Which sequence architecture keeps analog in-memory compute's advantage? A cost-model answer with a design rule

W. Schulz

Jul 27, 2026

Abstract

Analog in-memory compute accelerates static-weight matmuls but cannot touch attention's growing KV cache, so its decode advantage dilutes with context length. The attention window policy — sliding vs. full — not the attention fraction, decides whether a hybrid keeps the advantage. Distilled from ~70 pre-registered falsification experiments.

Status: Defensive technical publication. Establishes priority for the design rule below and publishes the supporting evidence as open know-how. Date: 2026-07-27. Revised: 2026-07-29 — folded in the completed prefill-flank kill tests (K-series), the fabricated-silicon throughput/area calibration (K2), and the quantified gain-cell falsifier (T2). Author of record: Will Schulz. Provenance: distilled from the `codesign-analog-moe/` falsification campaign (~70 pre-registered experiments, rounds 1–13). Every headline number is reproduced in that directory; experiment IDs are cited inline so each claim is traceable to its `run_*.py` / `SYNTHESIS-*.md` source.

> What is and isn't claimed. This note does not claim a new hardware substrate, a new model architecture, or a quantum/optical advantage — the parent program falsified those. It claims one design rule for deploying already-published hybrid language models on already-published analog in-memory hardware, plus the measured evidence that fixes the rule's boundary. The rule is a checkable engineering guideline; it is deliberately published rather than filed (the patent gate on this thread returned do-not-file, four for four — the contribution has the shape of a design rule and a measurement, which are unpatentable and unenforceable, so priority + credibility + defensive disclosure is the correct form of protection).

Abstract

Analog in-memory compute (AIMC) accelerates static-weight matmuls and cannot touch the dynamic, growing KV cache of attention. For a transformer this means AIMC's decode-time advantage dilutes with context length — from ~3.9× at 2k tokens to ~1.05× at 131k — because the KV-bound attention path comes to dominate system energy. Replacing attention with a static-weight sequence model (SSM/Mamba-class) removes the KV cache and restores a context-independent ~11.8× compute advantage, but pure-SSM models fail arbitrary-content recall, so production systems use SSM + attention hybrids. The central result of this note is that the attention *window policy* (sliding-window vs. full), not the attention *fraction*, decides whether the hybrid keeps AIMC's advantage. A full-attention hybrid — even at the ~8–12% attention fraction of quality-validated designs (Nemotron-H, Jamba) — forfeits the

win at exactly the long contexts the hybrid was built for (down to 1.2–2.3× at ≥32k). A sliding-window (Samba-class) hybrid holds ~5.6–6.8× constant in context length. We confirm the quality precondition on a real 2-billion-parameter checkpoint: RecurrentGemma-2B recalls an arbitrary passkey with probability 1.0 inside its 2048-token window and 0.0 beyond it, with the usable recall range measurably *tighter* than the window. We give the accompanying capacity result (fabricated AIMC is ~22× less weight-dense than SRAM, placing the near-term lane at 1–8B edge models, not 70B datacenter) and the boundary of hardware-aware training (it protects recall the architecture *has*; it cannot create recall the architecture *lacks*).

---

1. The design rule

> AIMC's decode advantage on a hybrid language model is a function of the attention window. Choose the window ≥ the workload's recall range, and prefer a *sliding-window* (Samba-class) hybrid over a *full-attention* (Jamba/Nemotron-class) hybrid. AIMC's win = win(W): it is large and context-independent for a bounded window and collapses toward unity for full attention at long context.

Sharpened, and with the one caveat the real-checkpoint data forces:

- It is the window policy (sliding vs. full), not the attention fraction, that decides the win. An 8%-attention hybrid with *full* attention loses the advantage; a 12%-attention hybrid with a *4k sliding window* keeps it.
- "Window ≥ recall range" is necessary but not sufficient. Measured on real hardware weights, the effective recall range at long context is tighter than the window (~0.5·W for reliable recall at deployment length L = 2·W). Size the window with margin over the workload's recall range, and let the margin grow with deployment context length.

The rest of this note is the evidence that makes each clause of the rule measured rather than asserted.

