \tag{8}$$

while $\mathbf{b}^{(r)}$ is obtained by evaluating the local recurrence with zero incoming state. All ranks then exchange their transfer summaries using an all-gather. Because affine chunk transformations compose, rank r can reconstruct its correct incoming state from the summaries of all preceding chunks:

$$\mathbf{h}_{\text{in}}^{(r)} = \mathbf{b}^{(r-1)} + \mathbf{A}^{(r-1)} \mathbf{b}^{(r-2)} + \cdots + \mathbf{A}^{(r-1)} \mathbf{A}^{(r-2)} \cdots \mathbf{A}^{(1)} \mathbf{b}^{(0)}. \tag{9}$$

A second local pass then evaluates the recurrence with the corrected incoming state and produces the final hidden states and outputs.

Chunk-Summary Context Parallelism

Step 1: Each rank r independently computes its chunk-local transfer summary $(\mathbf{A}^{(r)}, \mathbf{b}^{(r)})$, where

$$\mathbf{A}^{(r)} = \overline{a}_{0:S-1}^{(r)}, \qquad \mathbf{b}^{(r)} = \mathbf{h}_{\text{out}}^{(r)}\big|_{\mathbf{h}_{\text{in}}^{(r)} = \mathbf{0}}.$$

Step 2: All-gather $(\mathbf{A}^{(r)}, \mathbf{b}^{(r)})$ across ranks.

Step 3: Each rank composes the summaries of preceding chunks to reconstruct its correct incoming state $\mathbf{h}_{\text{in}}^{(r)}$.

Step 4: Each rank reruns its local recurrence with $\mathbf{h}_{\text{in}}^{(r)}$ to produce the correct hidden states and outputs.

The first local pass does not produce the final recurrent activations; its role is to summarize how each chunk transforms an arbitrary incoming state. Our design therefore performs part of the recurrent computation twice. Each local pass requires $O(S)$ computation per rank for a chunk of length S, while reconstructing the incoming state from the gathered summaries requires $O(R)$ work. The all-gather materializes $O(R)$ chunk summaries per rank. This storage can be reduced to $O(1)$ by iteratively propagating and composing states through P2P communication, as suggested by Dao & Gu (2024) and illustrated in Figure 2, but doing so reintroduces the serial cross-rank dependency. Our design instead replaces the R-rank state-propagation chain with two parallel local passes separated by a single collective synchronization, trading redundant local computation and additional communication for a substantially shorter distributed critical path. The duplicated recurrence accounts for only part of the overall Mamba2 layer cost, since much of the remaining computation is already parallel over the sequence dimension. We evaluate this tradeoff empirically in § 5.

The same execution also provides the signal required by SEC. Context parallelism naturally materializes recurrent states at sequence-chunk boundaries, since these states define the interface between adjacent shards. We reuse these boundary states to estimate the state-similarity profile $S(k)$ and distance-dependence statistic $V(t)$ from § 3.1. Monitoring state equilibration therefore requires no additional full-sequence state materialization or custom recurrent kernel.

Table 1: RULER performance (%) for One-Stage, Multi-Stage, SEC-Default, and SEC-Early.

