Evermind: A Self-Updating, On-Device Cognitive Architecture Unifying Selective State-Space Generation, Write-Through Knowledge, and Trainable Affect

Sean Hogg

Builderforce.ai · Correspondence: seanhogg@gmail.com

Technical Report, 2026 · Reference implementation: `builderforce-memory` (engine / runtime / MCP), v2026.6.32

Abstract—Frontier large language models (LLMs) are *frozen*: their parametric knowledge is fixed at a training cutoff, updates require an expensive retrain cycle, and bolt-on retrieval grows an append-only store in which stale and current facts accumulate under distinct keys until reconciled by hand. We present **Evermind**, a cognitive architecture that treats currency, not scale, as the primary design axis. Evermind is organized as three cooperating layers mirroring a coarse neuro-functional decomposition: a *cortex* — a shared-expert hybrid selective state-space model (SSM) performing linear-time generation with on-device gradient updates; a *hippocampus* — a write-through knowledge memory governed by a reconciliation operator that *replaces beliefs on write* rather than appending them; and a *limbic* layer — a small trainable recurrent affect head that modulates generation. We formalize each layer. For the cortex we give the selective-scan recurrence, its zero-order-hold and exponential-trapezoidal discretizations, and the SSD and complex-valued variants, and we prove the recurrence is a monoid scan admitting an $O(L\log L)$-span parallel evaluation. For the hippocampus we define *Write-Through Cognition*, prove it maintains a single-incumbent invariant (no reconciliation backlog), and show a version-token recall cache yields $O(1)$ global invalidation on every belief replacement. The system runs entirely on WebGPU with zero runtime dependencies and exports to portable formats (safetensors, ONNX, GGUF, Hugging Face). We are explicit about validation status: the architecture, kernels, reconciliation algorithm, and export pipeline are implemented and tested (ONNX logit parity $<10^{-5}$ against the reference forward pass); the comparative *currency*, *footprint*, and *ownership* theses against frozen LLMs are stated as falsifiable hypotheses

with a measurement protocol, and are not yet empirically established. We invite replication and adversarial review.

Index Terms—State-space models, Mamba, selective scan, continual learning, retrieval-augmented generation, write-through cache, on-device inference, WebGPU, knowledge editing, affective computing.

I. Introduction

The dominant paradigm in language modeling couples a very large Transformer [1] with a fixed training corpus. This design buys breadth of capability at the cost of *temporal rigidity*: the model's beliefs are crystallized at a cutoff date, and the only sanctioned route to a new belief is a new training run. Retrieval-augmented generation (RAG) [2] mitigates the symptom by attaching an external store, but conventional stores are *append-only*: a corrected fact is added *alongside* the fact it corrects, and the contradiction is deferred to retrieval-time ranking or a manual reconciliation job. The model never truly *learns*; it accumulates.

This paper asks a different question. Instead of "how do we make the largest possible frozen model?", we ask "what is the smallest coherent system whose *knowledge is always current by construction*, that *owns* its own generation, and that *fits* inside the runtime where the work happens?" Our answer, **Evermind**, rests on three commitments: (1) linear-time, trainable generation via a selective SSM [3], [4] cheap enough to update online on the serving device; (2) write-through knowledge, where update is *replacement* keyed by a stable subject identifier, so reads are always current and there is no reconciliation backlog; and (3) on-device, dependency-free execution on WebGPU, exportable to portable formats.

We deliberately frame Evermind as a *systems and architecture* contribution. The thesis that an always-current small model can *outperform* a frozen frontier model on knowledge-sensitive tasks is attractive but, as of this writing, unproven; Section IX states it as a hypothesis with a test protocol rather than reporting a result we have not earned. What we *do* claim, and substantiate, is that the architecture is well-defined, internally consistent, implemented, and that its core algorithms have the formal properties we prove.

Contributions. (i) a formal specification of a three-layer cognitive architecture (Fig. 1) and its inter-layer contracts; (ii) a consolidated mathematical treatment of the hybrid selective-scan cortex (S6, SSD, complex ET) with a proof that the recurrence is a monoid scan (Prop. 1); (iii) *Write-Through Cognition*: a formal reconciliation operator, a single-incumbent invariant (Thm. 1), and an $O(1)$ version-token invalidation scheme (Prop. 3); (iv) a perplexity-aware inference router and an online distillation loop; (v) an honest validation account separating tested from hypothesized claims, with a reproducible protocol.

II. Related Work

State-space sequence models. Structured state-space layers [5] gave $O(L)$ sequence modeling; Mamba [3] made the dynamics input-*selective* (S6) with a hardware-aware parallel scan; Mamba-2 [4] recast the selective scan as structured state-space duality (SSD), exposing a matrix-multiply form. Evermind's cortex implements S6, SSD, and a complex-valued MIMO variant with exponential-trapezoidal discretization, plus optional attention [1] layers in a hybrid schedule.

Continual learning and knowledge editing. Continual learning fights catastrophic forgetting [6]; editing methods such as ROME [7] perform localized parametric edits. Evermind differs in *where* currency lives: factual currency is delegated to a write-through symbolic memory with explicit replacement semantics, while the parametric cortex adapts slowly via selective online distillation, sidestepping the credit-assignment fragility of editing weights per fact.

Retrieval augmentation. RAG [2], dense retrieval [8], rank fusion (RRF) [9], and diversity reranking (MMR) [10] are standard for recall. Evermind uses these but adds the missing *write* discipline: its store is reconciled, not appended.

On-device and affective modeling. We run SSM *training* (not just inference) on WebGPU with no ML-framework dependency. The limbic layer relates to affective computing [11]: personality is encoded as fixed setpoints, the limbic cell supplies bounded dynamics.

III. System Overview

A request x enters an *inference router* (Section VII) that decides whether to serve x from the on-device cortex or escalate to an optional frontier bridge. The cortex (Section IV) generates language; before and during generation it *recalls* from and *writes through* to the hippocampus (Section V); the limbic layer (Section VI) modulates the response. All three layers are differentiable and trainable on the serving device.