---

2. Why the window is the lever: the dilution curve (Amdahl in watts)

AIMC replaces static-weight matmuls with an O(1)-energy crossbar read; it cannot store or stream the per-token KV cache, which stays on a digital/attention path. So the achievable system speedup is Amdahl-bounded by the *fraction* of decode energy that is static-weight matmul — and that fraction shrinks as context grows and KV dominates.

Measured decode-energy advantage of AIMC over digital, by context length (A-series, `run_aimc.py`):

Context L	AIMC + transformer	AIMC + pure SSM
2k	3.86×	11.8×
8k	1.77×	11.8×
32k	1.20×	11.8×
131k	1.05×	11.8×

The transformer's advantage dilutes to nothing; the SSM's is flat in L because it has no KV cache (M-series, `run_ssm_aimc.py`; M1/M2). The mechanism: a Mamba-class layer's input-dependent selectiv-

ity is only 0.37% of layer compute (M3) — the large matmuls are static and AIMC-native, while attention is dynamic-weight and KV-bound and thus AIMC-hostile. AIMC's clean niche on a transformer is therefore prefill, not decode: 19.1× at prefill length 512, 9.3× at 2048 (A3), because prefill is compute-bound with no KV re-read.

That prefill niche has since been put through its own pre-registered kill tests and is fully bounded (K-series, `run_prefill_flank.py`, `docs/DIRECTION-prefill-flank.md`): the win dilutes with prompt length by causal-attention growth (5.9× at 4k, 3.6× at 8k, 1.7× at 32k — K3); it exists only in power-capped deployments, since digital wins the silicon-bound metrics (K1, and K2 below); the KV a quantized analog prefill box *produces* is portable to a clean digital decoder with no measurable penalty at ≥5-bit on a real 2.8B checkpoint (K4 — the disaggregated-prefill product clears its quality gate, with the untrained cliff at 5→4 bits being precisely what hardware-aware training exists to move); and shipping that KV to the decode fleet costs ≤0.1% of prefill compute (K5 — disaggregation is energetically free).

This is why the architecture — specifically what carries long-range information — is the whole game for an analog deployment.

---

3. The win(W) curve: window policy, not attention fraction (the differential result)

The naive reading of §2 — "use a pure SSM for 11.8× at any length" — fails the quality precondition (§4): pure SSMs cannot do arbitrary recall, so real systems interleave attention. The question that decides the hardware win is then: *how* is that attention configured?

AIMC system win vs. an honest digital-hybrid-at-the-same-attention-fraction baseline (H-series, `run_hybrid_gates.py`; H1/H2):

Context L	f=0 (pure SSM)	f=0.077 (Nemotron-H, 4/52 layers)	f=0.125 (Jamba, 1:7)	f=1 (transformer)
2k	11.8×	8.8×	7.8×	3.9×
32k	11.8×	2.28×	1.86×	1.2×
131k	11.8×	1.35×	1.23×	1.05×
512k	11.8×	1.09×	1.06×	1.01×

A hybrid with as little as 8% full attention collapses to 1.2–2.3× at ≥32k: the KV of even a few full-attention layers dominates once the SSM layers are made cheap. The attention *fraction* is small; the attention *policy* is what costs the win.

Replace those full-attention layers with a 4k sliding window (Samba-class) and the win becomes constant in L: 6.8× at f=0.077, 5.6× at f=0.125, because windowed KV is bounded. An independent analytical cost curve (W3, `run_window_range.py`, f=0.125, L=131k) reproduces the same law by window size:

Window w	AIMC win (f=0.125, L=131k)
4k	5.63×
16k	2.60×

Window w	AIMC win (f=0.125, L=131k)
64k	1.44×
full	1.23×

(W3 is a pure cost-model result; the toy W1/W2 trainability control failed its guard, so no *quality* claim was banked from the toy — that gap is what §4 fills with a real checkpoint. The full-attention endpoints of W3 and H1/H2 agree to rounding: 1.23× at f=0.125, L=131k.)

Design consequence: the deployable target for analog hardware is an SSM backbone + sliding-window attention (Samba-class), not an SSM + full-attention hybrid (Jamba/Nemotron-class). This is the item the prior-art survey (§7) found *not stated anywhere*.

