WHY DO SNNS CONVERT BUT NOT TRAIN?

Anonymous authors
Paper under double-blind review

ABSTRACT

Why does a spiking neural network (SNN) converted from a trained Transformer nearly retain its accuracy, but one trained from scratch fall behind a Transformer of the same size? If spikes set the limit, conversion and its long integration windows are the better route; if training sets it, the limit has to be located first. We study this question through the release identity of the spiking neuron. At the non-leaky endpoint of conversion a neuron integrates linearly, whereas in the leaky regime of training its release depends on when past inputs arrived, a dependence to be learned. By the chain rule, a past token reaches the prediction, and its gradient the parameters, only through four links, in which it is written into a membrane, released, read out and credited. For each link we derive why gradient training leaves it closed and which temporal prior opens it, and we build these priors into a spiking language model, NeuronSpark. Trained from scratch at four scales, NeuronSpark reaches the level of mainstream architectures of the same size in training loss and BabyLM evaluations. Register-transfer-level estimates give it the lowest energy per token, with an advantage that grows with scale.

1 INTRODUCTION

Spiking neural networks (SNNs) are receiving growing attention as a basis for energy-efficient sequence models (Roy et al., 2019; Schuman et al., 2022). A spiking neuron keeps its history in a membrane potential and emits an event only when that potential crosses a threshold, so the work per token does not grow with the context and silent neurons cost nothing on event-driven hardware (Merolla et al., 2014; Davies et al., 2018). Both properties suit language models with long contexts. A spiking language model can be built through two routes. One converts a trained artificial neural network (ANN), usually a Transformer, by replacing its activations with spiking neurons (Rueckauer et al., 2017; Bu et al., 2022); the other trains the spiking network directly from scratch with surrogate gradients (Neftci et al., 2019; Zhu et al., 2024).

The two routes lead to opposite outcomes. Converted spiking Transformers, language models included, come within a small margin of their source networks once each neuron integrates over enough time steps (You et al., 2024; Hwang et al., 2024; Chen et al., 2026), whereas spiking language models trained from scratch remain behind Transformers of the same size (Zhu et al., 2024; Xing et al., 2024). The neurons are the same in both cases, so the gap cannot come only from what spiking neurons can represent. If the gap is read as a limit of spikes, direct training gives way to conversion and its long integration windows, which cancel the latency and energy advantage; if it is read as an optimization detail, components are added by trial and error without knowing which part of the network they repair. Why a spiking network that converts into an accurate model cannot learn the same function from data therefore remains an open question.

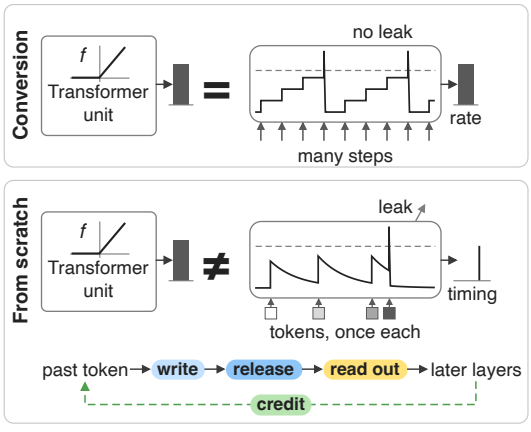


Figure 1: **Same neuron, two operating points.**

Existing analyses describe each route at a single operating point of the neuron. Conversion analyses decompose the error of a converted network into clipping, quantization and unevenness of firing rates over a long window (Deng & Gu, 2021; Li et al., 2021; Bu et al., 2022), which presumes that a neuron's output is a function of its input rate. Analyses of direct training study how surrogate

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.

gradients pass the threshold (Zenke & Vogels, 2021; Gygax & Zenke, 2025) and why gradients vanish along long recurrences (Bengio et al., 1994; Pascanu et al., 2013), which presumes that the network can already express the temporal dynamics it needs. Neither line relates the two routes through the same object, so neither says what the network trained from scratch is missing.

We relate the two routes through the release identity of the spiking neuron, an exact account of how much a neuron releases given its input and its leak. At the non-leaky endpoint that conversion uses, the identity makes the neuron a linear integrator over its window, so a converted network only has to match firing rates. A network trained from scratch works away from this endpoint, where each membrane is updated once per token and leaks, so what a neuron releases depends on when past inputs arrived, and each temporal dependency has to be learned (Figure 1). Applying the chain rule to the network, we show that a past token reaches the prediction, and its gradient the parameters, only if it is written into a membrane, released through the spiking nonlinearity, read out by a later layer, and credited back. Each of these four links is a factor of one product, so a single closed link removes the token from both the prediction and the gradient.

The answer to the question is therefore specific to each link. For each link we identify why gradient training leaves it closed, and we derive a temporal prior that opens it. The four priors are a retention gate that depends on the token, a bipolar graded release, a surrogate slope matched to the membrane noise, and a residual path for the values read from memory. With these priors, NeuronSpark trained from scratch reaches the level of mainstream architectures of the same size, the Transformer and the linear-time model Mamba-2, at four scales; in controlled comparisons, each prior opens the link it targets. We do not claim that spiking networks beat mainstream architectures. We explain why conversion succeeds without learning temporal dynamics and what direct training must add to reach the same accuracy.

Our contributions are as follows:

i. We derive the release identity of the bipolar soft-reset neuron and prove that the neuron is linear at the non-leaky endpoint and history dependent away from it.
ii. We decompose the dependence of a prediction on past tokens into a product of four links, write, release, readout and credit, and prove that no path from an earlier token bypasses them.
iii. We derive for each link why gradient training leaves it closed and which temporal prior opens it, and we build these priors into NeuronSpark, a spiking language model with bipolar PLIF neurons and synapse pools.
iv. We train NeuronSpark, Transformers and Mamba-2 from scratch at four scales with three seeds each, and test the predicted mechanism quantities, the parity with mainstream architectures and the energy per token.

2 Related Work

ANN-to-SNN conversion methods. Conversion replaces the activations of a trained ANN with integrate-and-fire neurons and rescales their thresholds so that firing rates approximate activations (Cao et al., 2015; Diehl et al., 2015; Rueckauer et al., 2017). Later methods shorten the required time window by soft reset and threshold balancing (Han et al., 2020; Sengupta et al., 2019), by calibrating the error layer by layer (Deng & Gu, 2021; Li et al., 2021), and by training the source network with clipped and quantized activations (Bu et al., 2022). Spiking counterparts of softmax, normalization and attention extend conversion to Transformers and to large language models (You et al., 2024; Hwang et al., 2024; Chen et al., 2026), and large spiking language models such as SpikingBrain start from a converted pretrained Transformer and continue its training rather than training from scratch (Pan et al., 2026). All of these methods run the neuron without leak and read its output as a rate over a window; our release identity shows why a network trained from scratch cannot rely on this endpoint.

Directly trained spiking models. Surrogate gradients replace the derivative of the threshold with a smooth pseudo-derivative and make backpropagation through time applicable to SNNs (Zenke & Ganguli, 2018; Wu et al., 2018; Shrestha & Orchard, 2018; Bellec et al., 2018; Neftci et al., 2019). Learnable membrane time constants improved directly trained SNNs (Fang et al., 2021), spike-driven self-attention brought direct training to vision Transformers (Zhou et al., 2023; Yao et al., 2023), and spiking language models followed with spiking recurrences (Zhu et al., 2024), spiking BERT variants (Lv et al., 2023; Bal & Sengupta, 2024) and elastic bidirectional spikes (Xing et al., 2024). These works report a gap to ANNs of the same size and attribute it to the information lost

by quantizing activations into spikes. We instead locate the gap in whether the temporal dynamics can be learned, and our bipolar release follows from its effect on the gradient rather than on the information a spike carries.

Linear-time sequence models. Linear attention, state-space models and gated recurrences replace the quadratic mixing of attention with a recurrent state (Katharopoulos et al., 2020; Gu et al., 2022; Gu & Dao, 2024; Dao & Gu, 2024; Peng et al., 2023; 2025; Yang et al., 2024; Beck et al., 2024; De et al., 2024). Their training depends on how the recurrence is initialized. Structured state spaces start from designed spectra (Gu et al., 2020; 2022), Mamba samples its step sizes over a range of time scales (Gu & Dao, 2024), and linear recurrent units keep eigenvalues close to the unit circle (Orvieto et al., 2023). A spiking layer is a recurrence of this family whose output passes a threshold and whose state is clamped, so its time constants face the same learnability question with an extra dead zone. Our time-constant spectrum is the PLIF counterpart of these initializations, and we compare against Mamba-2 as the linear-time model closest to NeuronSpark.

Trainability of recurrent and spiking networks. Gradients along long recurrences vanish or explode (Bengio et al., 1994; Hochreiter & Schmidhuber, 1997; Pascanu et al., 2013), and gate initializations such as chrono initialization set which time scales a recurrent network can learn (Tallec & Ollivier, 2018). Rectified units whose output stays at zero stop receiving gradient (Lu et al., 2020), and surrogate gradient learning is robust to the shape of the surrogate derivative but sensitive to its scale (Zenke & Vogels, 2021). Value residual learning passes the values of the first attention layer to deeper layers of a Transformer (Zhou et al., 2025). Each of these results concerns one mechanism in isolation; we show which of these mechanisms closes which link of a spiking network.

3 PRELIMINARIES

Leaky spiking neurons. A spiking channel is a parametric leaky integrate-and-fire (PLIF) neuron (Fang et al., 2021). At token t it receives a current I_t, keeps a fraction $a_t \in [0,1]$ of its carried state v_{t-1}, and releases part of its membrane potential u_t by a release rule R with threshold $\vartheta_t \geq 0$:

$$u_t = a_t\, v_{t-1} + I_t, \qquad S_t = R(u_t;\vartheta_t), \qquad v_t = u_t - S_t. \tag{1}$$

We call a_t the retention, S_t the release and v_t the soft reset; only S_t leaves the neuron. The one-sided graded release $S_t = (u_t - \vartheta_t)_+$ releases only above the threshold, and the bipolar graded release $S_t = (u_t - \vartheta_t)_+ - (-u_t - \vartheta_t)_+$ also releases below $-\vartheta_t$, with the reset $v_t = \mathrm{clamp}(u_t, -\vartheta_t, \vartheta_t)$. An input neuron of channel c sets its retention by a gate on its input x_t, $\eta_t = \sigma(w_\eta x_t + b_\eta)$, $a_t = 1 - \eta_t$ and $I_t = \eta_t x_t$, with per-channel parameters w_η and b_η; the gate is static if $w_\eta = 0$ and element-wise otherwise. Training replaces $\partial S_t / \partial u_t$ by a surrogate g_t in which each step becomes a logistic function of slope α (Neftci et al., 2019); the forward pass is exact.