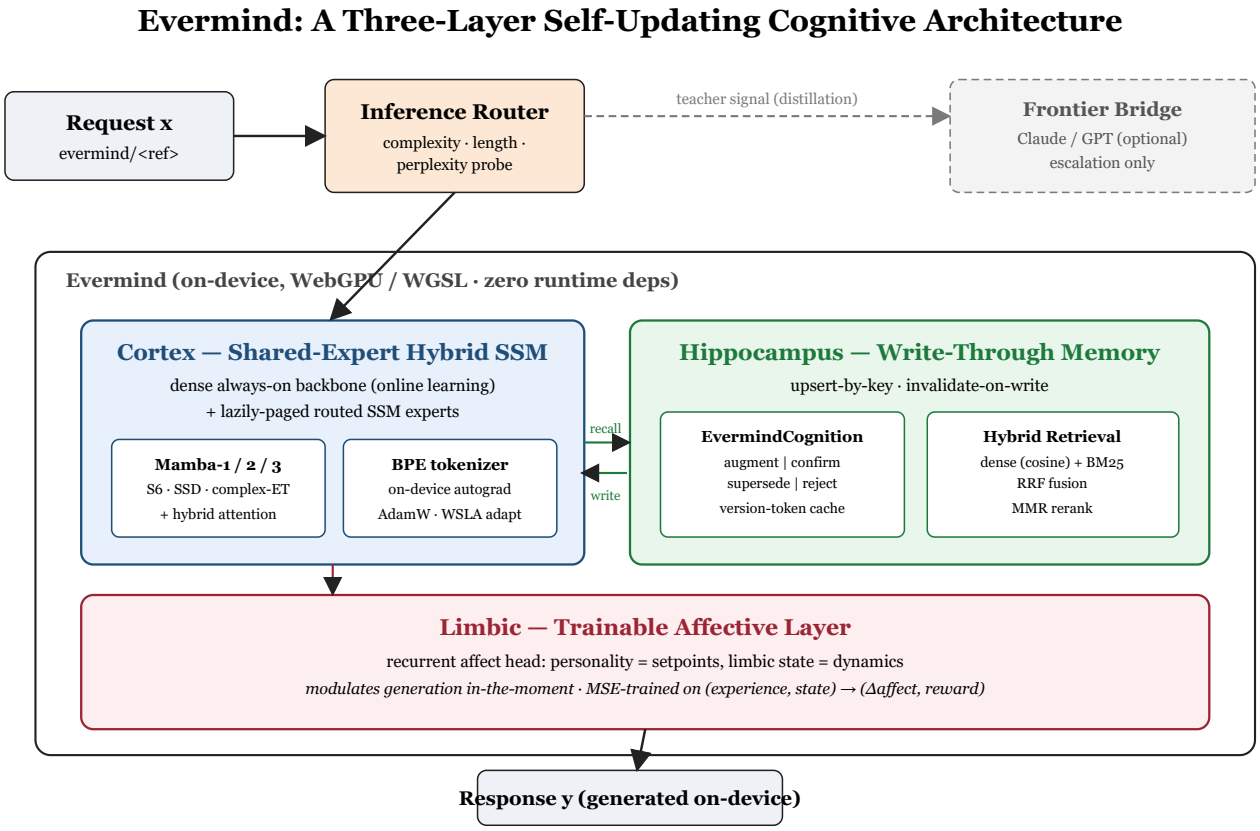

Fig. 1. Evermind's three-layer architecture. The cortex (shared-expert hybrid SSM) generates; the hippocampus (write-through memory) supplies and absorbs knowledge; the limbic layer modulates affect. The frontier bridge is an optional routing target and a distillation teacher, not a runtime dependency.

Notation. d is model width (`dModel`); $D=ed$ the expanded inner width with expansion e; N the SSM state dimension; L sequence length; H heads; K the causal convolution width. Default reference configuration: $d{=}512,\ e{=}2\Rightarrow D{=}1024,\ N{=}16,\ K{=}4,\ H{=}4,\ L_{\text{layers}}{=}8$.

IV. The Cortex: A Hybrid Selective State-Space Generator

A. Selective scan (S6)

A single SSM channel maintains $h_t\in\mathbb{R}^N$ under an input-dependent linear recurrence. The continuous system $\dot h=Ah+Bx,\ y=Ch$ is discretized per token with a selective step Δ_t. For stability A is stored in log-space as $a=\log(-A)$ so $A_{\text{cont}}=-\exp(a)<0$. Zero-order hold gives

$$\Delta_t = \operatorname{softplus}(\delta_t)=\log(1+e^{\delta_t}),$$ $$\bar A_t = \exp(\Delta_t A_{\text{cont}}),\qquad \bar B_t = \frac{\bar A_t-1}{A_{\text{cont}}}\,B_t,$$

and the discrete recurrence and readout are

$$h_t = \bar A_t\odot h_{t-1} + \bar B_t\,x_t,\qquad y_t = C_t^{\top}h_t + D\,x_t,$$

where $\odot$ is the Hadamard product and D a learned skip. *Selectivity*: Δ_t,B_t,C_t are projected from x_t, so the dynamics depend on content. These equations are exactly the kernel in `selective_scan.ts:5-71`.

B. The block

With input $u\in\mathbb{R}^d$ and a residual stream:

$$\tilde u=\operatorname{RMSNorm}(u)=\frac{u}{\sqrt{\tfrac1d\sum_i u_i^2+\epsilon}}\odot\gamma,\quad [x,z]=W_{\text{in}}\tilde u,$$ $$x'=\operatorname{SiLU}(\operatorname{Conv1d}_K(x)),\quad \operatorname{SiLU}(v)=v\,\sigma(v),$$ $$[\Delta,B,C]=W_x x',\quad y=\operatorname{SelectiveScan}(x',\Delta,B,C,D),$$ $$o=W_{\text{out}}\big(y\odot\operatorname{SiLU}(z)\big),\qquad u'=u+o.$$

The gate $\operatorname{SiLU}(z)$ is the standard gated-SSM nonlinearity; RMSNorm uses no mean subtraction. Tensor shapes follow `mamba1_block.ts:125-144`.