Model	Len	Method	S1	S2	S3	M1	M2	M3	MV	MQ	VT	CWE	FWE	QA1	QA2	Avg.
Nemotron-H-8B	32k	One-Stage	100.0	64.1	95.7	46.7	91.8	64.1	45.8	75.7	62.1	67.1	62.9	8.2	16.2	61.6
		Multi-Stage	100.0	98.8	100.0	47.8	96.4	68.6	54.1	78.9	86.1	68.6	97.6	7.2	17.0	70.8
		SEC-Default	100.0	100.0	100.0	50.1	100.0	70.8	56.8	82.5	83.2	72.0	92.1	7.6	17.8	**71.8**
		SEC-Early	100.0	98.8	100.0	55.9	96.4	68.7	60.6	79.0	79.2	70.0	98.4	7.2	17.0	71.6
	64k	One-Stage	100.0	75.1	57.8	51.3	88.8	63.1	35.8	74.9	81.8	67.1	93.7	4.6	14.6	62.2
		Multi-Stage	100.0	94.2	96.6	53.2	91.7	65.4	65.4	54.4	91.6	67.0	93.7	8.2	16.2	69.0
		SEC-Default	100.0	73.3	90.8	55.7	96.3	67.4	68.7	57.1	95.8	70.3	98.3	8.6	17.0	**69.2**
		SEC-Early	100.0	94.2	96.8	49.9	91.8	65.4	65.4	54.4	91.6	67.0	93.7	8.2	16.2	68.8
	128k	One-Stage	100.0	81.9	96.8	53.2	87.2	66.2	65.8	71.6	91.6	51.6	93.7	7.5	17.1	**68.0**
		Multi-Stage	100.0	96.1	93.9	55.2	81.3	63.5	48.0	75.2	79.7	60.1	91.1	7.7	8.1	66.1
		SEC-Default	100.0	100.0	98.6	51.6	71.6	66.6	50.5	78.9	83.7	63.1	95.6	8.1	8.5	67.4
		SEC-Early	100.0	96.1	96.2	55.2	91.7	63.5	58.7	75.2	79.7	60.1	91.1	7.7	8.1	**68.0**
Bamba-9B-v2	32k	One-Stage	100.0	100.0	70.7	69.7	89.4	32.8	68.9	80.1	71.2	1.1	61.3	44.3	43.6	64.1
		Multi-Stage	100.0	100.0	100.0	77.0	94.8	32.8	74.0	81.0	79.1	1.2	73.7	44.5	43.6	69.4
		SEC-Default	100.0	100.0	100.0	80.8	92.3	34.4	77.4	84.8	83.0	1.2	74.8	44.6	40.8	70.3
		SEC-Early	100.0	100.0	100.0	79.7	99.5	34.4	76.3	85.0	74.3	1.3	77.2	46.5	45.1	**70.7**
	64k	One-Stage	99.1	98.6	95.3	72.6	92.4	17.5	71.6	67.2	61.3	0.7	62.6	45.2	37.6	63.2
		Multi-Stage	100.0	97.1	96.2	72.6	92.6	20.8	74.2	73.0	62.4	0.7	62.8	45.2	37.6	64.2
		SEC-Default	100.0	93.7	100.0	76.2	96.7	20.3	77.8	76.6	65.6	0.7	65.2	47.4	38.7	**66.1**
		SEC-Early	100.0	100.0	100.0	69.8	93.9	21.6	73.8	76.2	55.5	0.7	65.9	47.5	39.5	64.9
	128k	One-Stage	100.0	98.2	98.5	52.8	74.6	5.4	65.6	64.2	35.5	0.6	58.3	42.8	33.9	56.2
		Multi-Stage	100.0	99.7	98.2	53.0	77.6	5.1	66.9	63.5	36.7	0.7	58.3	42.8	35.6	56.8
		SEC-Default	100.0	100.0	100.0	55.5	78.3	5.7	67.6	69.2	39.0	0.8	61.2	44.9	36.3	**58.3**
		SEC-Early	100.0	100.0	89.2	51.7	78.3	5.3	66.2	69.0	39.1	0.7	61.2	44.4	37.4	57.1

5 EXPERIMENTS

Models. We evaluate SEC on two publicly available hybrid SSM–Transformer models: Nemotron-H-8B (NVIDIA et al., 2025b), with an 8K native context length, and Bamba-9B-v2 (Ganti et al.), with a 4K native context length.

Datasets. We train on a mixture of Nemotron-CC (Su et al., 2024) and Dolma (Soldaini et al., 2024). Training data are sampled with a length-biased policy and packed with EOS-delimited document boundaries. We evaluate long-context performance with RULER (Hsieh et al., 2024) at 32K, 64K, and 128K, reporting all 13 synthetic-collection subtasks individually and as an average. We additionally evaluate the same checkpoints on InfiniteBench to test whether the observed trends extend beyond synthetic long-context tasks. For short-context capability preservation, we report 4K PG-19 perplexity and ARC-C, HellaSwag, and Winogrande.

Training schedules. We compare four long-context adaptation schedules under the same data, optimizer, global token batch size, and positional-scaling configuration. *One-Stage* directly adapts the model at the 128K target context using the full 9B-token budget. *Multi-Stage* follows a full multi-stage schedule, training for 3B tokens at 32K before continuing for 6B tokens at 128K. *SEC-Default* uses the same 6B-token target stage as Multi-Stage but advances from 32K once $V(t) \leq \tau$, with $\tau = 5 \times 10^{-4}$. In our default setting, this transition occurs after approximately 0.25B tokens of 32K training. *SEC-Early* uses a more aggressive transition criterion, $\tau = 10^{-3}$, and advances after approximately 0.15B tokens at 32K. We use this variant to evaluate sensitivity to the exact equilibration point. Across the multi-stage schedules, only the maximum context length and associated rotary-embedding scaling change between stages, while the global token batch size is held constant. Full data-preparation and hyperparameter details are provided in Appendix A.