Spiking layers and memory layers. The network stacks L layers on a residual stream $\boldsymbol{h}_t \in \mathbb{R}^D$. Three of every four layers are spiking layers, in which PLIF input neurons read $\mathrm{RMSNorm}(\boldsymbol{h}_t)$, and their release $\boldsymbol{s}_t$ drives a PLIF block of $D{\cdot}N$ spiking channels whose current, retention and threshold are projections of $\boldsymbol{s}_t$ ($\boldsymbol{W}_{\mathrm{in}} \to \boldsymbol{I}_t$, $\boldsymbol{W}_{\mathrm{decay}} \to \boldsymbol{a}_t$, $\boldsymbol{W}_{\mathrm{th}} \to \boldsymbol{\vartheta}_t$). Its gated output $\boldsymbol{y}_t = \sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t) \odot \boldsymbol{W}_{\mathrm{out}}\boldsymbol{S}_t + \boldsymbol{W}_{\mathrm{skip}}\boldsymbol{s}_t$ returns to the residual stream through an output projection. The fourth layer is a memory layer, in which synaptic projections form a pattern $\boldsymbol{p}_t$, a key $\boldsymbol{k}_j$ and a content $\boldsymbol{m}_j$ for every token; a synapse pool per head stores the contents under synaptic tagging τ_j, weights them by an eligibility trace $b_{t,j}$ with gain γ_h, and compares keys by dendritic pattern matching with gain β_h. Lateral inhibition normalizes the matches over the pool, and selective reactivation returns

$$\boldsymbol{r}_t = \sum_{j \leq t} w_{t,j}\, \tau_j\, \boldsymbol{m}_j, \qquad w_{t,j} = \frac{\exp(\beta_h\, \hat{\boldsymbol{p}}_t^{\top} \hat{\boldsymbol{k}}_j + b_{t,j})}{\sum_{i \leq t} \exp(\beta_h\, \hat{\boldsymbol{p}}_t^{\top} \hat{\boldsymbol{k}}_i + b_{t,i})}, \tag{2}$$

which a gated output projection adds to the residual stream. Every layer ends with a spiking gated FFN behind its own input neurons, whose gate and up neurons are PLIF neurons, and the LM head reads $\mathrm{RMSNorm}(\boldsymbol{h}_t)$. We call this network with the one-sided release, element-wise gates and the steep slope $\alpha = 96$ the base network.

Two routes and one gap. Conversion takes a trained ANN and replaces each activation by a neuron at the non-leaky endpoint $a_t \equiv 1$ that runs T steps per input, reading its output as the rate $\frac{1}{T}\sum_t S_t$ (Rueckauer et al., 2017). Training from scratch runs the network once per token with learned retentions $a_t < 1$ and optimizes all parameters by the surrogate gradient. We measure the gap between the final training cross-entropy of a spiking network and that of a Transformer of the same size trained on the same tokens. Spiking language models trained from scratch remain behind Transformers of

the same size, whereas conversion reaches the accuracy of its source; the problem is to explain this difference and to derive what the base network has to change for the gap to vanish.

4 WHY TRAINING FROM SCRATCH LAGS

A leaky neuron releases a function of when its inputs arrive and not only of how large they are (Section 4.1), so a network trained from scratch has to learn temporal parameters that a converted network never needs. A past token reaches the prediction, and its gradient the parameters, only through a product of four factors (Section 4.2), so the question becomes which factor gradient training leaves at zero. For each factor we derive why it stays at zero in the base network and what a component has to do to open it, and we design the component from this requirement (Section 4.3).

4.1 LEAK MAKES THE RELEASE DEPEND ON TIMING

The first step compares what a neuron releases over a window with what it receives, since the soft reset subtracts exactly what the neuron releases.

Proposition 4.1 (Release identity)**.** *Let a neuron follow equation 1 with any release rule. For every* $T \geq 1$*,*

$$\sum_{t=1}^{T} S_t = \sum_{t=1}^{T} I_t + a_1 v_0 - v_T - \sum_{t=1}^{T-1} (1 - a_{t+1})\, v_t. \tag{3}$$

At the non-leaky endpoint $a_t \equiv 1$ *with* $|v_0| \leq \vartheta_{\max}$*, the mean bipolar release therefore equals the mean current up to* $2\vartheta_{\max}/T$*, whereas for* $a_t < 1$ *the last sum depends on the order of the currents and not only on their sum.*

Writing the soft reset as $S_t = a_t v_{t-1} + I_t - v_t$ and summing over t, boundary terms telescope and the leak leaves the last sum; the bipolar reset bounds the boundary terms by the threshold (Appendix A.2). Proposition 4.1 separates the two routes. Conversion runs each neuron at $a \equiv 1$ for T steps, where its rate is a fixed function of the input rate, the identity for the bipolar and a rectifier for the one-sided release (Lemma A.1), so a converted network only has to match rates and its error falls as T grows (Rueckauer et al., 2017; Bu et al., 2022). A network trained from scratch updates each membrane once per token with $a < 1$, where the release depends on the history of the input and a dead zone opens with the leak, so the nonlinearity of the neuron exists because of its leak. No source network supplies this map, so the retention, the threshold and the reset have to be learned from data; the leaky network is not less expressive, but its function depends on temporal parameters that training has to find.

4.2 FOUR LINKS CARRY A PAST TOKEN

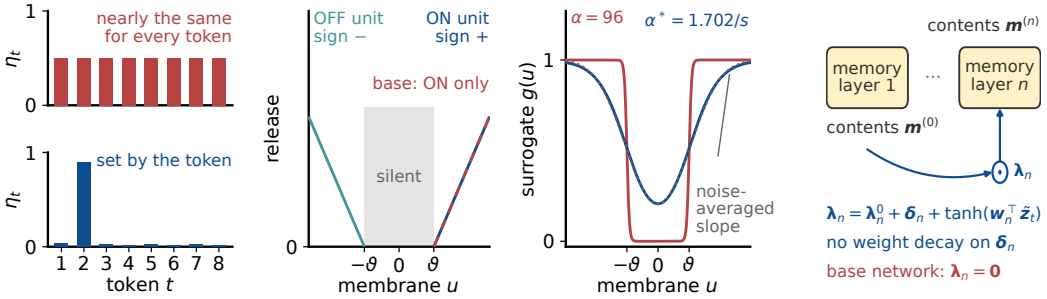
To ask what training has to find, we follow a past input through the block, writing $\boldsymbol{\Psi}'_t = \partial \boldsymbol{S}_t / \partial \boldsymbol{u}_t$ for the slope of the release.

Lemma 4.2 (Four links)**.** *Consider the membrane recurrence of a PLIF block with either release of Section 3. Between two resets, every path from its input* $\boldsymbol{s}_q$ *to its output* $\boldsymbol{y}_t$ *at lag* $t - q \geq 1$ *passes through* $\boldsymbol{\Psi}'_t$*, the retention product* $\prod_{r=q+1}^{t} \boldsymbol{a}_r$ *and* $\operatorname{diag}(\sigma(\boldsymbol{W}_{\text{gate}} \boldsymbol{s}_t))\boldsymbol{W}_{\text{out}}$*; the path through the current is*

$$\begin{aligned} \left.\frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{s}_q}\right|_{\boldsymbol{I}_q} &= \operatorname{diag}\big(\sigma(\boldsymbol{W}_{\text{gate}} \boldsymbol{s}_t)\big)\, \boldsymbol{W}_{\text{out}}\, \operatorname{diag}(\boldsymbol{\Psi}'_t)\, \operatorname{diag}(\boldsymbol{A}_{tq})\, \boldsymbol{W}_{\text{in}}, \\ \boldsymbol{A}_{tq} &= \prod_{r=q+1}^{t} \boldsymbol{a}_r \odot \prod_{r=q}^{t-1} (\mathbf{1} - \boldsymbol{\Psi}'_r), \end{aligned} \tag{4}$$

and the gradient that reaches the current runs back along the transposed product.

One step of the recurrence gives $\partial \boldsymbol{u}_{r+1} / \partial \boldsymbol{u}_r = \operatorname{diag}(\boldsymbol{a}_{r+1}) \operatorname{diag}(1 - \boldsymbol{\Psi}'_r)$, where $1 - \boldsymbol{\Psi}'_r$ marks the channels that release nothing at r, so the product of these steps is $\boldsymbol{A}_{tq}$, and the paths through $\boldsymbol{a}_q$ and $\boldsymbol{\vartheta}_q$ join the same chain (Appendix A.3). Lemma 4.2 names four links. A past token is *written* if $\boldsymbol{A}_{tq} \neq 0$, *released* if $\boldsymbol{\Psi}'_t \neq 0$, *read out* if the output projection and the later memory layers carry it to the prediction, and *credited* if the transposed product, in which training replaces each release slope by the surrogate, reaches the parameters; one closed link removes the token from the prediction and from the gradient at once. The write and release factors are functions of the retention, the threshold and the reset, the readout of a distant token passes through the eligibility trace of a memory layer, and the credit reaches the same temporal parameters; since these are the quantities



(a) write: the token sets η_t (b) release: ON and OFF units (c) credit: slope set by noise (d) readout: anchored path

Figure 2: **The four temporal priors against the base network.** Blue: with the prior; red: the base network.

that Proposition 4.1 identifies as temporal, a component that opens a link is a temporal prior, not extra width or depth.

4.3 Why each link stays closed, and the prior that opens it

Each proposition below states a property of the base network that keeps one link closed; negating it gives a requirement on any component that opens the link, and the design that follows is the simplest change of the base network that meets it (Figure 2), with sizes set by the protocol (Appendix A.1).

Write link. The input neurons of a spiking layer decide what enters the block, and in the base network their gate gives every token of a channel nearly the same retention; the question is whether such a retention can keep a token that the prediction needs later, and whether training can make it selective.

Proposition 4.3 (Write link)**.** *Between releases, a retention that is the same for every token gives the token n steps back at most a fraction $m_n = n^n/(n+1)^{n+1} \approx e^{-1}/(n+1)$ of the membrane variance of a channel with independent stationary inputs; and when some tokens need to be kept and others read in, the gradient of any gate parameter that enters every token alike, such as b_η, sums the two classes with opposite signs.*

Unrolling the recurrence at a constant retention gives the bound, and differentiating the gate gives the sums (Appendix A.4). A token needed 32 tokens later thus keeps at most 1% of the membrane whatever the retention, the gate bias stops where the two classes balance instead of learning which tokens to keep, and an element-wise gate, which reads only its own channel, fares no better. The retention therefore has to depend on the token through a learned read of the whole input, whose gradient adds the two classes.

Design 4.3 (Token projection gate)**.** The simplest change of the base gate that meets both requirements adds a linear read of the whole input to its logit, $\eta_t = \sigma(w_\eta x_t + b_\eta + (\boldsymbol{U}\boldsymbol{V}\tilde{\boldsymbol{x}}_t)_c)$, with $\boldsymbol{U}$ launched at zero and $\tilde{\boldsymbol{x}}_t$ the input centered on the running mean of its row, so that the row constant stays in the bias (Lemma A.5). The retention then differs between a token to keep and the tokens that follow it, so the bound m_n no longer applies, and the gradient of $\boldsymbol{U}$ weights each token by its own read $\boldsymbol{V}\tilde{\boldsymbol{x}}_t$, so the two classes add instead of cancelling (Appendix A.4). Since a constant retention serves one lag at a time, the FFN input neurons also launch from a spectrum of time constants, which weight decay is kept from contracting (Lemma A.4).