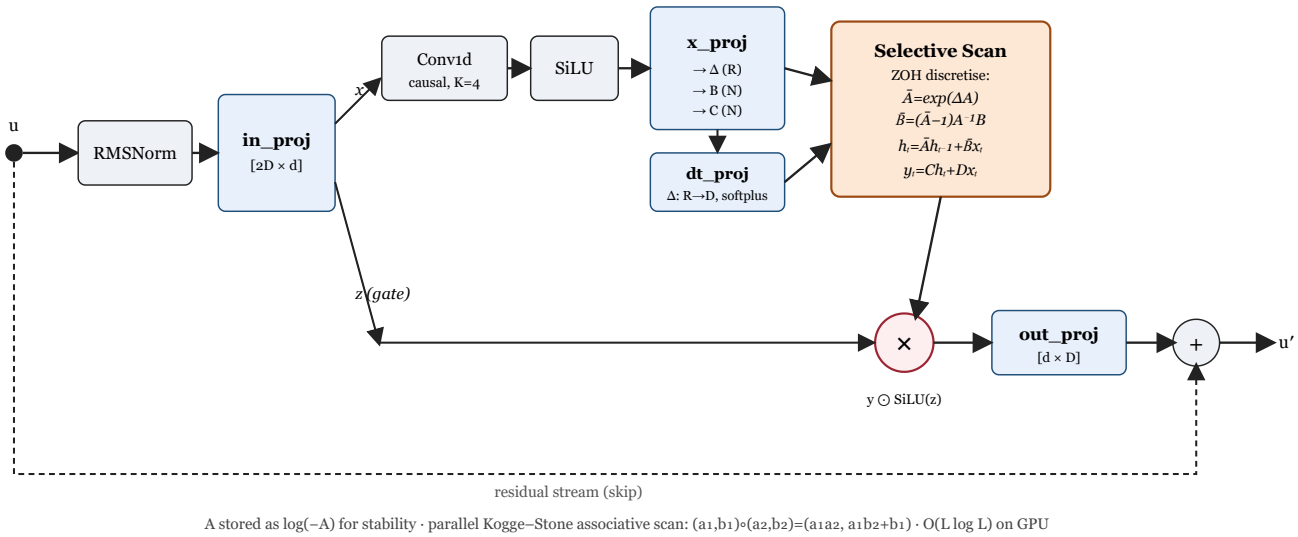

Fig. 2. Hybrid SSM block (S6). The selective projection produces the input-dependent (Δ,B,C) that drive the scan; A is stored as $\log(-A)$. A gated branch $\operatorname{SiLU}(z)$ modulates the scan output before down-projection and residual add.

C. Structured state-space duality (Mamba-2)

The SSD variant collapses A to one scalar *per head* with a chunked scan. With $\Delta_t=\operatorname{softplus}(\delta_t+\delta_{\text{bias}})$ and per-head log rate $A_{\log}$,

$$\bar A_t=\exp\!\big(-\operatorname{softplus}(A_{\log})\,\Delta_t\big),\qquad h_t=\bar A_t h_{t-1}+B_t x_t.$$

Because $\bar A_t$ is a scalar gate, a length-L chunk admits the dual matrix form $Y=(\mathcal L\odot CB^{\top})X$ with $\mathcal L_{ij}=\prod_{k=j+1}^{i}\bar A_k$ for $i\ge j$, else 0 (`ssd.ts:80,114`). Grouping B,C into G groups (default $G{=}1$) trades expressivity for memory.

D. Complex-valued MIMO with ET discretization (Mamba-3)

The Mamba-3 layer carries complex state $h_t\in\mathbb{C}^{N/2}$ (interleaved re/im), with $A=\exp(\rho+i\theta)$. The exponential-trapezoidal (ET) discretization uses the exact complex update

$$\bar A_t=\exp(\Delta_t\rho)\big(\cos(\Delta_t\theta)+i\sin(\Delta_t\theta)\big),$$
$$\bar B_t=(\bar A_t-1)A^{-1}B_t,\qquad y_t=\Re\big(C_t^{\top}h_t\big),$$

giving oscillatory eigenmodes (rotations on the unit circle scaled by $e^{\Delta\rho}$) a real diagonal A cannot represent (`complex_ssd.ts:70-85`).

E. The recurrence is a parallelizable monoid scan

> **Proposition 1 (Monoid scan).** Define on pairs $(a,b)\in\mathbb{R}\times\mathbb{R}$ the operator $(a_1,b_1)\circ(a_2,b_2)=(a_1a_2,\ a_1b_2+b_1)$. Then $\circ$ is associative with identity $(1,0)$, and the prefix $(\cdot,h_t)=(a_1,b_1)\circ\cdots\circ(a_t,b_t)$ with $a_t=\bar A_t,\ b_t=\bar B_t x_t,\ h_0=0$ satisfies the recurrence $h_t=\bar A_t h_{t-1}+\bar B_t x_t$.

Proof. Associativity: for three pairs, both bracketings give first component $a_1a_2a_3$ and second $a_1a_2b_3+a_1b_2+b_1$; identity is immediate. For the prefix, induct on t: the partial product equals $\big(\prod_{k\le t}a_k,\ \sum_{k\le t}(\prod_{j>k}a_j)b_k\big)$, whose second component is $a_t\sum_{k<t}(\prod_{j>k}a_j)b_k+b_t=a_t h_{t-1}+b_t=h_t$. □

> **Proposition 2 (Span–work).** The states $\{h_t\}_{t=1}^L$ can be computed by a Kogge–Stone inclusive scan in $\lceil\log_2 L\rceil$ parallel steps (span $O(\log L)$) and $O(L)$ work per (channel, state) pair.

