CC-Mamba: Chunk-Conditioned Mamba for Linear-Time, Content-Aware Global Memory

Skalli Yahya (EPFL MSc Data Science) yahya.skalli@epfl.ch

Keywords: state-space models, Mamba, long-context, hybrid models, efficient inference

Motivation

Modern LMs face an efficiency/quality trade-off: Transformers give sharp local reasoning but incur quadratic cost and large KV caches [1]; SSMs (S4) and selective SSMs (Mamba/Mamba-2) give linear-time scans and tiny state but use dynamics that are mostly fixed at global scale [2, 3, 4]. Prior hybrids interleave attention with SSM layers (SPADE, Jamba) or chunkwise recurrence (RetNet), but either keep quadratic attention somewhere or leave SSM dynamics globally fixed [5, 6, 7].

Proposed Approach: CC-Mamba (Chunk-Conditioned Mamba)

Idea. Split the input into chunks. A small local encoder (Transformer) produces per-chunk summaries U_i and a single vector z_i. A *controller* maps z_i to small, stability-safe edits of SSM/Mamba parameters $(A_i, B_i, C_i, \Delta_i)$ *once per chunk*; within the chunk we run a linear-time scan and *carry the recurrent state* x to the next chunk.

Stable parametrization (practical). With gate $w_i \in [0, 1]$,

$$A_i = \mathrm{diag}\big(-\mathrm{softplus}(a_0 + w_i \tanh \delta a_i)\big),\ B_i = B_0 + w_i\, W_B z_i,\ C_i = C_0 + w_i\, W_C z_i,\ \Delta_i = \mathrm{softplus}(\Delta_0 + w_i\, \delta\Delta_i).$$

Discretize per chunk: $\bar{A}_i = e^{\Delta_i A_i}$, $\bar{B}_i = \big(\int_0^{\Delta_i} e^{\tau A_i}\, d\tau\big) B_i$. (SSD/Mamba-2 implementations provide efficient scans [4].)

Why it's different (vs. prior hybrids).

- *Content-aware global memory, linear cost.* Each segment *programs* decay/oscillation/coupling via z_i; no quadratic attention.
- *Fewer controller calls.* Selection runs once per chunk (vs. per token in Mamba) $\Rightarrow$ lower overhead/HBM traffic; still adaptive.
- *Piecewise-smooth dynamics.* Reduces token-to-token jitter and boundary artifacts while retaining precise local attention.

Method Sketch

Front-end (two plug-ins). Option A: local Transformer $\to H_i$ (keep all tokens). **Option B:** local Transformer $\to$ BiGRU $\to$ attention pooling $\to U_i$ ($m \ll c$ summaries). Controller reads $z_i = \mathrm{Pool}(U_i)$.

Scan (within chunk i). For tokens/summaries $u_t \in U_i$: $x_{t+1} = \bar{A}_i x_t + \bar{B}_i u_t, \quad y_t = g_i \odot (C_i x_t) + (1 - g_i) \odot \mathrm{skip}(u_t)$. Carry x to the next chunk.

Impact

CC-Mamba targets the "sweet spot" for long-context LMs: *Transformer-level local precision + content-aware, linear-time global memory* with tiny state—promising better quality at lower latency and cost than fixed-SSM hybrids or quadratic attention.

References

[1] 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.

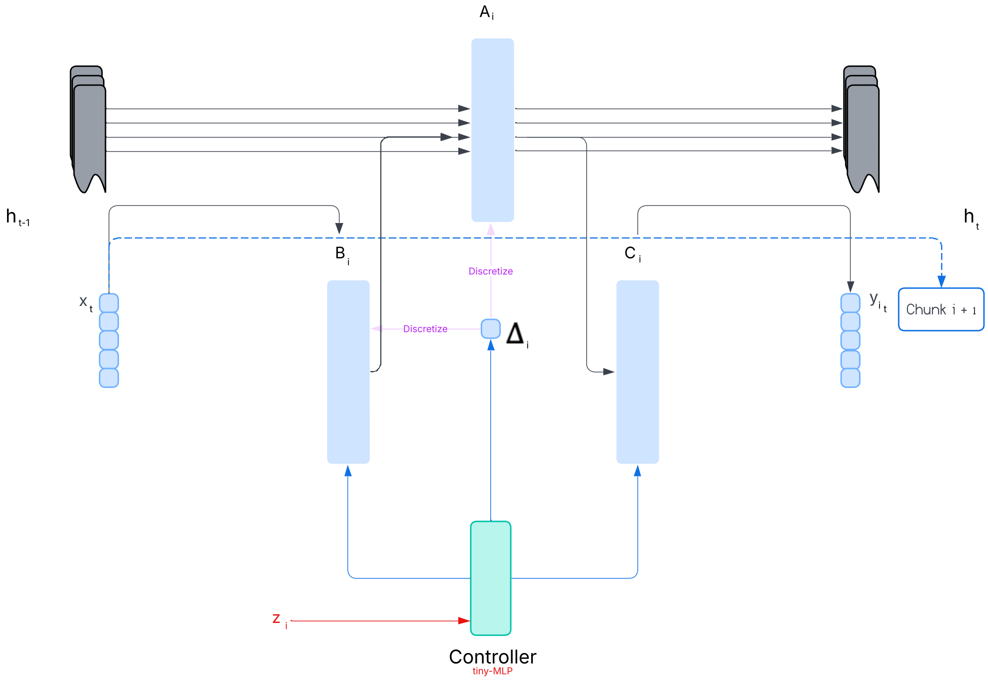

Figure 1: CC-Mamba overview. Diagram style *inspired by* the core Mamba figure [3].

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

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

[4] Tri Dao and Albert Gu. Transformers are ssms via structured state space duality (mamba-2). *arXiv:2405.21060*, 2024.

[5] Simiao Zuo, Haoming Jiang, Shiyu Li, Tuo Zhao, et al. SPADE: State space augmented transformer for efficient long sequence modeling. *arXiv:2212.08136*, 2022. ICLR 2023.

[6] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, et al. Jamba: A hybrid transformer–mamba language model. *arXiv:2403.19887*, 2024.

[7] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, et al. Retentive network: A successor to transformer for large language models. *arXiv:2307.08621*, 2023.