Release link. A unit that never crosses its threshold passes nothing downstream; the question is whether training can revive it, and which release rule leaves fewer such units.

Proposition 4.4 (Release link)**.** *A unit that releases nothing on a batch receives an exactly zero gradient on its readout column, for any surrogate and precision, and an input gradient proportional to that column, which AdamW only shrinks. Under the same drive, a unit silent under the bipolar release is also silent under the one-sided release, which for a symmetric Gaussian drive discards half of the release energy at every threshold.*

The readout gradient is a sum of terms proportional to the unit's release, and the second statement compares the conditions $u_t < \vartheta$ and $|u_t| < \vartheta$ and integrates the Gaussian tails (Appendix A.5). A one-sided unit whose drive is mostly negative therefore stays silent, and the optimizer can only push its threshold below zero, where the unit becomes affine and loses its state. The release therefore has to be silent only inside a band around zero.

Design 4.4 (Bipolar release)**.** The simplest release that is silent only inside a band around zero adds to the one-sided release its mirror image below $-\vartheta_t$, which is the bipolar release of Section 3, used in every neuron and block. By Proposition 4.4, under the same drive it leaves silent only units that the one-sided release also silences, a mostly negative drive now releases, and a symmetric drive keeps the half of the release energy that the one-sided release discards; its clamp reset keeps the parallel scan (Lemma A.6).

Credit link. The forward pass of a neuron is exact, so the surrogate alone decides how much gradient enters a membrane below threshold and how far it travels back. For a channel with the bipolar release, threshold ϑ and constant retention a, trained with the surrogate $g_t = \sigma(\alpha(u_t - \vartheta)) + \sigma(\alpha(-u_t - \vartheta))$, write $\rho_t = a(1 - g_t)$ for the backward retention; the entry $E_t = \sum_{k\geq 0} g_{t+k} \prod_{j<k} \rho_{t+j}$ is the gradient under a unit demand on every token, and the backward horizon $H_t = \sum_{k\geq 0} \prod_{j<k} \rho_{t+j}$ measures how far it travels.

Proposition 4.5 (Credit link)**.** *For every sequence of surrogates, $aE_t + (1-a)H_t = 1 - \prod_{j\geq t} \rho_j$, and the surrogate is smallest at $u_t = 0$, where $g_t = 2\sigma(-\alpha\vartheta)$.*

The deficit $1 - aE_t - (1-a)H_t$ is multiplied by ρ_t at every step back, and the derivative of the logistic function is even and falls with $|x|$ (Appendix A.6). A steep slope thus leaves a channel below threshold almost without entry, and widening it buys entry at the price of horizon, one for one, so the loss should first fall and then rise as the slope decreases. The slope therefore has to match the spread of the membrane.

Design 4.5 (Matched slope)**.** The slope is the only free parameter of the surrogate, and the simplest rule that meets the requirement matches the surrogate to the expected derivative of the release under Gaussian membrane noise of scale s, which holds within 0.0095 at $\alpha^* = 1.702/s$ (Lemma A.8). With s measured on the base network before any prior is added, this gives $\alpha^* \approx 8$ for its PLIF neurons, at which the floor $2\sigma(-\alpha^*\vartheta)$ exceeds 0.05 for every threshold below 0.46, against 0.04 at $\alpha = 96$.

Readout link. The memory layers carry distant tokens to the prediction, and their wide heads read nearly the whole pool; the question is how much of such a readout varies with the token.

Proposition 4.6 (Readout link)**.** *Let a head tag its events with a constant τ_j and read contents whose mean square splits into global, row and token shares $\pi_\mathrm{g} + \pi_\mathrm{r} + \pi_\mathrm{t} = 1$. Under the independence conditions of Appendix A.7, the share of its readout that is constant over rows and tokens is $c = \pi_\mathrm{g}/(\pi_\mathrm{g} + \pi_\mathrm{r} + \pi_\mathrm{t}\kappa)$ with $\kappa \geq 1/N_2$, where $N_2 = 1/\sum_j w_{t,j}^2$ counts the events the head reads, and contents passed through SiLU have $\pi_\mathrm{g} > 0$.*

The weights sum to one, so the readout splits into the same three components, with the token term reduced by κ (Appendix A.7). After normalization a constant share c scales the readout factor of Lemma 4.2 by $\sqrt{1-c}$, so a wide head of a deep memory layer, which spreads its weights over many events, passes little of what it reads. The deep heads therefore need a path to contents that still vary with the token, and the path has to be open from the start, since a path launched at zero receives no aligned gradient (Lemma A.9).

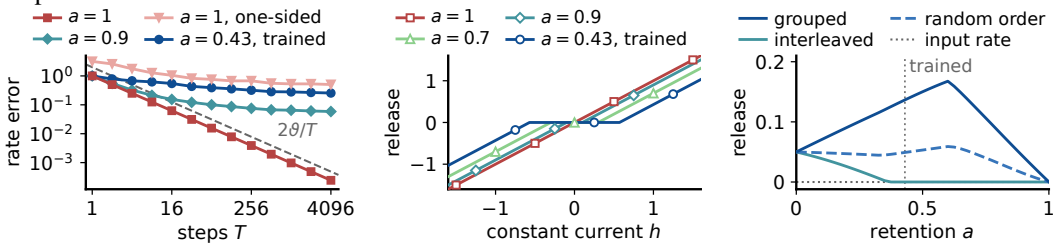
Design 4.6 (Anchored value residual)**.** The simplest path that meets both requirements adds the contents of the first memory layer, computed before any memory readout enters the residual stream, to those of each later memory layer, $\boldsymbol{m}_j^{(n)} \leftarrow \boldsymbol{m}_j^{(n)} + \boldsymbol{\lambda}_n \odot \boldsymbol{m}_j^{(0)}$, a value residual (Zhou et al., 2025) whose coefficient launches at an anchor $\boldsymbol{\lambda}_n^0 \neq \mathbf{0}$. By Lemma A.9 the parameters downstream then receive gradients through this path from the first update, and, as for the time constants of Design 4.3, weight decay is kept from the coefficient, so only a consistent gradient moves it off the anchor.

NeuronSpark. NeuronSpark is the base network with Designs 4.3 to 4.6 in its input neurons, its neurons and blocks, its PLIF neurons and its later memory layers (Figure 2), together with secondary components that act on the same factors (Appendix A.8); Appendix A.1 gives the whole network (Figure 11) with all equations and sizes. Section 5 tests the chain in the same order, first the identity and the four links, then where the difference to a Transformer forms, then each link with its prior, and finally the full model.

Relation to other sequence models. Read as an ANN layer, a bipolar PLIF block is a per-channel recurrence with an input-dependent decay, as in Mamba (Gu & Dao, 2024), whose state is clipped to $[-\vartheta_t, \vartheta_t]$ and which emits only the excess $S_t = u_t - v_t$ over this band, so a block whose channels stay inside the band emits nothing; a synapse pool is a forgetting attention (Lin et al., 2025) with $b_{t,j}$ as its summed log forget gates and each event weighted by its synaptic tag, and with $\gamma_h = 0$ and $\tau_j \equiv 1$ it reduces to softmax attention with temperature $1/\beta_h$.

5 EXPERIMENTS

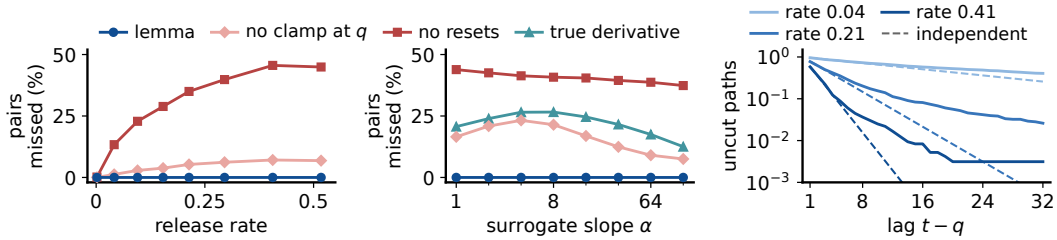
We check predictions P1–P7 of Appendix A.9.6, which follow the steps of Section 4, in simulation, on 200M models and synthetic streams, and on the full model against a Llama-style Transformer (Touvron et al., 2023a) and Mamba-2 (Dao & Gu, 2024) at four scales from 200M to 2.0B parameters. All models share the BabyLM 2026 Strict corpus (Choshen et al., 2026), tokenizer (Yang et al., 2025), token order, optimizer and seeds, and every difference is paired by seed. Appendix Table 2 gives the decisive test and result of each prediction, and Appendix A.9 maps each panel to its protocol.



(a) endpoint (b) dead zone (c) input order

Figure 3: **Release identity on a single neuron.**

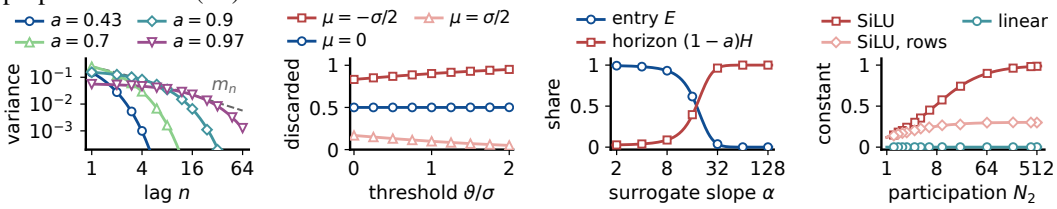
The release identity. A neuron releases its input rate only at the non-leaky endpoint, and with leak its release depends on the retention and on the order of its inputs (P1). At retention one the rate error of the bipolar neuron falls as $1/T$, against 0.50 for the one-sided neuron and 0.26 for a leaky neuron at the trained retention (Figure 3a), the release follows the dead zone of Lemma A.1 (b), and at the trained retention the same currents release 0.14 per step when grouped and nothing when interleaved (c). We next ask through which links a past input reaches the output (P1).



(a) exact release (b) surrogate gradient (c) reset survival

Figure 4: **Finite-difference and automatic-differentiation checks of the four-link product.**

The four links. A past input reaches the block output only through the product of Lemma 4.2, and a release cuts the path (P1). The product misses none of $21{,}656$ Jacobians against finite differences, whereas dropping the clamp factor at the input token misses up to 7% (Figure 4a), it matches automatic differentiation of the surrogate to 10^{-16} (b), and the resets cut all but 20% of the channel paths over 8 tokens at a release rate of 0.21 (c). We next ask whether the closed forms of the link propositions hold (P1).