Proof. Immediate from Prop. 1 and the standard parallel-prefix result for associative operators. The implementation tiles time into 64-lane workgroups dispatched over $(\lceil D/8\rceil,\lceil N/8\rceil,\text{batch})$ (`selective_scan.ts:74-146`). □

On commodity GPUs this realizes the $O(L)$-work, $O(\log L)$-span profile characteristic of selective SSMs, in contrast to the $O(L^2)$ work of dense attention.

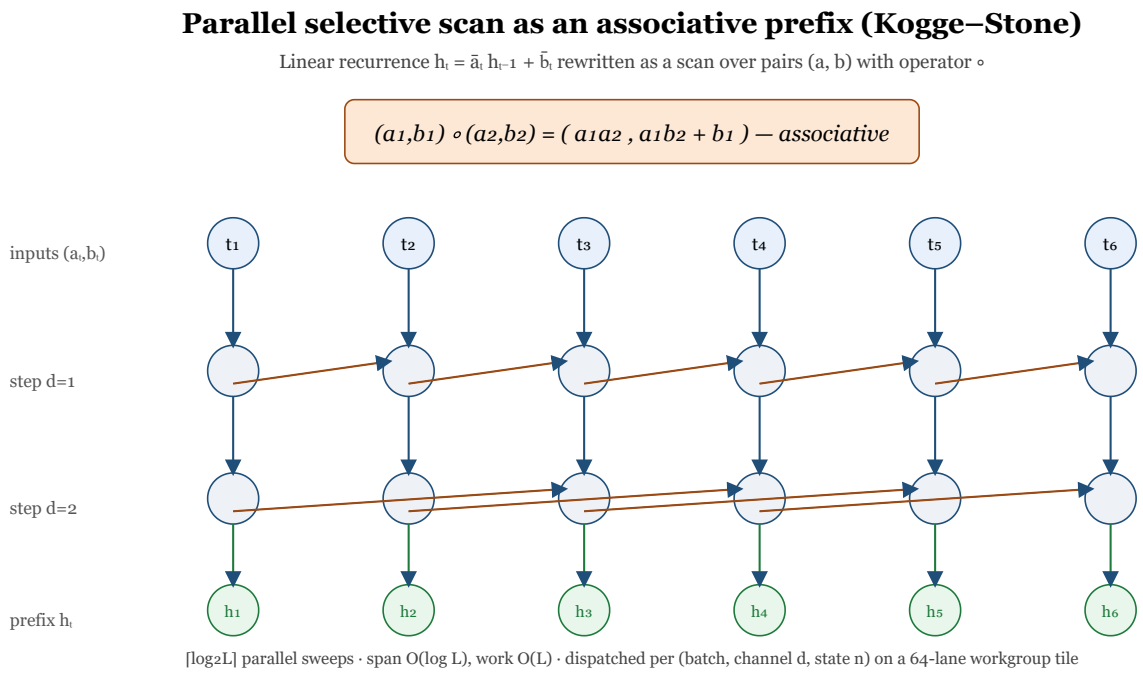

Fig. 3. The selective recurrence as an associative prefix over pairs \((a,b)\). \(\lceil\log_2 L\rceil\) sweeps yield all prefix states.

F. On-device training and selective fast-adaptation

A tape-based reverse-mode autograd (`autograd.ts`) records each forward op as a closure replayed in reverse. The loss is token cross-entropy

$$\mathcal L=\frac1L\sum_t\Big(\log\textstyle\sum_v e^{z_{t,v}}-z_{t,y_t}\Big),$$

optimized with decoupled-weight-decay AdamW [12] on the GPU (`weight_update.ts`):

$$m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t,\quad v_t=\beta_2 v_{t-1}+(1-\beta_2)g_t^2,$$ $$\theta_t=\theta_{t-1}(1-\eta\lambda)-\eta\,\frac{\hat m_t}{\sqrt{\hat v_t}+\epsilon},$$

with bias-corrected moments, defaults \(\eta{=}10^{-4},\beta_1{=}0.9,\beta_2{=}0.999,\lambda{=}0.01\), global-norm clipping at \(1.0\). **Weight-Selective Layer Adaptation (WSLA)** restricts online updates to the selective-projection rows emitting \(B,C\) — the \(2GN\) rows of \(W_{\text{in}}\) that route content into state — freezing the bulk representation (`mamba2_block.ts:299–309`). This makes the distillation loop (Section VIII) cheap enough to run in a few epochs on-device.

V. The Hippocampus: Write-Through Cognition

Caching keeps *answers* fresh; Evermind's hippocampus keeps *knowledge* fresh. We formalize it as a write-through cache with an explicit conflict resolver over beliefs.

A. Beliefs, keys, and the store

> **Definition 1 (Belief and store).** A *belief* is a pair $b=(k,c)$ with stable subject key $k\in\mathcal K$ (from a canonicalizer, not a per-write id) and content c, carrying importance $\iota(b)\in[0,1]$. A *store* $\Sigma:\mathcal K\rightharpoonup\mathcal C$ is a partial map holding at most one incumbent content per key.