5.1 MAIN RESULTS

We first evaluate whether advancing at state equilibration preserves the benefit of multi-stage long-context adaptation. Table 1 compares One-Stage, Multi-Stage, SEC-Default, and SEC-Early on Nemotron-H-8B and Bamba-9B-v2. Across both models, Multi-Stage generally outperforms One-Stage, confirming the benefit of an intermediate 32K adaptation stage. SEC retains this benefit while using only a small fraction of the intermediate-stage budget: Multi-Stage trains for 3B tokens at 32K, whereas SEC-Default advances after approximately 0.25B tokens before using the same 6B-token 128K stage. SEC-Default matches or improves upon Multi-Stage at every reported evaluation length. On Nemotron-H-8B, SEC-Default achieves average RULER scores of 71.8, 69.2, and 67.4 at 32K, 64K, and 128K, compared with 70.8, 69.0, and 66.1 for Multi-Stage. On Bamba-9B-v2, the corresponding scores are 70.3, 66.1, and 58.3, compared with 69.4, 64.2, and 56.8. Thus, training substantially longer at the intermediate context provides no consistent downstream advantage over advancing at state equilibration. The shorter intermediate stage reduces the total training budget

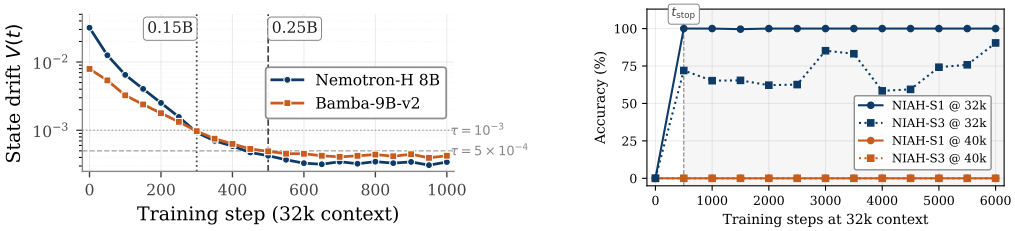


Figure 3: State equilibration and transfer during the 32K intermediate stage. **Left**: State drift $V(t)$ during 32K training, sampled every 50 steps. **Right**: NIAH accuracy during 32K training, evaluated at 32K and 40K.

by approximately 30%. The corresponding wall-clock savings are substantial: on 32 A100 GPUs, SEC-Default reduces training time from 22 to 13 hours for Nemotron-H-8B and from 26 to 15 hours for Bamba-9B-v2, a reduction of roughly 42% relative to Multi-Stage.

The same qualitative trend holds on InfiniteBench, as shown in Table 2. SEC-Default closely matches the full Multi-Stage baseline on retrieval for both Nemotron-H-8B and Bamba-9B-v2, despite using only a small fraction of the 32K-stage training budget. Performance on language, code, and mathematics is more task-dependent, but the reduced-budget SEC schedules remain broadly comparable overall.

Table 2: InfiniteBench performance across long-context adaptation schedules.

Model	Method	Retrieve	En	Code	Math
Nemotron-H-8B	One-Stage	47.9	14.1	7.9	9.5
	Multi-Stage	63.5	15.4	6.0	10.5
	SEC-Default	62.3	14.9	7.2	9.5
	SEC-Early	62.2	16.0	7.1	9.4
Bamba-9B-v2	One-Stage	52.5	14.1	1.0	6.8
	Multi-Stage	56.7	15.3	1.0	7.8
	SEC-Default	56.6	14.9	0.7	7.5
	SEC-Early	55.7	14.5	0.1	7.5

5.2 Analysis Results

Does State Equilibration Predict When to Advance? Fig. 3 (left) plots the distance-impact statistic $V(t)$ during the 32K intermediate stage. For both models, $V(t)$ decreases rapidly, e.g., from approximately 3×10^{-2} for Nemotron-H-8B and 8×10^{-3} for Bamba-9B-v2, and reaches a stable low-V regime around 0.25B training tokens that persists through subsequent training. SEC-Early transitions after approximately 0.15B tokens while $V(t)$ is still decreasing ($\sim 10^{-3}$), whereas SEC-Default transitions near the onset of the stable regime at $V(t) \approx 5 \times 10^{-4}$. Despite different native context lengths, both models show similar equilibration trajectories.

Fig. 3 (right) shows that downstream performance evolves differently. At 32K, NIAH-S1 saturates near the SEC-Default transition, while the harder NIAH-S3 task continues improving from approximately 72% to 90% with further 32K training. However, these gains do not transfer even to 40K, where both tasks remain at zero throughout training. Thus, after recurrent-state dynamics stabilize, the model continues to specialize to the current context length. This supports the two-phase behavior identified in § 3: rapid *state equilibration* followed by *context-length specialization*. It also explains why SEC-Default slightly outperforms SEC-Early in Table 1: the latter advances before the distance-impact trajectory has fully entered its stable regime.

How Sensitive Is SEC to the Transition Point? We compare SEC-Default, which advances near the transition into the stable low-V regime, with the earlier SEC-Early transition. Despite reducing the intermediate-stage budget from 0.25B to 0.15B tokens, SEC-Early remains competitive: Across models and context lengths in Table 1, the two variants differ by at most 1.2 points, with SEC-Default performing better in four cases. This suggests that SEC does not depend on a precisely tuned transition point, as long as the model is near the state-equilibration regime.