(a) write (b) release (c) credit (d) readout

Figure 5: **Closed forms of the four link propositions against simulation.**

Closed forms of the links. The closed forms of the four link propositions match simulation (P1). Simulation stays within 0.007 of every closed form, with a static retention keeping at most 1.1% of a token 32 steps back (Figure 5a), the one-sided release discarding half of the energy of a symmetric drive (b), a wider surrogate trading backward horizon for entry one for one (c), and a wide head passing 0.99 of SiLU contents as a constant, against none of linear contents (d). We next ask where a network trained from scratch falls short of the Transformer (P2).

Where the difference forms. The difference to the Transformer forms over the last 2 to 32 tokens of context and not on the current token (P2). On new targets a network without the write, credit and readout priors reaches the loss of the Transformer from one token of context, and each doubling of the context adds 0.017 to 0.043 nats to its deficit, 0.166 nats with 32 tokens (Figure 6a). We next ask whether an input neuron can keep a token that long (P3).

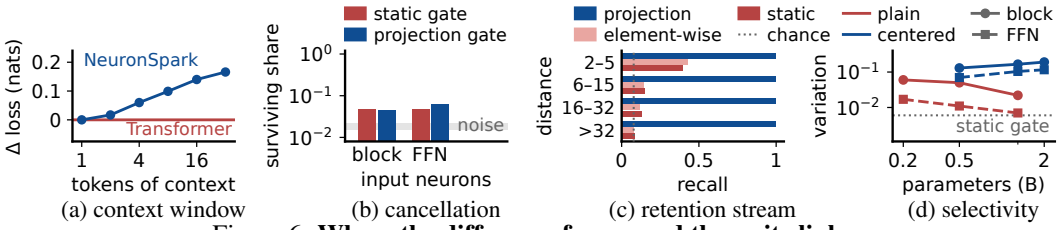


Figure 6: **Where the difference forms, and the write link.**

The write link. The gradient of the retention bias cancels across tokens, whereas the token projection gate keeps a token and, with a centered input, stays selective at scale (P3). Only 4 to 7% of the per-token gradient of the retention bias survives the sum over a batch, against about 2% for noise (Figure 6b); the static and element-wise gates recall a token past 32 distractors at chance and the projection gate at 0.996 (c); and at 1.3B the per-token variation of the block input gates is 0.022 without centering and 0.166 with it (d). We next ask whether a kept token is released (P4).

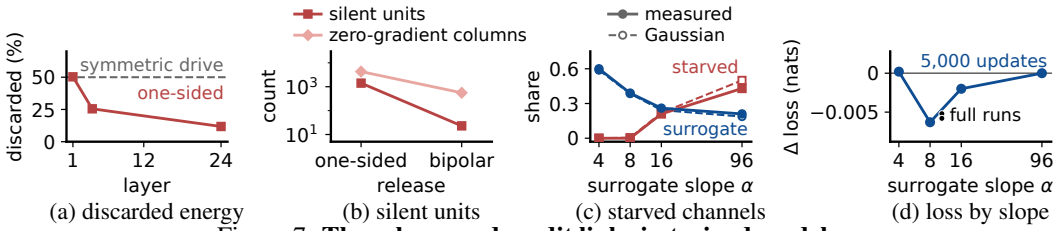


Figure 7: **The release and credit links in trained models.**

The release link. The one-sided release discards the negative side of the drive and leaves units that no gradient reaches, and the bipolar release removes them (P4). The negative side carries 50.3% of the release energy in the first layer, as Proposition 4.4 predicts for a symmetric drive (Figure 7a), and on paired checkpoints the bipolar release reduces the units that never release from 1,413 to 23 (b). We next ask whether a released token is credited back through the surrogate (P5).

The credit link. A steep surrogate starves the channels whose membrane stays below threshold, and a slope near the matched value removes them and lowers the loss (P5). At slopes 96, 16, 8 and 4 the share of starved PLIF channels is 0.43, 0.21, 0 and 0, close to a Gaussian membrane (Figure 7c), and the loss first falls and then rises as the slope decreases, while full runs at slope 8 lower the final loss by 0.0052 and 0.0058 on two seeds (d). We next ask whether the memory layers pass what they read to the prediction (P6).

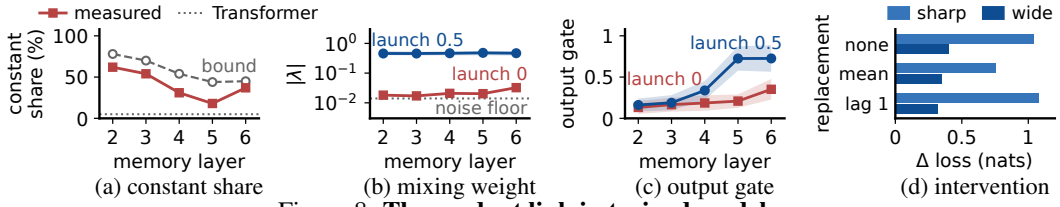
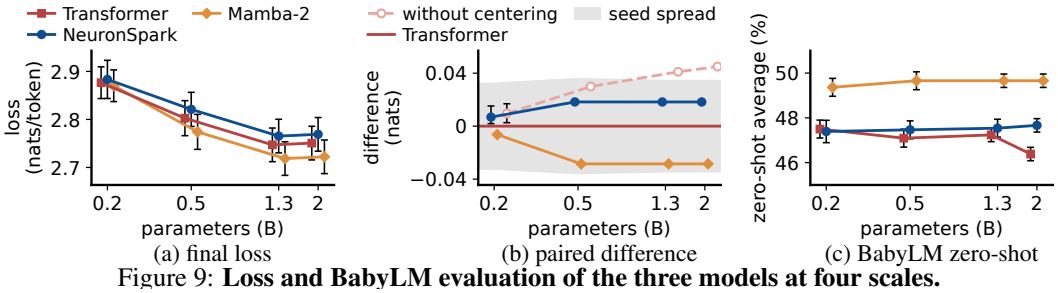


Figure 8: **The readout link in trained models.**

The readout link. Without a value residual the deep wide heads pass mostly a constant, and a residual launched away from zero lets them pass the contents of the first memory layer (P6). The constant takes 18 to 62% of the wide-head readout, below the bound of Proposition 4.6 and above the 5% of the Transformer's attention output (Figure 8a); launched at zero the mixing weight stays at its noise floor, as Lemma A.9 predicts, whereas launched at 0.5 it holds (b) and opens the output gates of the deepest wide heads (c), and replacing the carried contents by their mean costs the wide heads 0.35 of 0.40 nats (d). With the four links open, we next ask whether the full model reaches the level of mainstream architectures across scales (P7).

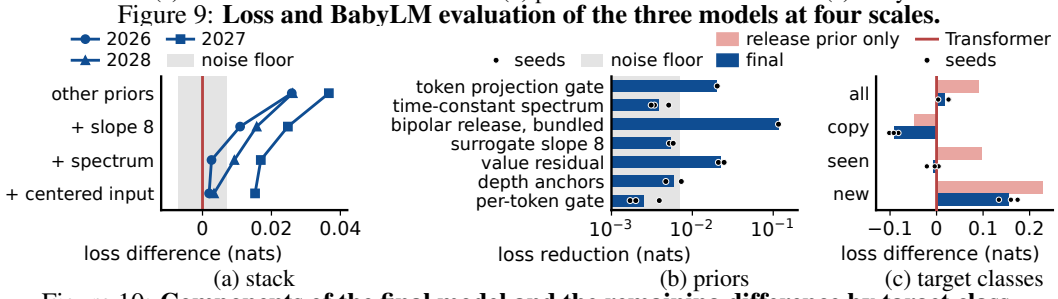
Parity across scales. NeuronSpark reaches the loss of mainstream architectures at every scale, and without the centered input its difference grows with the width (P7). The seed-paired difference of NeuronSpark to the Transformer is +0.007 nats at 200M and +0.018 at the larger scales, inside the Transformer's seed spread of 0.033 to 0.036 and of the size of the difference between the Transformer and Mamba-2, −0.006 at 200M and −0.029 at 500M, against +0.010 to +0.045 without the centered input (Figure 9a, b). We next ask how each prior contributes to the parity (P7).

Components and the remainder. Every prior lowers the loss, the priors together close the difference on every seed, and what remains lies on new targets (P7). At 200M, adding the matched slope, the time-constant spectrum and the centered input to a network with the other priors lowers the difference from +0.026–0.037 to +0.002–0.015 nats on the three seeds (Figure 10a), and each



(a) final loss (b) paired difference (c) BabyLM zero-shot

Figure 9: **Loss and BabyLM evaluation of the three models at four scales.**



(a) stack (b) priors (c) target classes

Figure 10: **Components of the final model and the remaining difference by target class.**

prior lowers the loss by 0.002 to 0.116 nats (b). The final model leads the Transformer by 0.092 nats on copy targets and ties on seen targets, while new targets remain 0.13 to 0.17 nats behind (c), a remainder that Appendix A.10 places in the silent spiking branch of the FFN. We next ask whether the parity in loss carries over to the evaluation tasks (P7).

Table 1: **BabyLM 2026 evaluation of the three models at four scales.**

	200M			500M			1.3B			2.0B		
Task	TF	NS	M2	TF	NS	M2	TF	NS	M2	TF	NS	M2
BLiMP	71.2	71.3	**74.6**	68.2	71.8	**76.0**	67.8	72.3	**76.0**	65.5	72.6	**76.0**
BLiMP supplement	60.9	60.4	**63.5**	60.1	60.1	**63.5**	61.8	58.6	**63.5**	59.9	58.4	**63.5**
Entity tracking	17.4	16.2	**19.9**	17.5	16.4	**19.9**	18.8	17.7	**19.9**	19.4	18.3	**19.9**
COMPS	53.5	54.0	**54.6**	53.7	**54.6**	**54.6**	53.3	54.0	**54.6**	53.3	54.0	**54.6**
GlobalPIQA	34.5	**35.0**	34.2	**36.0**	34.5	34.2	34.5	**35.2**	34.2	33.7	**35.0**	34.2
Average	47.5	47.4	**49.4**	47.1	47.5	**49.7**	47.2	47.5	**49.7**	46.4	47.7	**49.7**
Seed deviation	0.4	0.5	0.4	0.4	0.4	0.4	0.3	0.4	0.3	0.3	0.3	0.3

Zero-shot accuracy (%), mean of three seeds; TF: Transformer, NS: NeuronSpark, M2: Mamba-2; best of each scale in bold.

Evaluation and energy. On the BabyLM 2026 tasks NeuronSpark scores at the level of the Transformer (P7). Its zero-shot average stays within 0.4 points of the Transformer's at 200M and 500M, and its BLiMP is 71.3 against 71.2 at 200M and 71.8 against 68.2 at 500M, while Mamba-2 is about two points higher on average (Table 1 and Figure 9c). Event-driven projections give NeuronSpark the lowest energy per token of the three models at every scale, and the gap widens as the models grow (Appendix A.9.5). In these experiments, a spiking language model trained from scratch reaches the level of mainstream architectures in loss and in evaluation once temporal priors open the four links through which a past token reaches the prediction.