B. The reconciliation operator

Every candidate belief passes through one pipeline (Fig. 4): *canonicalize → recall incumbent → evaluate evidence → reconcile → write-through*. Let $\sigma=\Sigma(k)$ be the incumbent (possibly $\bot$) and $\varepsilon\in\{\textsc{supportsNew},\neg\textsc{supportsNew}\}$ an evidence verdict. The operator V returns:

$$V(k,c,\sigma,\varepsilon)=\begin{cases} \textsf{augment}, & \sigma=\bot,\ \iota\leftarrow0.6\\ \textsf{confirm}, & \sigma=c,\ \iota\leftarrow\min(1,\iota+0.1)\\ \textsf{supersede}, & \sigma\neq c,\ \varepsilon=\textsc{supportsNew},\ \iota\leftarrow\max(0.9,\iota)\\ \textsf{reject}, & \sigma\neq c,\ \varepsilon=\neg\textsc{supportsNew} \end{cases}$$

with write rule $\Sigma'(k)=c$ if $V\in\{\textsf{augment},\textsf{supersede}\}$, else $\Sigma'(k)=\sigma$. This is `EvermindCognition.commit()` (`EvermindCognition.ts:80–116`).

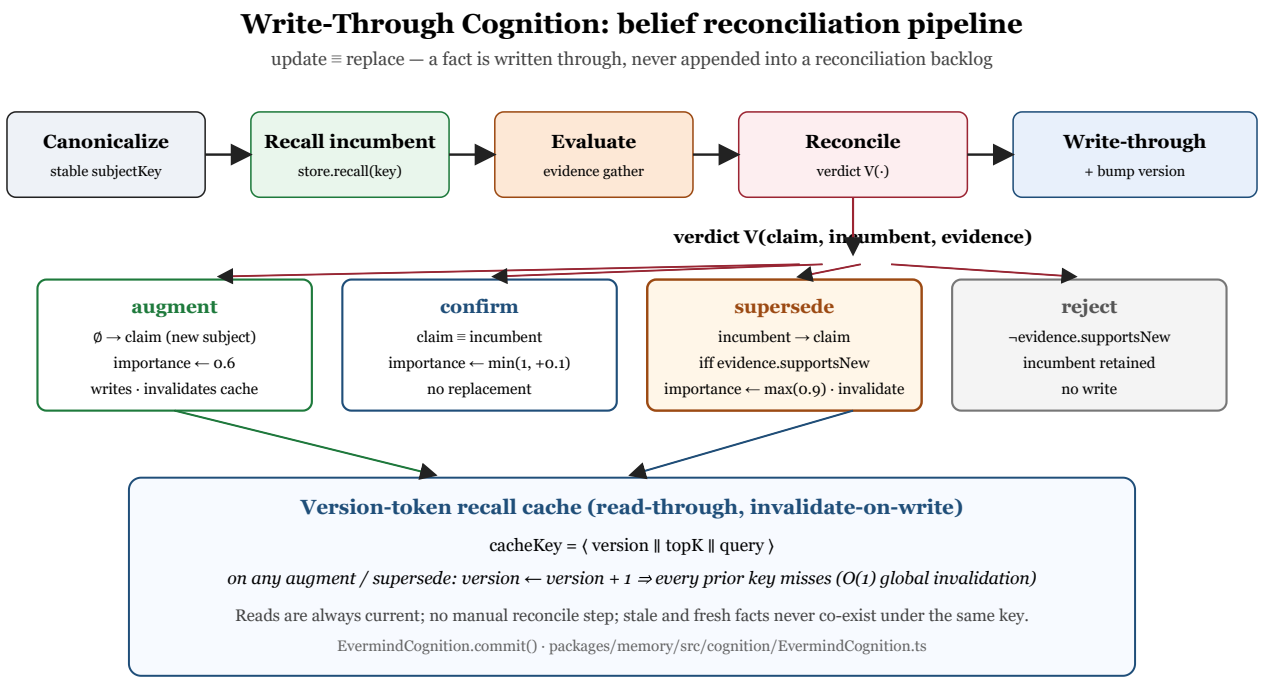

Fig. 4. Write-Through Cognition. Each candidate fact is canonicalized to a stable subject key, reconciled against the single incumbent, and written through; *augment* and *supersede* bump a global version token that invalidates the recall cache in $O(1)$.

C. Single-incumbent invariant

> **Theorem 1 (No reconciliation backlog).** For any finite sequence of commits applied to an initially empty store, at every step Σ holds at most one content per key, and for each k the retained content is that of the most recent *augment*/*supersede* on k (or $\bot$). Consequently no two contents are ever simultaneously held under the same key.

Proof. Σ is a partial *function*, so it maps each k to at most one content. Induct on commits: the empty store satisfies the claim vacuously; the write rule only overwrites $\Sigma(k)$ (augment/supersede) or leaves it unchanged (confirm/reject), so after each step $\Sigma(k)$ equals the content of the last overwriting commit on k. No append exists, hence two contents never co-exist under one key. □

This is the precise sense in which Evermind "corrects in place": unlike an append-only RAG store, a superseded fact is *gone*, not merely outranked, so retrieval cannot resurface it.

D. Version-token recall cache

> **Proposition 3 ($O(1)$ global invalidation).** Let the recall cache key be $\kappa=\langle\nu\parallel\text{topK}\parallel q\rangle$ with global version $\nu\in\mathbb N$ incremented on every augment/supersede. A single increment $\nu\leftarrow\nu+1$ invalidates *all* previously cached recalls in $O(1)$ time and space, without per-entry traversal.

Proof. After the increment, every previously stored key embeds the stale token $\nu-1\neq\nu$, so no subsequent lookup (embedding ν) matches a stale entry; stale entries are never read again and are reclaimed lazily. The counter update is constant-time. □

This is `_bumpVersion()` and the namespaced `recall()` cache (`EvermindCognition.ts:54-145`). The cache never serves a recall predating the most recent replacement: *reads are always current*.