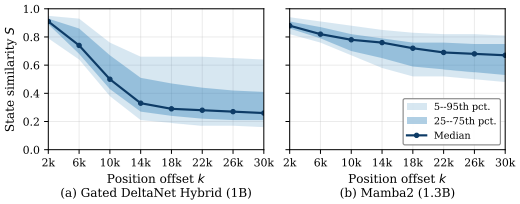


Figure 4: State similarity profile for a Gated DeltaNet hybrid and Mamba2.

Does State Drift Generalize Beyond Mamba Hybrids? We further examine a Gated DeltaNet hybrid model and the official Mamba-2 checkpoint, as shown in Fig. 4. Both exhibit distance-dependent state drift: in the Gated DeltaNet hybrid, median state similarity decreases from 0.91 at 2K offset to 0.26 at 30K, while Mamba-2 shows a milder decline from 0.88 to 0.67 over the same range. These results suggest that the phenomenon is not specific to the Mamba-based hybrid models used in our main experiments.

Does SEC Preserve Short-Context Capabilities? We evaluate the same checkpoints on 4K PG-19 perplexity together with ARC-C, HellaSwag, and Winogrande, as shown in Table 3. Across both models, SEC remains broadly comparable to the longer Multi-Stage baseline, with no systematic degradation in short-context language modeling or general reasoning performance. These results indicate that reducing the intermediate long-context training budget does not come at the expense of short-context capability.

Table 3: Comparison of short-context language modeling and downstream task performance across configurations.

Model	Method	PG-19 4k ↓	ARC-C ↑	HellaSwag ↑	WinoGrande ↑
Nemotron-H-8B	One-Stage	9.87	88.55	82.28	81.56
	Multi-Stage	9.81	90.42	81.49	82.40
	SEC-Default	9.75	87.47	82.64	81.98
	SEC-Early	9.79	87.13	82.53	80.19
Bamba-9B-v2	One-Stage	10.59	64.08	84.47	77.97
	Multi-Stage	9.64	63.53	81.96	79.21
	SEC-Default	9.89	62.84	82.41	79.26
	SEC-Early	9.96	63.38	82.97	79.90

How Does SEC Compare with Training-Free Context Extension? We compare against UPI and LongMamba, representative training-free methods for stateful models. We apply each method to the base checkpoint, the 32K/0.15B checkpoint at the early SEC transition, and the 32K/3B checkpoint used by the Multi-Stage baseline. Training-free extension performs poorly when applied directly to the base models, but improves substantially after 32K adaptation. Even so, it remains below dedicated SEC training at 128K, suggesting that training-free extension is complementary to multi-stage adaptation.

Table 4: RULER performance (%) of UPI and LongMamba applied to base and 32K-adapted checkpoints.

Model	Checkpoint	UPI		LongMamba	
		64k	128k	64k	128k
Nemotron-H-8B	Base	10.25	5.12	11.56	9.75
	32k / 0.15B	15.46	8.20	11.92	9.18
	32k / 3B	20.26	9.57	16.97	8.70
Bamba-9B-v2	Base	9.68	6.71	10.28	3.92
	32k / 0.15B	18.79	14.26	11.75	3.92
	32k / 3B	24.93	16.65	15.35	4.29

Does the Two-Pass Context-Parallel Design Improve Throughput? Fig. 5 compares weak-scaling forward-pass throughput of our two-pass all-gather implementation with serial P2P state propagation. The all-gather design consistently achieves higher throughput, and the gap widens as the sequence length and context-parallel degree increase. This confirms that replacing the cross-rank recurrent dependency with parallel local passes and a single collective synchronization more than offsets the additional local recurrence work and communication. The duplicated recurrence has limited overhead because much of Mamba2's layer computation remains parallel over the sequence dimension, including computation of the discretization parameters such as Δ_t. As a result, the two-pass design substantially shortens the distributed critical path without doubling the overall layer cost.

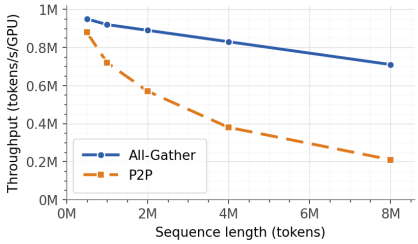


Figure 5: Forward-pass throughput of Mamba2 under context parallelism, comparing our all-gather implementation with serial P2P state propagation.

6 Conclusion

We study how hybrid SSM-Transformer models adapt when their context length is progressively extended. Our results reveal a two-phase process: recurrent-state dynamics stabilize early, while subsequent training primarily improves performance at the current context length and transfers poorly to longer contexts. This reveals that intermediate stages need not be trained to convergence. Based on this insight, we introduce State-Equilibration Curriculum (SEC), which advances the context length once current-state dynamics stabilize. Across Nemotron-H-8B and Bamba-9B-v2, SEC matches or improves fully trained multi-stage schedules while using about 30% fewer training tokens and reducing wall-clock training time by up to 42%.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, interpreting results, and generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, are not applicable to this work.