6 Conclusion

Conversion keeps a spiking neuron at a non-leaky endpoint where it integrates linearly, whereas training from scratch places it in a leaky regime where its release depends on the timing of past inputs. The chain rule of the network splits this learning problem into four links, write, release, readout and credit. In the base network they are closed by a static retention gate, a one-sided release, a surrogate too steep to pass gradient into slow membranes and a memory readout dominated by constants, none of which concerns what spikes can represent. A temporal prior derived for each link opens it. Numerically, the identity, the four-link product and the closed forms hold in simulation, and each prior opens its link on 200M models. Together the priors let a spiking language model trained from scratch reach the level of mainstream architectures of the same size, the Transformer and Mamba-2, and use less energy per token than either. Our experiments train on a single corpus for one epoch, while the four links hold for any spiking network trained with surrogate gradients. The remaining difference on never-seen targets points to the silent spiking branch of the FFN as the next link to open.

AI USE STATEMENT

In this work, we used generative AI tools to polish the text and to create the figures. Coding agents drew the figures as matplotlib scripts and editable vector diagrams. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y. Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations (ICLR)*, pp. 15664–15730, 2024.

Malyaban Bal and Abhronil Sengupta. SpikingBERT: Distilling BERT to train spiking language models using implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10998–11006, 2024. doi: 10.1609/aaai.v38i10.28975.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems*, volume 37, pp. 107547–107603, 2024.

Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Guy E. Blelloch. Prefix sums and their applications. Technical Report CMU-CS-90-190, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.

Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015. doi: 10.1007/s11263-014-0788-3.

Tyler A. Chang, Catherine Arnett, Abdelrahman Sadallah, Abdelrahman Eldesokey, Abeer Kashar, Abolade Daud, Abosede Grace Olanihun, Adamu Labaran Mohammed, Adeyemi Praise, Adhikarimayum Meerajita Sharma, et al. Global PIQA: Evaluating commonsense reasoning across 100+ languages and cultures, 2025. URL https://arxiv.org/abs/2510.24081.

Long Chen, Xiaotian Song, and Yanan Sun. LAS: Loss-less ANN-SNN conversion for fully spike-driven large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1730–1738, 2026. doi: 10.1609/aaai.v40i3.37151.

Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Jaap Jumelet, Tal Linzen, Aaron Mueller, Suchir Salhan, Raj Sanjay Shah, Alex Warstadt, and Ethan Gotlieb Wilcox. BabyLM turns 4 and goes multilingual: Call for papers for the 2026 BabyLM workshop, 2026. URL https://arxiv.org/abs/2602.20092.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. doi: 10.1109/MM.2018.112130359.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024. URL https://arxiv.org/abs/2402.19427.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2015. doi: 10.1109/IJCNN.2015.7280696.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2661–2671, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022.

Julia Gygax and Friedemann Zenke. Elucidating the theoretical underpinnings of surrogate gradient learning in spiking neural networks. *Neural Computation*, 37(5):886–925, 2025. doi: 10.1162/neco_a_01752.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13558–13567, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Sangwoo Hwang, Seunghyun Lee, Dahoon Park, Donghun Lee, and Jaeha Kung. SpikedAttention: Training-free and fully spike-driven transformer-to-SNN conversion with winner-oriented spike shift for softmax operation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 67422–67445, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165. PMLR, 2020.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.213.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6316–6325. PMLR, 2021.

Zhixuan Lin, Evgenii Nikishin, Xu He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. In *International Conference on Learning Representations (ICLR)*, pp. 69704–69738, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying ReLU and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28(5):1671–1706, 2020. doi: 10.4208/cicp.OA-2020-0165.

Changze Lv, Tianlong Li, Jianhan Xu, Chenxi Gu, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. SpikeBERT: A language spikformer learned from BERT with knowledge distillation, 2023. URL https://arxiv.org/abs/2308.15122.

Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345 (6197):668–673, 2014. doi: 10.1126/science.1254642.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2928–2949. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.eacl-main.213.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26670–26698. PMLR, 2023.

Yuqi Pan, Yupeng Feng, Jinghao Zhuang, Siyu Ding, Han Xu, Zehao Liu, Bohan Sun, Yuhong Chou, Xuerui Qiu, Anlin Deng, et al. SpikingBrain: Spiking brain-inspired large models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318. PMLR, 2013.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, et al. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.936.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, et al. RWKV-7 "Goose" with expressive dynamic state evolution. In *Conference on Language Modeling (COLM)*, 2025.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019. doi: 10.1038/s41586-019-1677-2.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017. doi: 10.3389/fnins.2017.00682.

Catherine D. Schuman, Shruti R. Kulkarni, Maryam Parsa, J. Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022. doi: 10.1038/s43588-021-00184-y.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019. doi: 10.3389/fnins.2019.00095.

Sumit B. Shrestha and Garrick Orchard. SLAYER: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Corentin Tallec and Yann Ollivier. Can recurrent neural networks warp time? In *International Conference on Learning Representations (ICLR)*, 2018.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pp. 1–34. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.conll-babylm.1.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018. doi: 10.3389/fnins.2018.00331.

Xingrun Xing, Zheng Zhang, Ziyi Ni, Shitao Xiao, Yiming Ju, Siqi Fan, Yequan Wang, Jiajun Zhang, and Guoqi Li. SpikeLM: Towards general spike-driven language modeling via elastic bi-spiking mechanisms. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54698–54714. PMLR, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024.

Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64043–64058, 2023.

Kang You, Zekai Xu, Chen Nie, Zhijie Deng, Qinghai Guo, Xiang Wang, and Zhezhi He. SpikeZIP-TF: Conversion is all you need for transformer-based SNN. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57367–57383. PMLR, 2024.

Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, 2018. doi: 10.1162/neco_a_01086.

Friedemann Zenke and Tim P. Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021. doi: 10.1162/neco_a_01367.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28341–28356. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1375.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *International Conference on Learning Representations (ICLR)*, 2023.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. SpikeGPT: Generative pre-trained language model with spiking neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

A PROOFS, PROTOCOL, AND DIAGNOSTICS

This appendix follows the argument of Section 4. Section A.1 gives every equation and size of NeuronSpark. Sections A.2 and A.3 prove the release identity and the four-link lemma, which together reduce the gap of a network trained from scratch to four factors of one product. Sections A.4 to A.7 then treat the links one at a time, each first showing why training leaves the factor at zero and then why the temporal prior makes it nonzero, and Section A.8 covers the secondary priors. Section A.9 tests the predictions in simulation, on 200M models and across scales, with the protocol, the baselines, the evaluation and the energy model, and Section A.10 locates the difference that remains. Table 2 maps each prediction to its decisive test and finding.

Table 2: **Summary of claims and tests.**

claim	test	finding
P1: identity	One neuron; three input orders	Bipolar error $< 2\vartheta/T$ at $a = 1$; 0.14 grouped, 0 interleaved at $a = 0.43$.
P1: four links	Finite and automatic differences	No Jacobian missed (3×10^{-10}); without the clamp at q, up to 7%.
P1: closed forms	Four closed forms in simulation	Within 0.007; $aE + (1-a)H = 1$ to 1.6×10^{-15}.
P2: context	Cold windows, new targets	0.000 with 1 token, +0.166 with 32.
P3: bound	Token past 32 distractors	Static 0.083, element-wise 0.079 (chance 0.078); projection 0.996.
P3: cancellation	Bias gradient over a batch	4–7% survives, against 2% for noise.
P3: weight decay	Recall time constants	$[4, 256] \to [2.5, 46]$ tokens; decay alone predicts $[2.7, 54.5]$.
P3: centering	Gates at 500M and 1.3B	Saturated $13.1 \to 0\%$ and $57.7 \to 0\%$; loss -0.0115 at 500M.
P4: silence	Frozen readout columns	1,233 columns at exactly 0, 89% from silent units.
P4: one-sided	Paired checkpoints	Silent units $1{,}413 \to 23$; zero-gradient columns $4{,}297 \to 557$.
P4: scan	Random clamp-affine triples	Closure 3×10^{-7} in FP32; associativity exact.
P5: slope	Slopes 96, 16, 8, 4; runs at 8	Starved 0.43, 0.21, 0, 0; full runs -0.0052, -0.0058.
P6: constant	Deep wide heads	0.18–0.62, below the bound.
P6: launch	Mixing weight, gates	Zero launch at the noise floor; 0.5 holds 0.46–0.48.
P6: carry	Carried contents replaced	Wide heads: mean +0.35 of +0.40 nats; sharp heads: shift +1.08.
P7: parity	Three models, four scales	NeuronSpark +0.007–+0.018 within 0.033–0.036; Mamba-2 -0.006 and -0.029.
P7: evaluation	BabyLM 2026 zero-shot tasks	Average within 0.4 of the Transformer at 200M and 500M; Mamba-2 about 2 higher.
P7: components	Runs with and without each prior	0.002–0.116 each; final +0.002 to +0.015.

A.1 MODEL DETAILS

Residual stream and layers. NeuronSpark embeds tokens with a table tied to the LM head and passes a residual stream $\boldsymbol{h}_t \in \mathbb{R}^D$ through L layers; every fourth layer, starting with the fourth, is a memory layer and the others are spiking layers, so $L/4$ memory layers alternate with groups of three spiking layers. At 200M, $D = 512$, $N = 4$ ($D \cdot N = 2{,}048$ spiking channels per block), $L = 24$ with six memory layers of two heads each, the FFN width is 256, and the vocabulary has 151,669 tokens; the model has 209.4M parameters. Table 5 lists the other scales. In Figure 11 the left column runs from the tokens to the LM head through L layers, a memory layer after every three spiking layers; the gray panel shows both layer types, and its insets expand the bipolar PLIF block and the synapse pool.