E. Hybrid recall

Retrieval (Fig. 5) fuses a dense and a sparse ranker. Dense similarity is cosine over L_2-normalized SSM embeddings,

$$\operatorname{sim}(q,d)=\frac{\langle q,d\rangle}{\lVert q\rVert\,\lVert d\rVert}\in[-1,1],$$

(`similarity/index.ts:32-42`; a Jaccard fallback is used when embeddings are absent). Sparse ranking is Okapi BM25,

$$\operatorname{BM25}(q,d)=\sum_{t\in q}\operatorname{idf}(t)\frac{f_{t,d}(k_1+1)}{f_{t,d}+k_1\big(1-b+b\frac{|d|}{\overline{|d|}}\big)},$$

$\operatorname{idf}(t)=\log\!\big(1+\frac{N-n_t+0.5}{n_t+0.5}\big)$, $(k_1{=}1.5,\ b{=}0.75)$. The rankings merge by reciprocal rank fusion and diversify by maximal marginal relevance,

$$\operatorname{RRF}(d)=\sum_\ell\frac{w_\ell}{k+\operatorname{rank}_\ell(d)},\ k=60;\qquad \operatorname{MMR}(d)=\lambda\operatorname{sim}(q,d)-(1-\lambda)\max_{s\in S}\operatorname{sim}(d,s),\ \lambda=0.7.$$

Recall returns a hard-capped top-\(K\) (default \(5\)) with truncated content, so memory *lowers* rather than inflates prompt size — a deliberate token-economy property.

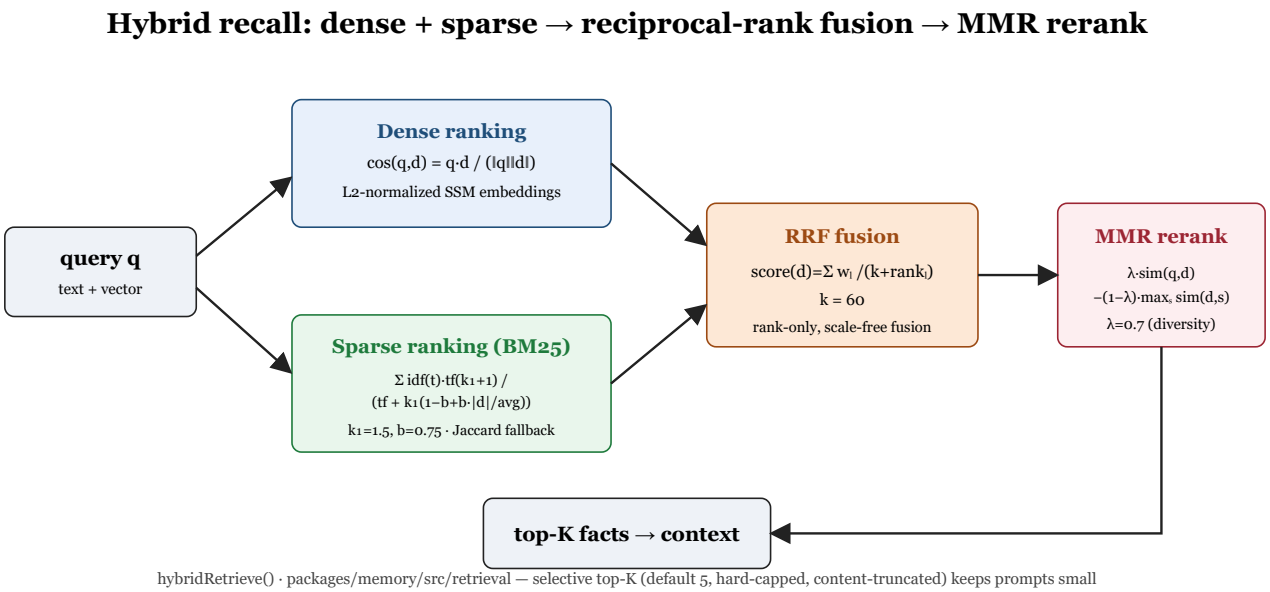

Fig. 5. Hybrid recall: cosine dense ranking and BM25 sparse ranking fused by RRF and diversified by MMR.

VI. The Limbic Layer

The limbic head (Fig. 6) is a small gated recurrent cell mapping an experience embedding \(x\in\mathbb R^{32}\) and affective state \(s\in\mathbb R^{8}\) to a bounded affect delta and a scalar reward estimate. With hidden \(h\in\mathbb R^{16}\) and per-channel gate \(a=\sigma(A)\),

$$\text{pre}=W_x x+W_s s,\quad h'=a\odot h+(1-a)\odot\tanh(\text{pre}),$$
$$\Delta s=\tanh(W_o h'+b_o)\in(-1,1)^8,\quad \hat r=w_r^\top h'+b_r.$$

The update is a gated leaky integrator: \(a\) controls how much prior affect persists versus how much new experience is admitted (`limbic_model.ts:14-18`). *Personality* is encoded as fixed setpoints (a persona's baseline \(s\)); the limbic cell supplies the dynamics *around* those setpoints. Training minimizes MSE on observed \((\Delta s,r)\) targets with truncated BPTT(1) and AdamW (\(\eta{=}0.05\)) (`limbic_trainer.ts:107-150`).