---

4. The quality precondition, measured on a real 2B checkpoint

The win(W) curve is a cost model. Whether a bounded window is *usable* depends on whether the recurrent path can substitute for attention on arbitrary recall — a quality question the toy experiments could not settle. We measured it directly on a pretrained checkpoint (D4, `run_d4_checkpoints.py`).

Model: RecurrentGemma-2B (Griffin architecture, sliding-window = 2048, trained context 8k). Task: single arbitrary passkey (the hardest recall case — high-entropy, no semantic shortcut), n=10 per cell, ±~0.15. Distance = tokens from needle to query.

Beyond-window cells (decisive — past the window only the recurrent state can carry the key):

Context L	needle distance	beyond window?	recall
4096	3686	yes	0.0
4096	2048	boundary	0.6
4096	1024	no	0.5
4096	409	no	1.0
8192	7372	yes	0.0
8192	4096	yes	0.0
8192	2048	boundary	0.0
8192	819	no	1.0

Within the window and within trained context, recall is 1.0 everywhere (all L=1024 and L=2048 cells, every depth). Beyond the window it is 0.0, uniformly. The recurrent path carries nothing retrievable of an arbitrary passkey.

Two refinements matter for the rule:

1. The edge is graded, and moves inward with context length. At the *same* needle distance of 2048 (exactly the window edge), recall is 0.6 at L=4096 but 0.0 at L=8192 — doubling total context collapses edge recall to zero. Even well inside the window (distance 1024), recall falls from 1.0 at L≤2048 to 0.5 at L=4096.
2. Therefore the effective recall range is tighter than the window — here ~0.5·W for reliable recall at L=2·W. "Window ≥ recall range" is necessary but not sufficient; the margin must grow with deployment length.

Verdict (D4): *the attention window is the quality boundary for arbitrary-content recall on a real 2B Griffin checkpoint, and the boundary is, if anything, stricter than the cost model assumed.* This upgrades the window rule from toy-plus-cost-model to measured, and it retroactively validates the toy mechanism (pure-SSM recall failure) that the failed trainability control could not confirm.

Bounds (honest): single passkey (semantic retrieval may degrade more gracefully); one model family (Griffin/RecurrentGemma-2B); the within-model curve *shape* is the deliverable, not cross-model absolute comparisons (which stay training-data-confounded).

---

5. Where AIMC belongs: the capacity inversion (edge, not datacenter)

Two further campaign results bound *which* deployments the rule applies to.

Capacity (H4, D2). On fabricated silicon, compute-grade AIMC stores roughly 29k weights/mm² (HERMES-class PCM, Nature Electronics 2023) versus ~634k weights/mm² for SRAM (Groq-class) at 4-bit — SRAM is ~22× denser. A 70B model is therefore ~16,700 HERMES-class chips vs ~153 Groq-class (~110×). AIMC area is ADC/periphery-dominated, not cell-dominated, so the deficit does not close with denser cells alone. The density a durable win needs is named (D2): ~945k 4-bit weights/mm² at ENOB $\geq$ 4, matching 3nm SRAM-CIM (TSMC ISSCC'24, 3.78 Mb/mm²) — every published NVM-CIM macro is 15–33× short. Near-term AIMC lane = 1–8B edge/embedded models; 70B datacenter residency is a 3D-stacked-NVM roadmap bet. The honest competitor for the residency dividend is Groq (which already monetizes weight residency digitally at ~10× over GPU), not an HBM-streaming GPU.

The same inversion holds on the compute-throughput axis, now calibrated against fabricated silicon rather than guessed tile areas (K2, 2026-07-29): the best fabricated multi-core AIMC measures 0.40 TOPS/mm² (HERMES 64-core, 8-bit-I/O mode — itself >15× better than prior multi-core AIMC), with the single-core no-fabric best at 1.59, against ~4 TOPS/mm² for an INT8 digital MAC array (~1 full-chip at matched node). Digital wins tokens/s/mm² by ~2.5× at matched granularity and ~10× chip-vs-array. Both axes of silicon economics — weight density and compute density — favor digital; AIMC's case is J/token, which is why every lane in this note is a *power-bound* lane.