Input neurons. A spiking layer computes $\boldsymbol{z}_t = \mathrm{RMSNorm}(\boldsymbol{h}_t)$ and passes it through a group of bipolar PLIF input neurons, one per channel of $\boldsymbol{z}_t$, with the token projection gate of Section 4.3. Channel c follows $\eta_t = \sigma(w_\eta z_t + b_\eta + (\boldsymbol{U}\boldsymbol{V}\tilde{\boldsymbol{z}}_t)_c)$, $u_t = (1-\eta_t)v_{t-1} + \eta_t z_t$, $S_t = (u_t - \vartheta)_+ - (-u_t - \vartheta)_+$ and $v_t = u_t - S_t$, where the projection reads the centered input $\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t - \frac{1}{t-1}\sum_{s<t}\boldsymbol{z}_s$

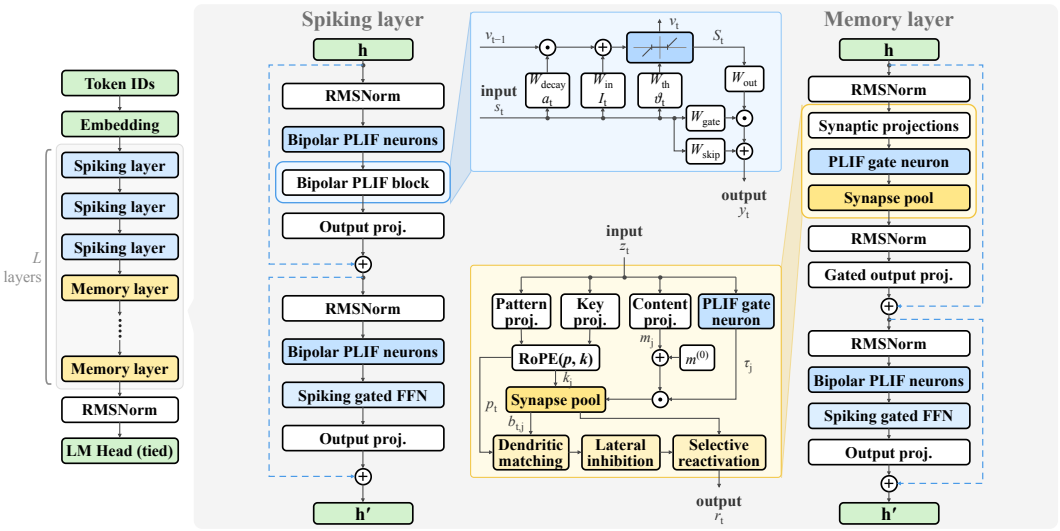


Figure 11: **NeuronSpark architecture.**

($\tilde{\boldsymbol{z}}_1 = \mathbf{0}$), with $\boldsymbol{U} \in \mathbb{R}^{D\times 16}$ initialized at zero, $\boldsymbol{V} \in \mathbb{R}^{16\times D}$ initialized uniformly, and a per-channel threshold ϑ. With a constant input the membrane settles at $u = z$, so η sets only the time constant $\tau = 1/\eta$. The release $\boldsymbol{s}_t$ of this group is the only signal that reaches the block. All gate weights w_η start at zero. The FFN input neurons launch from the time-constant spectrum, $\tau_c = \exp(\mathrm{linspace}(\log 2, \log 100, D))_c$, with $b_{\eta,c} = -\log(\tau_c - 1)$ plus Gaussian jitter of standard deviation 0.5, so that $\eta \approx 1/\tau_c$; the other input neurons launch at $\tau = 2$ with the same jitter. The thresholds of the FFN input neurons are matched on the launched model, where a bisection sets each ϑ_c so that the channel's release rate under the encoding recurrence equals the release rate of the fastest quartile of its group.

Bipolar PLIF block. The block has $D \cdot N$ channels with current $\boldsymbol{I}_t = \boldsymbol{W}_{\mathrm{in}}\boldsymbol{s}_t + \boldsymbol{j}_t$, retention $\boldsymbol{a}_t = \sigma(\boldsymbol{W}_{\mathrm{decay}}\boldsymbol{s}_t) \odot \sigma(\boldsymbol{b})$ and threshold $\boldsymbol{\vartheta}_t = \vartheta_{\min} + \mathrm{softplus}(\boldsymbol{W}_{\mathrm{th}}\boldsymbol{s}_t)$, where $\boldsymbol{b}$ holds a frozen spectrum $\sigma(b_c) = 2^{-1/\tau_c}$ with τ_c log-spaced over $[1, 64]$ plus a zero-initialized offset, $\boldsymbol{W}_{\mathrm{decay}}$ has no bias, $\vartheta_{\min} = 0.02$ and $\boldsymbol{W}_{\mathrm{th}}$ starts at zero. The membrane follows equation 1 with the bipolar release channel by channel, and the output $\boldsymbol{y}_t = \sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t) \odot \boldsymbol{W}_{\mathrm{out}}\boldsymbol{S}_t + \boldsymbol{W}_{\mathrm{skip}}\boldsymbol{s}_t$ enters the residual stream through an output projection, $\boldsymbol{h}_t \leftarrow \boldsymbol{h}_t + \boldsymbol{W}_{\mathrm{o}}\boldsymbol{y}_t$. The current $\boldsymbol{j}_t$ carries the Hebbian recall trace and the power-law history current. The recall trace keeps a matrix state $\boldsymbol{M}_t = \mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{M}_{t-1} + \hat{\boldsymbol{k}}_t\boldsymbol{c}_t^\top$ with unit keys $\hat{\boldsymbol{k}}_t \in \mathbb{R}^{24}$, contents $\boldsymbol{c}_t \in \mathbb{R}^{96}$ and per-slot decays $\lambda_n = 2^{-1/\tau_n}$, τ_n from 4 to 256 tokens, and reads $\boldsymbol{M}_t^\top\hat{\boldsymbol{k}}_t$; a per-token decay gate and a floor on the key norm keep the trace well conditioned, and the recall reaches the block output through a multiplicative readout gate. The power-law history current adds $\boldsymbol{Q}\, t^{-\alpha}\sum_{s<t}[(t-s)^\alpha - (t-s-1)^\alpha]\boldsymbol{P}\boldsymbol{s}_s$ with rank 32, four learnable exponents α launched at 0.5 and $\boldsymbol{Q}$ initialized at zero. The membrane starts every row from zero and carries its state across the 1,024 tokens of the row; the block is trained with a segmented scan that is bitwise equal to the sequential recurrence.

Memory layer. A memory layer computes $\boldsymbol{z}_t = \mathrm{RMSNorm}(\boldsymbol{h}_t)$ and synaptic projections $\boldsymbol{p}_t, \boldsymbol{k}_t, \boldsymbol{m}_t$, each followed by a causal depthwise convolution of width 4 and a SiLU (Ramachandran et al., 2017); patterns and keys carry rotary positions (Su et al., 2024) and are normalized to unit length. Each of the two heads keeps a synapse pool. Synaptic tagging stores $\boldsymbol{m}_j$ with weight $\tau_j = \sigma(g_j + b_h - 2)$, where g_j is the release of a bipolar PLIF gate neuron and b_h a per-head bias. The trace decay $\bar{a}_{s,c} = \exp(-e^{A_c}\,\mathrm{softplus}(\boldsymbol{W}_{\mathrm{decay}}\boldsymbol{z}_s)_c)$ uses time constants e^{-A_c} log-spaced over $[1, 512]$, and the eligibility trace is $b_{t,j} = \gamma_h \sum_{j<s\le t}\frac{1}{256}\sum_c \log \bar{a}_{s,c}$ with a per-head gain γ_h launched at 1 for the sharp head and 0.03 for the wide head. Dendritic pattern matching scales $\hat{\boldsymbol{p}}_t^\top\hat{\boldsymbol{k}}_j$ by $\sqrt{d_k}$ and a learnable per-head gain, lateral inhibition normalizes the matches as in equation 2, and the selective reactivations of the two heads are concatenated, normalized, gated by $\sigma(\boldsymbol{W}_g\boldsymbol{h}_t)$ and projected back to the residual stream. Four extra patterns of rank 64 read the wide head's pool and add their reactivations through a zero-initialized projection, and a content product term of rank 128 adds $\boldsymbol{U}_m(\mathrm{SiLU}(\boldsymbol{A}_m\boldsymbol{z}_t) \odot \boldsymbol{B}_m\boldsymbol{z}_t)$ to the contents of the wide head, with $\boldsymbol{U}_m$ initial-

ized at zero. The value residual adds $\boldsymbol{\lambda}_n \odot \boldsymbol{m}_j^{(0)}$ to the contents of the n-th later memory layer, with $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_n^0 + \boldsymbol{\delta}_n + \tanh(\boldsymbol{w}_n^\top \tilde{\boldsymbol{z}}_t)$, anchors $\boldsymbol{\lambda}_n^0 = (0.5, \lambda_n^{\text{wide}})$ with $\lambda_n^{\text{wide}} = 0.5, 0.5, 1, 2, 2$, $\boldsymbol{\delta}_n$ initialized at zero, and a per-token gate on the causally centered input $\tilde{\boldsymbol{z}}_t$.

Spiking gated FFN and training. Every layer ends with $\boldsymbol{h}_t \leftarrow \boldsymbol{h}_t + \boldsymbol{W}'_{\text{o}}\big(\boldsymbol{W}_{\text{down}}(\boldsymbol{S}_t^{\text{gate}} \odot \boldsymbol{S}_t^{\text{up}}) + \boldsymbol{W}'_{\text{skip}}\boldsymbol{s}'_t\big)$, where $\boldsymbol{s}'_t$ is the release of the FFN input neurons, which read $\mathrm{RMSNorm}(\boldsymbol{h}_t)$, and the gate and up neurons are bipolar PLIF neurons driven by projections of $\boldsymbol{s}'_t$. All neurons use the surrogate of Proposition 4.5 with slope 8 for PLIF neurons and 96 for the block. Parameters and gradients are held in FP32 and the forward and backward passes run in bfloat16; one-dimensional parameters are exempt from weight decay. Two implementation details are not part of the theory. The output normalization of each memory layer launches at unit gain with its decay spectrum in FP32 and every mechanism draws its random launch independently, and the recall keys are normalized by $\max(\|\boldsymbol{k}\|, 10^{-2})$ instead of $\|\boldsymbol{k}\|$, which bounds the backward gain at a token whose block input is exactly silent.

A.2 Proof of the Release Identity

We prove Proposition 4.1 and Lemma A.1. No property of the release rule is used except that the reset subtracts the release.

A.2.1 The identity

Proof of Proposition 4.1. From $u_t = a_t v_{t-1} + I_t$ and $v_t = u_t - S_t$ we have, for every t,

$$S_t = a_t v_{t-1} + I_t - v_t.$$

Summing over $t = 1, \dots, T$ and separating the first and last terms gives

$$\begin{aligned}\sum_{t=1}^{T} S_t &= \sum_{t=1}^{T} I_t + a_1 v_0 + \sum_{t=1}^{T-1} a_{t+1} v_t - \sum_{t=1}^{T-1} v_t - v_T \\ &= \sum_{t=1}^{T} I_t + a_1 v_0 - v_T - \sum_{t=1}^{T-1} (1 - a_{t+1})\, v_t,\end{aligned}$$

which is equation 3. The rule R enters only through v_t, so the identity holds for graded, binary, one-sided and bipolar releases alike.

With $a_t \equiv 1$ the leak sum vanishes and

$$\sum_t S_t - \sum_t I_t = v_0 - v_T.$$

For the bipolar release $v_T = \mathrm{clamp}(u_T, -\vartheta_T, \vartheta_T)$, so $|v_T| \le \vartheta_{\max}$. With $|v_0| \le \vartheta_{\max}$ the difference is at most $2\vartheta_{\max}$, and dividing by T gives the bound on the mean release.