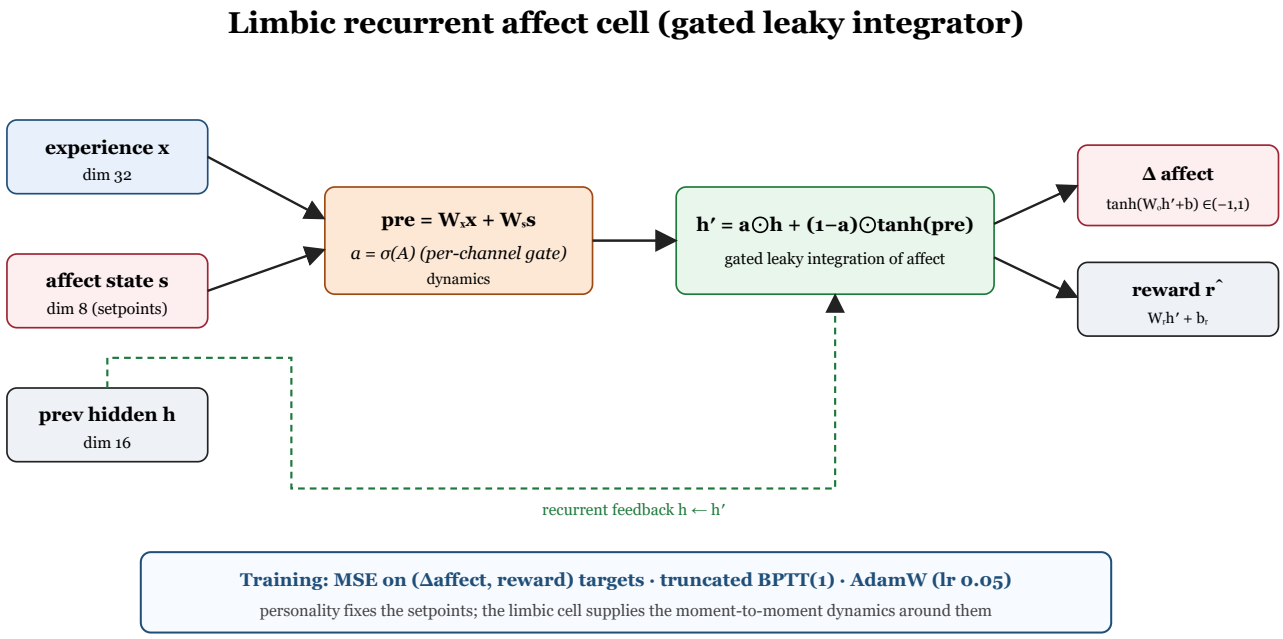

Fig. 6. The limbic recurrent affect cell: a gated leaky integrator producing a bounded affect delta and a reward estimate.

VII. Inference Routing

The router (Fig. 7) decides between on-device SSM generation and optional frontier escalation by a cheapest-first cascade (`InferenceRouter.ts:149-200`): (1) no bridge → serve from SSM, confidence \(1\); (2) honor a fixed strategy if set; else (3) escalate on a syntactic complexity pattern (“analyze”, “compare”, “step by step”), confidence \(0.9\); (4) escalate if \(|x|>1200\) chars, confidence \(0.85\); (5) run an optional perplexity probe, escalate if SSM perplexity exceeds \(\tau{=}80\) with confidence \(\min(0.95,\ 0.5+(\text{ppl}-\tau)/200)\). The default terminal is the SSM. Because the costly probe runs last and only when cheap predicates are inconclusive, expected routing cost is dominated by string predicates. \(\tau\) is a *tuning* threshold, not a measured benchmark.

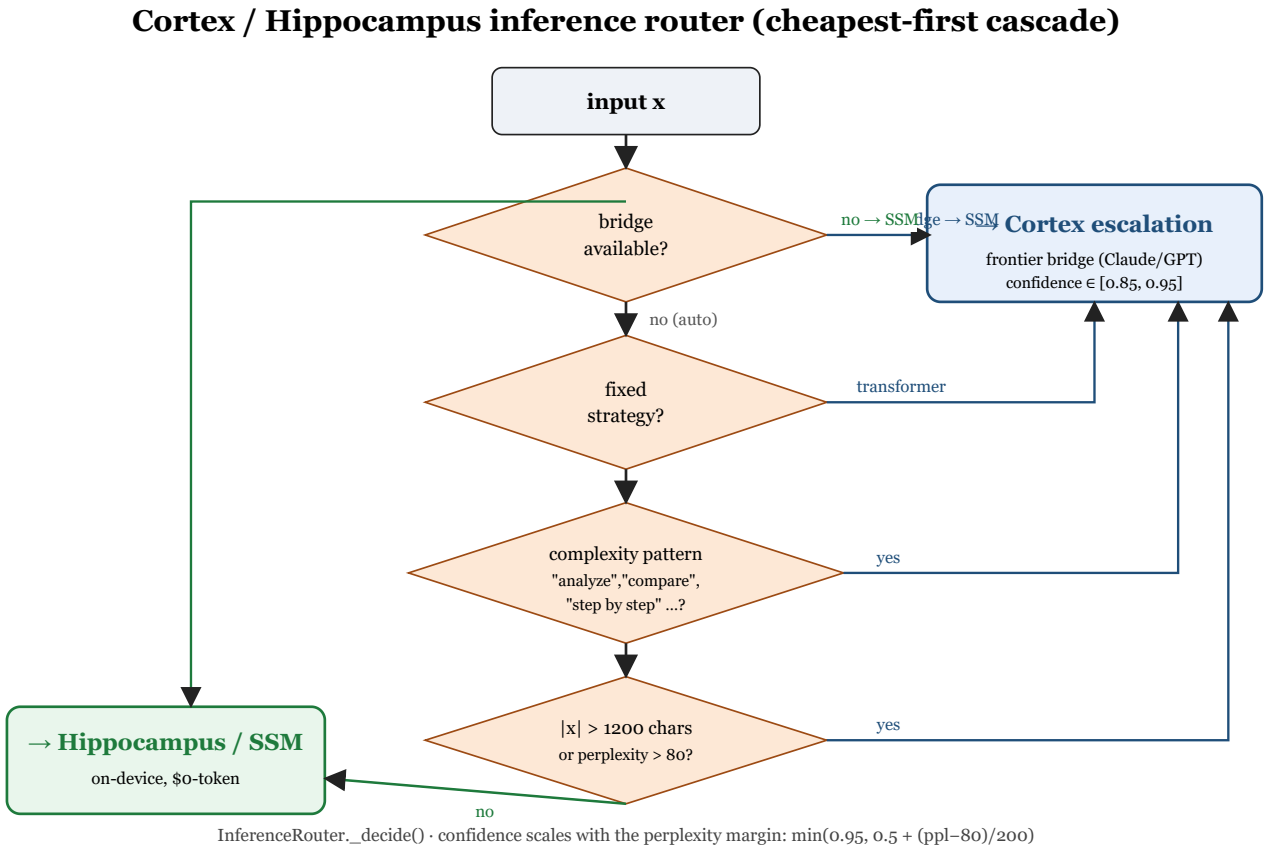

Fig. 7. The cheapest-first routing cascade. The SSM is the default terminal; the frontier bridge is reached only when cheap predicates or a perplexity probe indicate the on-device model is out of distribution.