Additionally, we used generative AI tools for edit a research paper to improve readability and identify relevant literature. We have reviewed all AI-assisted work. AI-revised text was subsequently reviewed and edited by the authors to ensure consistency with the rest of the manuscript, and AI-assisted method implementations were independently verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. Decimamba: Exploring the length extrapolation potential of mamba. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=iWSl5Zyjjw.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *CoRR*, abs/2306.15595, 2023.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024.

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.

Raghu Ganti, Lingsong Chu, Divya Kumari, Mudhakar Srivatsa, Antoni Viros, Davis Wertheimer, Naigang Wang, Nick Hill, Yu Chin Fabian Lim, Thomas Parnell, Joshua Rosenkranz, Chi-Chun (Charlie) Liu, and Garrett Goon. Bamba-9B-v2 - Fast and powerful! https://huggingface.co/blog/ibm-ai-platform/bamba-9b-v2.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In *ACL*, 2025.

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

Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search, 2025.

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

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

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Reza Yazdani Aminadabi, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. System optimizations for enabling

training of extreme long sequence transformer models. In *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing*, PODC '24, pp. 121–130, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706684. doi: 10.1145/3662158.3662806. URL https://doi.org/10.1145/3662158.3662806.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: transformers are better than state space models at copying. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Barak Lenz, 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, Opher Lieber, Or Dagan, Orit Cohavi, Raz Alon, Ro'i Belson, Roi Cohen, Rom Gilad, Roman Glozman, Shahar Lev, Shaked Meirom, Tal Delbari, Tal Ness, Tomer Asida, Tom Ben Gal, Tom Braude, Uriya Pumerantz, Yehoshua Cohen, Yonatan Belinkov, Yuval Globerson, Yuval Peleg Levy, and Yoav Shoham. Jamba-1.5: Hybrid transformer-mamba models at scale. *CoRR*, abs/2408.12570, 2024. URL https://doi.org/10.48550/arXiv.2408.12570.

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.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.

LocalLLaMA. Dynamically scaled rope further increases performance of long context LLaMA with zero fine-tuning. https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/, 2023a. Reddit post, r/LocalLLaMA.

LocalLLaMA. Ntk-aware scaled rope allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degration. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/, 2023b. Reddit post, r/LocalLLaMA.

Meta AI. Introducing llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/, July 2024.

Meta AI. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/, April 2025.

NVIDIA, :, Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, Alex Kondratenko, Alex Shaposhnikov, Alexander Bukharin, Ali Taghibakhshi, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amy Shen, Andrew Tao, Ann Guan, Anna Shors, Anubhav Mandarwal, Arham Mehta, Arun Venkatesan, Ashton Sharabiani, Ashwath Aithal, Ashwin Poojary,

Ayush Dattagupta, Balaram Buddharaju, Banghua Zhu, Barnaby Simkin, Bilal Kartal, Bita Darvish Rouhani, Bobby Chen, Boris Ginsburg, Brandon Norick, Brian Yu, Bryan Catanzaro, Charles Wang, Charlie Truong, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christian Munley, Christopher Parisien, Dan Su, Daniel Afrimi, Daniel Korzekwa, Daniel Rohrer, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Dima Rekesh, Dina Yared, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Eileen Long, Elliott Ning, Eric Chung, Erick Galinkin, Evelina Bakhturina, Gargi Prasad, Gerald Shen, Haifeng Qian, Haim Elisha, Harsh Sharma, Hayley Ross, Helen Ngo, Herman Sahota, Hexin Wang, Hoo Chang Shin, Hua Huang, Iain Cunningham, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jian Zhang, Jiaqi Zeng, Jimmy Zhang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jonathan Cohen, Joseph Jennings, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kezhi Kong, Krzysztof Pawelec, Kumar Anik, Kunlun Li, Kushan Ahmadian, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Luis Vega, Maer Rodrigues de Melo, Makesh Narsimhan Sreedhar, Marcin Chochowski, Mark Cai, Markus Kliegl, Marta Stepniewska-Dziubinska, Matvei Novikov, Mehrzad Samadi, Meredith Price, Meriem Boubdir, Michael Boone, Michael Evans, Michal Bien, Michal Zawalski, Miguel Martinez, Mike Chrzanowski, Mohammad Shoeybi, Mostofa Patwary, Namit Dhameja, Nave Assaf, Negar Habibi, Nidhi Bhatia, Nikki Pope, Nima Tajbakhsh, Nirmal Kumar Juluru, Oleg Rybakov, Oleksii Hrinchuk, Oleksii Kuchaiev, Oluwatobi Olabiyi, Pablo Ribalta, Padmavathy Subramanian, Parth Chadha, Pavlo Molchanov, Peter Dykas, Peter Jin, Piotr Bialecki, Piotr Januszewski, Pradeep Thalasta, Prashant Gaikwad, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Rabeeh Karimi Mahabadi, Rajen Patel, Ran El-Yaniv, Ranjit Rajan, Ria Cheruvu, Rima Shahbazyan, Ritika Borkar, Ritu Gala, Roger Waleffe, Ruoxi Zhang, Russell J. Hewett, Ryan Prenger, Sahil Jain, Samuel Kriman, Sanjeev Satheesh, Saori Kaji, Sarah Yurick, Saurav Muralidharan, Sean Narenthiran, Seonmyeong Bak, Sepehr Sameni, Seungju Han, Shanmugam Ramasamy, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shizhe Diao, Shreya Gopal, Shrimai Prabhumoye, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Siddhartha Jain, Somshubra Majumdar, Soumye Singhal, Stefania Alborghetti, Syeda Nahida Akter, Terry Kong, Tim Moon, Tomasz Hliwiak, Tomer Asida, Tony Wang, Tugrul Konuk, Twinkle Vashishth, Tyler Poon, Udi Karpas, Vahid Noroozi, Venkat Srinivasan, Vijay Korthikanti, Vikram Fugro, Vineeth Kalluru, Vitaly Kurin, Vitaly Lavrukhin, Wasi Uddin Ahmad, Wei Du, Wonmin Byeon, Ximing Lu, Xin Dong, Yashaswi Karnati, Yejin Choi, Yian Zhang, Ying Lin, Yonggan Fu, Yoshi Suhara, Zhen Dong, Zhiyu Li, Zhongbo Zhu, and Zijia Chen. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model, 2025a.