For time-varying currents and $a < 1$ the leak sum depends on the whole path $v_1, \dots, v_{T-1}$. Take $a = \frac{1}{2}$, $\vartheta = 1$, $v_0 = 0$ and the two currents $(\frac{3}{2}, \frac{1}{2})$ and $(\frac{1}{2}, \frac{3}{2})$, whose sums are equal. The first releases $\frac{1}{2}$ at $t = 1$ and nothing at $t = 2$, and the second releases nothing at $t = 1$ and $\frac{3}{4}$ at $t = 2$. Their total releases $\frac{1}{2}$ and $\frac{3}{4}$ differ, whereas at $a = 1$ both equal 1, as equation 3 requires. □

A.2.2 The endpoint rate and the dead zone

The next lemma gives the map from a constant input to the release per token at the endpoint and away from it.

Lemma A.1 (Endpoint rate and dead zone)**.** *At $a_t \equiv 1$ the one-sided release under a constant current $\bar{I}$ satisfies*

$$\frac{1}{T}\sum_{t=1}^{T} S_t \;\to\; (\bar{I})_+. \tag{5}$$

For $a_t \equiv a < 1$ and a constant current h, the bipolar release converges to

$$S^* = \operatorname{sign}(h)\,(|h| - (1-a)\vartheta)_+ \tag{6}$$

per token.

Proof. For the one-sided release $S_t = (u_t - \vartheta)_+$ the reset gives $v_t = \min(u_t, \vartheta)$, so $v_t \le \vartheta$ for all t, and we assume $v_0 \le \vartheta$. If $\bar{I} > 0$, then v_t grows by $\bar{I}$ per step until $v_{t-1} + \bar{I} \ge \vartheta$, after which $v_t = \vartheta$ and $S_t = \bar{I}$ for every later t, so the mean release tends to $\bar{I}$. If $\bar{I} \le 0$, then $u_t = v_{t-1} + \bar{I} \le \vartheta$ for every t and $S_t = 0$. This proves equation 5.

For a constant current h and $a < 1$ the bipolar state follows

$$v_t = F(v_{t-1}), \qquad F(v) = \operatorname{clamp}(av + h, -\vartheta, \vartheta).$$

Since the clamp is 1-Lipschitz, $|F(v) - F(v')| \le a|v - v'|$, so F is a contraction with a unique fixed point v^*, and $v_t \to v^*$ geometrically. If $|h| \le (1-a)\vartheta$, then $v^* = h/(1-a)$ lies in $[-\vartheta, \vartheta]$ and satisfies $av^* + h = v^*$, so $u^* = v^*$ and the release is zero. If $h > (1-a)\vartheta$, then $v^* = \vartheta$ because $a\vartheta + h > \vartheta$, and the release is

$$u^* - v^* = a\vartheta + h - \vartheta = h - (1-a)\vartheta;$$

the case $h < -(1-a)\vartheta$ is symmetric. Together these cases give equation 6. □

The first statement is the rate relation of integrate-and-fire neurons with reset by subtraction that conversion relies on (Rueckauer et al., 2017), written for graded releases. At the endpoint the neuron is therefore a fixed function of its input rate, the identity for the bipolar release and a rectifier for the one-sided release, whereas for $a < 1$ a dead zone of half-width $(1-a)\vartheta$ opens around zero. Figure 3 checks the three statements numerically (Appendix A.9.1).

A.2.3 Scope

The proposition concerns one neuron. It shows that the nonlinearity of the bipolar neuron at a fixed input rate appears only for $a < 1$, where it is a dead zone of half-width $(1-a)\vartheta$ followed by a clamp. It does not compare the function classes of a leaky network and a Transformer, and it does not bound how many steps conversion needs, which depends on the source network (Bu et al., 2022).

A.3 Proof of the Four-Link Lemma

We prove Lemma 4.2, together with its surrogate version and the adjoint recursion used in Sections A.4 to A.7. All operations below act channel by channel on the $D \cdot N$ channels of the block.

A.3.1 The forward product

Proof of Lemma 4.2. From $\boldsymbol{u}_{r+1} = \boldsymbol{a}_{r+1} \odot \boldsymbol{v}_r + \boldsymbol{I}_{r+1}$ and $\boldsymbol{v}_r = \boldsymbol{u}_r - \boldsymbol{S}_r$, the Jacobian of one step with respect to the membrane is

$$\frac{\partial \boldsymbol{u}_{r+1}}{\partial \boldsymbol{u}_r} = \operatorname{diag}(\boldsymbol{a}_{r+1})\,\operatorname{diag}(\mathbf{1} - \boldsymbol{\Psi}'_r), \tag{7}$$

where $\boldsymbol{a}_{r+1}$ and $\boldsymbol{I}_{r+1}$ depend on $\boldsymbol{s}_{r+1}$ only. For the exact bipolar release, $\boldsymbol{\Psi}'_r = \mathbf{1}\{|\boldsymbol{u}_r| > \boldsymbol{\vartheta}_r\}$ outside the null set $|\boldsymbol{u}_r| = \boldsymbol{\vartheta}_r$, so $\mathbf{1} - \boldsymbol{\Psi}'_r = \mathbf{1}\{|\boldsymbol{u}_r| < \boldsymbol{\vartheta}_r\}$ almost everywhere; for the one-sided release, $\mathbf{1} - \boldsymbol{\Psi}'_r = \mathbf{1}\{\boldsymbol{u}_r < \boldsymbol{\vartheta}_r\}$.

For $q < t$ the input $\boldsymbol{s}_q$ enters the block at token q through $\boldsymbol{I}_q$, $\boldsymbol{a}_q$ and $\boldsymbol{\vartheta}_q$; the gate $\sigma(\boldsymbol{W}_{\text{gate}}\boldsymbol{s}_q)$ and $\boldsymbol{W}_{\text{skip}}\boldsymbol{s}_q$ act only on $\boldsymbol{y}_q$. The three entries have the Jacobians

$$\left.\frac{\partial \boldsymbol{u}_q}{\partial \boldsymbol{s}_q}\right|_{\boldsymbol{I}_q} = \boldsymbol{W}_{\text{in}}, \qquad \frac{\partial \boldsymbol{u}_q}{\partial \boldsymbol{a}_q} = \operatorname{diag}(\boldsymbol{v}_{q-1}), \qquad \frac{\partial \boldsymbol{v}_q}{\partial \boldsymbol{\vartheta}_q} = \operatorname{diag}\big(\operatorname{sign}(\boldsymbol{u}_q) \odot \mathbf{1}\{|\boldsymbol{u}_q| \ge \boldsymbol{\vartheta}_q\}\big),$$

with $\mathrm{diag}(\mathbf{1}\{\boldsymbol{u}_q \geq \boldsymbol{\vartheta}_q\})$ in the last one for the one-sided release. From $\boldsymbol{u}_q$ the chain of one-step Jacobians equation 7 reaches $\boldsymbol{u}_t$ with the product

$$\prod_{r=q}^{t-1} \mathrm{diag}(\boldsymbol{a}_{r+1})\ \mathrm{diag}(\mathbf{1} - \boldsymbol{\Psi}'_r) = \mathrm{diag}(\boldsymbol{A}_{tq}),$$

and from $\boldsymbol{v}_q$ with the same product without the factor at $r = q$. Finally $\boldsymbol{y}_t = \sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t) \odot \boldsymbol{W}_{\mathrm{out}}\boldsymbol{S}_t + \boldsymbol{W}_{\mathrm{skip}}\boldsymbol{s}_t$ depends on $\boldsymbol{u}_t$ only through $\boldsymbol{S}_t$, with

$$\frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{u}_t} = \mathrm{diag}\left(\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t)\right) \boldsymbol{W}_{\mathrm{out}}\ \mathrm{diag}(\boldsymbol{\Psi}'_t).$$

Every path therefore carries $\boldsymbol{\Psi}'_t$, the product $\prod_{r=q+1}^{t} \boldsymbol{a}_r$ and $\mathrm{diag}(\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t))\boldsymbol{W}_{\mathrm{out}}$, and the path through the current is equation 4. The skip term contains $\boldsymbol{s}_t$ only. □

A.3.2 The adjoint

The gradient runs back along the transposed product. Let $\boldsymbol{e}_t = \boldsymbol{W}_{\mathrm{out}}^{\top} \mathrm{diag}(\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t))\boldsymbol{\delta}_t$ be the gradient that reaches $\boldsymbol{S}_t$ from the output at t. Since $\boldsymbol{u}_t$ affects the loss through $\boldsymbol{S}_t$ and through $\boldsymbol{v}_t$, the gradient $\boldsymbol{\lambda}_t = \partial\mathcal{L}/\partial\boldsymbol{u}_t$ satisfies

$$\boldsymbol{\lambda}_t = \boldsymbol{\Psi}'_t \odot \boldsymbol{e}_t + (\mathbf{1} - \boldsymbol{\Psi}'_t) \odot \boldsymbol{a}_{t+1} \odot \boldsymbol{\lambda}_{t+1}, \qquad \boldsymbol{\lambda}_q = \sum_{t \geq q} \boldsymbol{A}_{tq} \odot \boldsymbol{\Psi}'_t \odot \boldsymbol{e}_t, \tag{8}$$

which is the transposed product of the lemma. The parameter gradients are then

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\mathrm{in}}} = \sum_q \boldsymbol{\lambda}_q \boldsymbol{s}_q^{\top}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_r} = \boldsymbol{\lambda}_r \odot \boldsymbol{v}_{r-1},$$

so learning a retention for lag k needs both the backward product and the forward content $\boldsymbol{v}_{r-1}$ to be nonzero. The j-th column of the readout gradient, $\sum_t (\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t) \odot \boldsymbol{\delta}_t) S_{t,j}$, depends only on the forward release, which Proposition 4.4 uses.

A.3.3 The surrogate version

Training replaces $\boldsymbol{\Psi}'_r$ by $\boldsymbol{g}_r = \sigma(\alpha(\boldsymbol{u}_r - \boldsymbol{\vartheta}_r)) + \sigma(\alpha(-\boldsymbol{u}_r - \boldsymbol{\vartheta}_r))$, or by its first term for the one-sided release, and the same bookkeeping gives

$$\boldsymbol{A}^{\mathrm{sur}}_{tq} = \prod_{r=q+1}^{t} \boldsymbol{a}_r \odot \prod_{r=q}^{t-1} (\mathbf{1} - \boldsymbol{g}_r). \tag{9}$$

Dropping the clamp factor at $r = q$ would be wrong, since the soft reset at the input token already cuts the path. Appendix A.9.1 checks equation 4 against finite differences and equation 9 against automatic differentiation.

A.3.4 Additional write paths

The recall trace and the power-law history current add $\boldsymbol{j}_t$ to the current at t, with the kernels

$$\hat{\boldsymbol{k}}_t^{\top}\ \mathrm{diag}(\boldsymbol{\lambda})^{t-s} \hat{\boldsymbol{k}}_s \qquad \text{and} \qquad \frac{(t-s)^{\alpha} - (t-s-1)^{\alpha}}{t^{\alpha}}$$

in place of the retention product. Both enter $\boldsymbol{u}_t$ before the release, so the factors $\boldsymbol{\Psi}'_t$ and $\mathrm{diag}(\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t))\boldsymbol{W}_{\mathrm{out}}$ remain on these paths, and the multiplicative recall readout gate scales $\boldsymbol{W}_{\mathrm{out}}\boldsymbol{S}_t$ and contributes nothing when $\boldsymbol{S}_t = 0$.