VIII. Online Distillation

When the router escalates, the frontier response is treated as a teacher signal that adapts the cortex (`DistillationEngine.ts:117-209`). Given prompt x and teacher output $\hat y=\text{Teacher}(x)$, the student trains on $x\Vert\hat y$ under the LM objective with WSLA enabled and a few epochs (default 3). Two quality gates protect the update: a minimum-length gate (skip degenerate output) and a *maxPerplexity* gate that skips training when the SSM's perplexity on $\hat y$ is already below threshold — i.e. already learned. Adapted weights persist to disk/IndexedDB checkpoints, closing an online learning loop without a separate retrain stage. Convergence is characterized qualitatively here; quantitative curves are future work (Section IX).

IX. Validation Status and Evaluation Protocol

We separate, deliberately, *what is implemented and tested* from *what is hypothesized*. Conflating the two is the failure mode this section exists to prevent.

A. What is established

Component	Status	Evidence
S6 / SSD / complex kernels	implemented	forward/backward + ONNX parity \ (<10^{-5}\)
Autograd + AdamW + WSLA	implemented	training tests
Write-Through reconciliation	implemented + proven	Thm. 1, Prop. 3
Hybrid recall (cos/BM25/RRF/MMR)	implemented	retrieval tests
Limbic cell	implemented	MSE training harness
Router cascade	implemented	unit tests
Online distillation loop	implemented	integration path
Export (safetensors/ONNX/GGUF/HF)	implemented + tested	round-trip tests
Currency vs. frozen LLM	**hypothesis**	protocol §IX-C
Quality/perplexity vs. frontier	**hypothesis**	protocol §IX-C
Distillation convergence curves	**future work**	—
Affective-behavior validity	**future work**	—

B. Falsifiable hypotheses

H1 (Currency). On a stream of time-stamped factual updates, an Evermind hippocampus answers post-update queries with strictly lower *staleness* than an append-only RAG baseline of equal retrieval budget, and than a frozen LLM. **H2 (Footprint).** The stack sustains interactive generation on commodity WebGPU within a memory budget an order of magnitude below a frontier-class served model, at a stated quality operating point. **H3 (Adaptation).** WSLA online distillation reduces

SSM perplexity on teacher-distribution prompts monotonically over epochs without catastrophic degradation on a held-out general set.

C. Measurement protocol

H1: build a temporal knowledge benchmark of $(k,\ c_{\text{old}}\!\to c_{\text{new}},\ t)$ edits; after each edit, query the subject and m paraphrases; report *staleness rate*, *contradiction rate* (two live answers disagree), and *edit latency*, comparing Evermind write-through, append-only dense RAG, and a frozen LLM. Theorem 1 predicts an Evermind contradiction rate of 0 by construction — a directly falsifiable claim. **H2:** report peak GPU memory, tokens/s, and time-to-first-token at matched perplexity on held-out text across widths $d\in\{256,512,768\}$. **H3:** report per-epoch teacher-prompt perplexity and held-out general perplexity for WSLA vs. full fine-tune. All three are reproducible from the open packages; until run, H1–H3 remain *conjectures* and we make no comparative performance claim.

X. Discussion, Limitations, and Threats to Validity

The most important limitation is the one Section IX foregrounds: the comparative advantages that motivate Evermind are *architecturally plausible and partly provable* (the zero-contradiction property follows from Theorem 1) but *not yet empirically measured at scale*. Specific threats: (i) recall quality depends on embedding quality, and the headless path falls back to lexical overlap when SSM embeddings are absent; (ii) the canonicalizer producing subject keys is itself a model and a source of error — a mis-canonicalization splits or merges subjects and can defeat the single-incumbent invariant at the *key-assignment* boundary even though the store-level invariant holds; (iii) WSLA trades adaptation capacity for speed, and its sufficiency is empirical; (iv) WebGPU availability and driver variance bound portability. We regard these as the agenda, not objections to the architecture's coherence.

XI. Conclusion

We presented Evermind, a three-layer cognitive architecture in which a linear-time selective state-space cortex generates, a write-through hippocampus keeps knowledge

current by construction, and a trainable limbic layer modulates affect, all on-device with zero runtime dependencies. We gave a unified mathematical account of the SSM cortex, proved its recurrence is a parallelizable monoid scan, and formalized Write-Through Cognition with a single-incumbent invariant and \(O(1)\) cache invalidation. We separated proven and implemented properties from the performance hypotheses that remain to be tested, and supplied a protocol to test them. We hope the formalization and open implementation make Evermind a useful object of study — and an easy target for falsification — for the community.

References

1. A. Vaswani et al., "Attention is all you need," *NeurIPS*, 2017.
2. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," *NeurIPS*, 2020.
3. A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv:2312.00752*, 2023.
4. T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," *ICML*, 2024.
5. A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *ICLR*, 2022.
6. J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *PNAS*, 2017.
7. K. Meng et al., "Locating and editing factual associations in GPT," *NeurIPS*, 2022.
8. V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," *EMNLP*, 2020.
9. G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," *SIGIR*, 2009.
10. J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," *SIGIR*, 1998.
11. R. W. Picard, *Affective Computing*. MIT Press, 1997.
12. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *ICLR*, 2019.

Reproducibility: all equations cite source files in the open `builderforce-memory` package family (engine/runtime/MCP), v2026.6.32. Figures are vector SVG. The evaluation-protocol skeleton (§IX-C) accompanies this report.