NVIDIA, :, Aaron Blakeman, Aarti Basant, Abhinav Khattar, Adithya Renduchintala, Akhiad Bercovich, Aleksander Ficek, Alexis Bjorlin, Ali Taghibakhshi, Amala Sanjay Deshmukh, Ameya Sunil Mahabaleshwarkar, Andrew Tao, Anna Shors, Ashwath Aithal, Ashwin Poojary, Ayush Dattagupta, Balaram Buddharaju, Bobby Chen, Boris Ginsburg, Boxin Wang, Brandon Norick, Brian Butterfield, Bryan Catanzaro, Carlo del Mundo, Chengyu Dong, Christine Harvey, Christopher Parisien, Dan Su, Daniel Korzekwa, Danny Yin, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Denys Fridman, Dima Rekesh, Ding Ma, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Dusan Stosic, Eileen Long, Elad Segal, Ellie Evans, Eric Chung, Erick Galinkin, Evelina Bakhturina, Ewa Dobrowolska, Fei Jia, Fuxiao Liu, Gargi Prasad, Gerald Shen, Guilin Liu, Guo Chen, Haifeng Qian, Helen Ngo, Hongbin Liu, Hui Li, Igor Gitman, Ilia Karmanov, Ivan Moshkov, Izik Golan, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jarno Seppanen, Jason Lu, Jason Sewall, Jiaqi Zeng, Jiaxuan You, Jimmy Zhang, Jing Zhang, Jining Huang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jon Barker, Jonathan Cohen, Joseph Jennings, Jupinder Parmar, Karan Sapra, Kari Briski, Kateryna Chumachenko, Katherine Luna, Keshav Santhanam, Kezhi Kong, Kirthi Sivamani, Krzysztof Pawelec, Kumar Anik, Kunlun Li, Lawrence McAfee, Leon Derczynski, Lindsey Pavao, Luis Vega, Lukas Voegtle, Maciej Bala, Maer Rodrigues de Melo, Makesh Narsimhan Sreedhar, Marcin Chochowski, Markus Kliegl, Marta Stepniewska-Dziubinska, Matthieu Le, Matvei Novikov, Mehrzad Samadi, Michael Andersch, Michael Evans, Miguel Martinez, Mike Chrzanowski, Mike Ranzinger, Mikolaj Blaz, Misha Smelyanskiy, Mohamed Fawzy, Mohammad Shoeybi, Mostofa Patwary, Nayeon Lee, Nima Tajbakhsh, Ning Xu, Oleg Rybakov, Oleksii Kuchaiev, Olivier Delalleau, Osvald Nitski, Parth Chadha, Pasha Shamis, Paulius Micikevicius, Pavlo Molchanov, Peter Dykas, Philipp Fischer, Pierre-Yves Aquilanti, Piotr Bialecki, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Rabeeh Karimi, Rahul Kandu, Ran El-Yaniv, Raviraj Joshi, Roger Waleffe, Ruoxi Zhang, Sabrina Kavanaugh, Sahil Jain, Samuel Kriman, Sangkug Lym, Sanjeev Satheesh, Saurav Muralidharan, Sean Narenthiran, Selvaraj Anandaraj, Seonmyeong Bak,