Frontier ceiling (D1, watts-Amdahl). For a dense 70B model, the static-weight MAC work AIMC strictly replaces is ~21 mJ/token, against a measured system energy of ~1–16 J/token. The addressable share is therefore 0.07–2.1% (pure MAC) to 18–30% (whole matmul pipeline), giving a system-win ceiling of $\leq$1.02× to 1.43× — even the charitable "replace the entire accelerator" fantasy caps at 2.5×. Frontier-datacenter inference is triple-gated, and D1 alone caps the win at ~1.4× regardless of density, quality, or noise. The window rule is an edge-lane rule.

---

6. The boundary of hardware-aware training

A natural extension is to push the analog operating point to its densest setting (2-bit ≈ halves ADC/area) and recover any lost quality with hardware-aware training. The campaign measured exactly where this works and where it stops (D5, `run_d5_qat_recurrence.py` and `run_d5_real_rung.py`).

- 4-bit and 3-bit are analog-robust even under readout noise; degradation appears only at 2-bit (the known ~4-bit ceiling).

- Hardware-aware training recovers the 2-bit gap — but by how much depends on scale. On a toy deployable (hybrid 1-of-4) it recovers 92–98% of the 2-bit recall loss across 3 seeds. On real checkpoints the same mechanism transfers but recovery is partial (~50%): 2-bit naive post-training quantization catastrophically destroys real weights (held-out top-1 → 0.003–0.006 from a ~0.96 ceiling), and QAT recovers about half (pythia-160m transformer 0.58; mamba-130m SSM 0.46 at reduced budget). The 92–98% (toy) → ~50% (real) correction is load-bearing and is published here rather than overstated.
- The hard boundary (pure_rec bound): hardware-aware training is a robustness method, not a capability method. Pure-recurrent recall is ~0.20 to begin with, so there is nothing to recover — *it protects the recall the architecture already has (the hybrid), and cannot grant recall the architecture lacks (pure SSM)*. This is why §3–§4 cannot be circumvented by training alone: the window/hybrid structure is what creates the recall; training only preserves it under low-bit analog noise.

Caveats: the toy is MQAR (D=64, 4 layers, linear-attention SSM proxy); the noise model is a convention (straight-through per-row-max quantization + Gaussian readout noise ∝ RMS), not device-measured; the regime is narrow (only 2-bit bites; a 4-bit Pisa-class hybrid at ENOB ~4.5 is already robust).

---

7. What is already known (prior art)

This note sits inside an active field led by IBM/Jülich and others. Stated plainly so the differential contribution (§8) is unambiguous:

- AIMC + SSM co-design and QAT — *published.* QS4D (arXiv:2507.06079) does QAT of SSM kernels for memristive crossbars and already reports the periphery/ADC-dominated cost structure; HPD (arXiv:2508.11935); a compute-in-memory SSM implementation (Nature Communications 2025, s41467-025-68227-w). Digital SSM accelerators: MARCA (arXiv:2409.11440), XAMBA (arXiv:2502.06924), Mamba-X (arXiv:2508.02977).
- Extreme quantization of SSMs — *published.* BitNet-class ternary weights; sub-4-bit SSM quantization with QAT recovery is an active line (referenced by name in the campaign; a specific Ternary/Bi-Mamba arXiv ID is *not* carried in the source repo and is not asserted here).
- Analog attention — *published.* Leroux et al., *Analog in-memory computing attention mechanism*, Nature Computational Science 2025 (arXiv:2409.19315): gain-cell KV storage + analog dot-products, 1.5B params, GPT-2-comparable. (Its headline energy figures are vs. a GPU baseline — apply the usual baseline caveat.)
- Hybrid SSM/attention architectures — *published.* Jamba (arXiv:2403.19887; Jamba-1.5 arXiv:2408.12570), Nemotron-H (NVIDIA, 2025), and the sliding-window variant Samba (arXiv:2406.07522). Samba's long-context *quality* claims are the load-bearing external dependency of the design rule and should be verified independently before betting on them.
- Hardware-aware training to 4-bit LLM parity — *published.* IBM Analog Foundation Models (arXiv:2505.09663): 4-bit parity via ~20B-token hardware-aware retrain. This is IBM's active bet; AIMC-for-LLM is not open ground.
- Silicon anchors: IBM HERMES 64-core PCM (Nature Electronics 2023, s41928-023-01010-1); Groq LPU (SRAM-resident, ~1–3 J/token); Tiki-Taka/TTv2 (AIMC *training*, research-stage, asymmetric-update blocker).