Sergey Kashirsky, Seungju Han, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Sharon Clay, Shelby Thomas, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shyamala Prayaga, Siddhartha Jain, Sirshak Das, Slawek Kierat, Somshubra Majumdar, Song Han, Soumye Singhal, Sriharsha Niverty, Stefania Alborghetti, Suseella Panguluri, Swetha Bhendigeri, Syeda Nahida Akter, Szymon Migacz, Tal Shiri, Terry Kong, Timo Roman, Tomer Ronen, Trisha Saar, Tugrul Konuk, Tuomas Rintamaki, Tyler Poon, Ushnish De, Vahid Noroozi, Varun Singh, Vijay Korthikanti, Vitaly Kurin, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenliang Dai, Wonmin Byeon, Xiaowei Ren, Yao Xu, Yejin Choi, Yian Zhang, Ying Lin, Yoshi Suhara, Zhiding Yu, Zhiqi Li, Zhiyu Li, Zhongbo Zhu, Zhuolin Yang, and Zijia Chen. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models, 2025b.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Qwen Team. Qwen3-next-80b-a3b. https://qwen.ai/blog?id=4074cca80393150c248e508aa62983f9cb7d27cd&from=research.latest-advancements-list, September 2025. Blog post.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *ArXiv*, abs/2406.07522, 2024. URL https://api.semanticscholar.org/CorpusID:270380294.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. Longrope2: Near-lossless llm context window scaling, 2025.

Haochen Shen, Davis Wertheimer, Zheng Wang, Garrett Goon, Derrick Liu, Naigang Wang, Mudhakar Srivatsa, Raghu K. Ganti, and Minjia Zhang. From collapse to control: Understanding and extending context length in emerging hybrid models via universal position interpolation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MjmORKLHUI.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2402.00159.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset, 2024. URL https://arxiv.org/abs/2412.02595.

Shengyang Sun, Yian Zhang, Alexander Bukharin, David Mosallanezhad, Jiaqi Zeng, Soumye Singhal, Gerald Shen, Adithya Renduchintala, Tugrul Konuk, Yi Dong, Zhilin Wang, Dmitry Chichkov, Olivier Delalleau, and Oleksii Kuchaiev. Reward-aware preference optimization: A unified mathematical framework for model alignment, 2025.

A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. 2020.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models. *CoRR*, abs/2406.07887, 2024. URL https://doi.org/10.48550/arXiv.2406.07887.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov,

Mike Lewis, Sinong Wang, and Hao Ma. Effective long-context scaling of foundation models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.260. URL https://aclanthology.org/2024.naacl-long.260/.

Chejian Xu, Wei Ping, Peng Xu, Zihan Liu, Boxin Wang, Mohammad Shoeybi, Bo Li, and Bryan Catanzaro. From 128k to 4m: Efficient training of ultra-long context large language models, 2025. URL https://arxiv.org/abs/2504.06214.

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

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. Longmamba: Enhancing mamba's long-context capabilities via training-free receptive field enlargement. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fMbLszVO1H.

A Implementation and Training Details

This appendix collects the configuration details for the experimental runs reported in Section 5 and the implementation details of the context-parallel pipeline used throughout.

A.1 Training Configuration

Models. We use the publicly released checkpoints of Nemotron-H 8B (NVIDIA et al., 2025b) and Bamba-9B-v2 (Ganti et al.) as starting points. Architectural details, parameter counts, and pretraining context lengths follow the original releases.

Data. Training data is a blend of Nemotron-CC (Su et al., 2024) and Dolma (Soldaini et al., 2024). Natural long sequences are scarce in open-source datasets. Prior work has addressed this through synthetic data generation or by concatenating short sequences into longer ones. Our pipeline biases sampling toward longer documents using a length-dependent sampling probability,

$$p_{\text{doc}} = \min\left(\left(\frac{L_{\text{doc}}}{L_{\text{target}}}\right)^2, 1\right),$$

where documents shorter than a target length L_{target} are sampled with reduced probability. To avoid total document rejection in short-document datasets with max length limited to hundreds of tokens, we set $L_{\text{target}} = 2048$, and pack together documents shorter than the training length to form full-sized training sequences. Some prior works Gao et al. (2025) use attention masks to prevent inter-document attention within a packed sequence. While this matches the test-time data distribution, it also limits the practical input length to that of the underlying documents, which could be much lower than the physical training length. We instead concatenate sequences directly with an `[EOS]` token marking document boundaries, allowing the model to learn where relevant context ends, while still exposing it to out-of-distribution input lengths.

Optimizer and schedule. We use AdamW with learning rate $1\text{e}{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1, and gradient clipping at 1.0. Note that the specific SSM parameters (`A_log`, `D` and `dt_bias`) have no weight decay, following Dao & Gu (2024). Learning rate follows a linear warmup over 300 steps followed by a constant rate in the first stage, and a cosine decay in the second stage. The same schedule is applied across all recipes, scaled to each recipe's total step count.

Batch size. Per-batch token count is held constant across recipes at 524288 tokens per global batch. This means the per-step token count is identical regardless of context length, and the per-step compute scales with context length in the SSM and attention components.

Rotary embedding scaling. RoPE is only utilized by Bamba, whose base frequency is $\theta_{4k} = 10000$. For each stage, the rotary base frequency is adjusted linearly to support the stage's context length. Specifically, $\theta_{32\text{k}} = 320000$ for 32k-context training and $\theta_{128\text{k}} = 1280000$ for 128k-context training. No other positional-encoding modifications are applied.

Per-stage token budgets. Table 5 summarizes the token allocation across recipes. The state-aware recipe's stage 1 stops at 500 steps of 32k training (0.25B tokens), at which point $V(t)$ has reached its plateau; the more aggressive variant stops at 300 steps (0.15B tokens). Stage 2 token counts are identical across all two-stage recipes.

Recipe	Stage 1 (32k)	Stage 2 (128k)	Total
One-stage	—	9B	9B
Two-stage, full	3B	6B	9B
State-aware (0.25B)	0.25B	6B	6.25B
State-aware (0.15B)	0.15B	6B	6.15B

Table 5: Per-stage token budgets for the four recipes evaluated in Section 5.

A.2 Hardware and Parallelism Configuration

Training was performed on 32 NVIDIA A100 GPUs across 4 nodes. We use FSDP2 for parameter, activation, and optimizer sharding, configured to shard within each node to avoid slow inter-node communication. Context parallelism is applied orthogonally and conditioned on sequence length: for

sequences up to 128k, CP is applied intra-node (CP world size 8); for sequences longer than 128k, CP is applied across all nodes. Transformer layers under CP use Ring Attention (Liu et al., 2023); Mamba2 layers use the all-gather-based context-parallel implementation described in Section 4.2. Beyond the recurrent scan, Mamba2 includes a causal depthwise convolution along the sequence dimension. We handle this boundary dependency by passing the final kernel-width elements of each local chunk to the next rank, which then completes the convolution locally.

Wall-clock training times. Table 6 reports approximate wall-clock training times for each recipe and model. The state-aware recipe's $\sim$30% token reduction translates to a larger wall-clock reduction because the saved tokens are at the cheaper 32k context length.

Recipe	Nemotron-H 8B	Bamba-9B-v2
One-stage	24 h	28 h
Two-stage, full	22 h	26 h
State-aware (0.25B)	13 h	15 h
State-aware (0.15B)	12.5 h	14.5 h

Table 6: Approximate wall-clock training time per recipe and model on 32 A100 GPUs.

A.3 State Drift Measurement

State drift $V(t)$ is computed from the chunk-boundary hidden states materialized by the context-parallel pipeline, with chunk size $L_{\text{chunk}} = 2048$ tokens. We measure $V(t)$ on 40k-token sequences sampled from the PG-19 validation set, using 80 non-overlapping windows extracted from 50 source documents. During training, we evaluate $V(t)$ every 50 steps, and the trajectories are shown in Figure **??**.

B Comparison Between Retention Score and Effective Receptive Field Metrics

Prior work has approached the question of Mamba memory through Effective Receptive Field (ERF) analysis, using a metric known as Mamba Mean Distance (Ben-Kish et al., 2025; Ye et al., 2025):

$$\text{MMD} \;=\; \mathbb{E}_{j \le L}\big[d(j, L)\big] = \sum_{j=1}^{L} (L-j)\, \frac{|\mathbf{M}_{L,j}|}{\sum_{i=1}^{L} |\mathbf{M}_{L,i}|},$$

which captures the expected distance (in token positions) between the last token and the weighted influence of earlier tokens, derived from the unrolled recurrence matrix $\mathbf{M}$. However, $\mathbf{M}$ depends on the output projection $\mathbf{C}_L$ for the final token, so MMD measures memory accessible through one specific projection rather than memory present in the state itself. Information preserved in the hidden state but not read out by $\mathbf{C}_L$ is invisible to this metric, even though it may be retrieved by the projections of other tokens. The retention score, by contrast, operates directly on the hidden state and characterizes the decay of state information as a function of position offset, independent of any particular readout.

C Limitations

Our evaluation focuses on context extension up to 128K. While this context range is sufficient to expose the two-phase adaptation behavior studied in this work, whether the same state-equilibration dynamics persist at substantially longer contexts, such as 1M tokens and beyond, remains to be established. We evaluate end-to-end long-context adaptation primarily on RULER and InfiniteBench. These benchmarks cover a range of retrieval and reasoning behaviors, but they do not capture the full diversity of long-context applications. Evaluating SEC on broader real-world workloads and downstream tasks would further clarify where state equilibration provides a reliable stage-transition signal. Finally, our full SEC training experiments cover two hybrid architectures, Nemotron-H-8B and Bamba-9B-v2. We observe qualitatively similar distance-dependent state dynamics in additional stateful architectures, including Gated DeltaNet and Mamba-2, but do not perform full multi-stage

adaptation with SEC on these models. Establishing how the equilibration criterion transfers across a broader range of recurrent and hybrid architectures remains an important direction.