The ADC-dominance of AIMC energy/area, the feasibility of ~4-bit analog inference with retrain, and drift behavior are all confirmations of published results, not contributions.

8. Differential contribution

Against the prior art in §7, three items are not owned by it:

1. The window-policy rule (§3, H1/H2). *Full-attention hybrids forfeit the AIMC win at long context; the attention window policy (sliding vs. full), not the attention fraction, is what decides it.* The prior-art survey found this stated nowhere — it is a concrete, checkable design rule for anyone building hybrid models for analog hardware.
2. The capacity-inversion head-to-head (§5, H4). A direct fabricated-AIMC-vs-SRAM weight-density comparison (~22×) that relocates the near-term AIMC lane to 1–8B edge and names Groq, not GPUs, as the competitor to beat.
3. The dilution/regime curves (§2–§3, A2/A3/H1) and the real-checkpoint quality boundary (§4, D4) as the quantified Amdahl-in-watts story that ties the rule to a measured recall constraint at 2B scale.

Everything else here is confirmation of an active field.

9. Honest bounds

- Ratios, not absolutes. Internal per-token energies run ~250–750× below measured full-system figures (H6); only the *ratios* are meaningful. Every "×" above is a ratio against a same-assumptions baseline.
- Baseline choice is the biggest lever. The AIMC win is stated against an *honest digital-hybrid-at-the-same-attention-fraction* baseline. Against a weaker baseline (e.g. HBM-streaming) the numbers inflate; that inflation is not claimed.
- One model family for the quality boundary (Griffin/RecurrentGemma-2B), single-passkey recall (the hardest case). Semantic recall may be more forgiving.
- External dependencies not controlled here: Samba-class long-context quality (arXiv:2406.07522), and AIMC drift/endurance (IBM/Mythic's domain, assumed not solved here).
- The identified falsifier: gain-cell analog attention — quantified, and defused at demonstrated tech (T2, `run_gaincell_window.py`, arithmetic at the falsifier's own device parameters). The dilution mechanism (§2) is scoped to weight-stationary NVM crossbars, for which the KV cache is analog-hostile. Leroux et al. (§7) compute attention itself in analog with fast-write gain cells — but their own design is itself sliding-window (M=1024), which corroborates rather than threatens the rule, and demonstrated CMOS gain cells are window-bounded three ways: area (~8,590 mm^2 for full MHA at W=32k), standing refresh power that scales with W (~1.7 W at 32k), and 3-bit KV write precision. 66% of their attention energy is digital control — interface dominance again. The falsifier stays alive only three immaturities deep (ITO cell density + GQA + OSFET retention, jointly: 131k window in ~300 mm^2); if that corner matures, the window rule's *enforcement* moves from energy policy to device physics, but the rule itself survives at demonstrated tech.
- The rule is an edge-lane rule. Frontier-datacenter AIMC is separately gated (§5, D1) and is not addressed by the window policy.

10. Summary

For deploying language models on analog in-memory compute, the architecture decision that determines the hardware win is not "how much attention" but "what kind of attention window." Use an SSM backbone with sliding-window attention (Samba-class) and size the window with margin above the workload's recall range — the win is large and context-independent (~5.6–6.8×). Use full attention, even a little, and the win dilutes to ~1.2–2.3× at the long contexts that motivated the hybrid. The recall boundary is real and measured on a 2B checkpoint, and is tighter than the window itself. The lane is 1–8B edge models, where the compute share of system power is large enough for the win to survive, and where the competitor is Groq's digital residency, not a GPU. Hardware-aware training can hold this operating point down to 2-bit, but it preserves recall the architecture has; it does not create recall the architecture lacks.

Published by NeuroGenesys (AW3 Technology, Inc.). Canonical version: https://neurogenesys.si/research/aimc-window-codesign