A.3.5 Scope

The lemma is an exact statement for each sample and for first-order effects. It says that a zero factor removes a dependency and does not quantify how small a nonzero factor may be before it stops mattering, which the link propositions address. The memory layers add the readout factor $w_{t,j}\tau_j$ of equation 2 downstream of the block; only its product structure is used here.

A.4 WRITE LINK: PROOFS OF PROPOSITION 4.3

The input neurons of a spiking layer decide what enters the block. Their release $\boldsymbol{s}_q$ is where every path of Lemma 4.2 starts, and their own membrane recurrence is a retention product of the form equation 4 placed before the block. We prove the two statements of Proposition 4.3 in turn, the bound for a retention that is the same for every token and the cancellation of the gradient of any gate parameter that enters every token alike, and then derive the rest of the prior from them.

A.4.1 SETTING

We follow one channel c of a group of bipolar PLIF input neurons (Fang et al., 2021) and drop the index c where it is not needed. With input x_t, the gate logit and the gate are

$$z_t = w_\eta x_t + b_\eta + (\boldsymbol{U}\boldsymbol{V}\boldsymbol{x}_t)_c, \qquad \eta_t = \sigma(z_t), \tag{10}$$

and the membrane of equation 1, with $a_t = 1 - \eta_t$ and $I_t = \eta_t x_t$, is

$$u_t = v_{t-1} + \eta_t(x_t - v_{t-1}), \qquad S_t = (u_t - \vartheta)_+ - (-u_t - \vartheta)_+, \qquad v_t = u_t - S_t, \tag{11}$$

with a per-channel threshold $\vartheta \geq 0$. The gate is static if $w_\eta = 0$ and $\boldsymbol{U} = \boldsymbol{0}$, and element-wise if only $\boldsymbol{U} = \boldsymbol{0}$. Training uses AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$ and weight decay $\lambda_{\mathrm{wd}} = 0.1$. The learning rate rises linearly to 3×10^{-4} over 1,000 updates and follows a cosine to 3×10^{-5} at update 19,904, so that $\sum_k \mathrm{lr}_k = 3.2694$.

A.4.2 THE BOUND

The first statement concerns a gate that gives every token the same retention.

Lemma A.2 (Lag share under a constant gate)**.** *Let $\eta_t \equiv \eta \in (0, 1]$, let the inputs be independent with variance σ^2, and consider a window without release. In the stationary state the input n steps back contributes at most the share*

$$m_n = \frac{n^n}{(n+1)^{n+1}} \tag{12}$$

of the membrane variance, and

$$\frac{1}{e\,(n+1)} \leq m_n \leq \frac{e^{1/(n+1)}}{e\,(n+1)}. \tag{13}$$

Proof. Without release $v_t = u_t$, and unrolling $u_t = (1-\eta)u_{t-1} + \eta x_t$ gives

$$u_t = \sum_{n \geq 0} \eta(1-\eta)^n \, x_{t-n}.$$

The input at lag n therefore enters with the weight $\eta(1-\eta)^n$ whatever the tokens contain, which is the drive η times the retention product of equation 4 with $a_r \equiv 1 - \eta$ and no clamp factor. In the stationary state

$$\mathrm{Var}[u] = \eta^2\sigma^2 \sum_{k \geq 0} (1-\eta)^{2k} = \frac{\eta\,\sigma^2}{2-\eta},$$

and the lag-n term contributes $\eta^2(1-\eta)^{2n}\sigma^2$. With $q = (1-\eta)^2$ its share and the derivative of the share are

$$\frac{\eta^2(1-\eta)^{2n}\sigma^2}{\mathrm{Var}[u]} = q^n(1-q), \qquad \frac{d}{dq}\big[q^n(1-q)\big] = q^{n-1}\big(n(1-q) - q\big).$$

The derivative vanishes only at $q^* = n/(n+1)$, where the share equals $(n/(n+1))^n/(n+1) = m_n$. As η runs over $(0, 1]$, q runs over $[0, 1)$, so no η gives lag n a larger share. For equation 13, $(1 + \frac{1}{n})^n \leq e$ gives $(\frac{n}{n+1})^n \geq e^{-1}$, and $\ln(1-x) \leq -x$ gives

$$\Big(\frac{n}{n+1}\Big)^n = \exp\Big(n \ln\big(1 - \tfrac{1}{n+1}\big)\Big) \leq e^{-1}e^{1/(n+1)}.$$

□

The two sides of equation 13 differ by the factor $e^{1/(n+1)} \to 1$, so $m_n \approx e^{-1}/(n+1)$; for instance $m_6 = 0.057$, $m_{16} = 0.022$ and $m_{32} = 0.011$. The weight obeys the same bound, since $\eta(1-\eta)^n$ is largest at $\eta = 1/(n+1)$, where it also equals m_n, and at the launch value $\eta = \frac{1}{2}$ the share is $\frac{3}{4} \cdot 4^{-n}$. An element-wise gate depends only on the channel's own coordinate, with $|\eta_t - \sigma(b_\eta)| \le \frac{1}{4}|w_\eta x_t|$, so up to this modulation its kernel is $\eta(1-\eta)^n$ as well, and Lemma A.2 applies to it. This proves the first statement of Proposition 4.3.

A.4.3 Why a token projection escapes the bound

Keeping a token selectively requires a large η on the token to be written and a small η on the distractors that follow. A static gate has the same η on both, so every distractor overwrites with weight η and Lemma A.2 applies. An element-wise gate is a monotone function of one coordinate. When written tokens and distractors have the same distribution in every channel, half of the distractors are written at least as strongly as the token, and the gate does no better than a static one. A projection of rank one,

$$(\boldsymbol{U}\boldsymbol{V}\boldsymbol{x}_t)_c = \nu_c \, \boldsymbol{v}^\top \boldsymbol{x}_t,$$

instead reads a direction $\boldsymbol{v}$ that separates the two classes. With a hard gate at the midpoint of a separation d', misses and false writes both occur with probability $f = \Phi(-d'/2)$, and a written token survives k distractors with probability $(1-f)^k$. With soft gates of 0.9 on the written token and 0.01 on the distractors, the weight at lag 32 is $0.9 \cdot 0.99^{32} = 0.65$, against $m_{32} = 0.011$ for every static gate.

A.4.4 The cancellation

The second statement concerns the gradients of the gate parameters.

Lemma A.3 (Gradients of the gate)**.** *Let $\lambda_t = \partial \mathcal{L}/\partial u_t$ be the total derivative, which collects all later tokens through the adjoint recursion of Lemma A.7, and let*

$$\gamma_t = \lambda_t \, \eta_t(1-\eta_t) \, (x_t - v_{t-1}). \tag{14}$$

Summed over the batch,

$$\frac{\partial \mathcal{L}}{\partial b_\eta} = \sum_t \gamma_t, \qquad \frac{\partial \mathcal{L}}{\partial w_\eta} = \sum_t \gamma_t \, x_t, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{U}} = \sum_t \boldsymbol{\gamma}_t (\boldsymbol{V}\boldsymbol{x}_t)^\top. \tag{15}$$

Proof. The gate parameters enter every token only through z_t, with $\partial \eta_t/\partial z_t = \eta_t(1-\eta_t)$, and equation 11 gives $\partial u_t/\partial \eta_t = x_t - v_{t-1}$, so $\partial \mathcal{L}/\partial z_t = \gamma_t$. Since $\partial z_t/\partial b_\eta = 1$, $\partial z_t/\partial w_\eta = x_t$ and $\partial z_t/\partial \boldsymbol{U}_{c,r} = (\boldsymbol{V}\boldsymbol{x}_t)_r$, the chain rule gives the three sums, and stacking the channels gives the last one. □

The innovation $x_t - v_{t-1}$ has no systematic sign. In the stationary state $\mathbb{E}[u] = (1-\eta)\mathbb{E}[v] + \eta\mu$ and $\mathbb{E}[v] = \mathbb{E}[u] - \mathbb{E}[S]$, so

$$\mathbb{E}[x_t - v_{t-1}] = \frac{\mathbb{E}[S]}{\eta},$$

which vanishes for the bipolar release of a symmetric input. Now let the consumers downstream need some tokens read in, a class R with $\gamma_t < 0$ so that a descent step raises η_t, and others kept, a class H with $\gamma_t > 0$. By equation 15,

$$\frac{\partial \mathcal{L}}{\partial b_\eta} = \sum_{t \in H} |\gamma_t| - \sum_{t \in R} |\gamma_t| \tag{16}$$

is a difference of two sums, and the bias stops where they balance while $\sum_t |\gamma_t|$ stays large. Any gate parameter whose derivative $\partial z_t/\partial \theta$ takes one value on both classes gives that value times the

same difference. If instead a row of $\boldsymbol{V}$ gives $(\boldsymbol{V}\boldsymbol{x}_t)_r = \zeta_t > 0$ on R and $(\boldsymbol{V}\boldsymbol{x}_t)_r = -\zeta_t < 0$ on H, then

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{U}_{c,r}} = -\sum_{t \in R} |\gamma_t|\, \zeta_t - \sum_{t \in H} |\gamma_t|\, \zeta_t$$

adds both classes with one sign, and a descent step raises the logit on R and lowers it on H. Since $\boldsymbol{U}$ starts at zero but $\boldsymbol{V}\boldsymbol{x}_t \neq \boldsymbol{0}$, this gradient is present from the first update. This proves the second statement of Proposition 4.3.

A.4.5 Weight decay and the row constant

Two further lemmas complete the prior. Weight decay pulls the temporal parameters toward degenerate values, and a linear read of the input carries the row constant unless the input is centered.

Lemma A.4 (Weight decay on raw time constants)**.** *Decoupled weight decay pulls every time constant stored as a raw value toward its zero, which is a two-token time constant for b_η, one token for the logarithmic time constants of the recall trace, and a neuron that keeps no state for ϑ.*

Proof. Without the gradient term, each update multiplies a parameter by $1 - \mathrm{lr}_k \lambda_{\mathrm{wd}}$, so

$$p_T = p_0 \prod_{k<T} (1 - \mathrm{lr}_k \lambda_{\mathrm{wd}}) = p_0\, e^{-\lambda_{\mathrm{wd}} \sum_k \mathrm{lr}_k} \big(1 + O(\lambda_{\mathrm{wd}}^2 \textstyle\sum_k \mathrm{lr}_k^2)\big),$$

and over the schedule the factor is $e^{-0.1 \cdot 3.2694} = 0.7211$. The zeros are degenerate temporal settings. The value $b_\eta = 0$ gives $\eta = \frac{1}{2}$, a time constant of two tokens; $\log \tau = 0$ gives $\tau = 1$ token for the recall trace, whose time constants are stored as $\log \tau$; and $\vartheta = 0$ gives $S_t = u_t$ and $v_t = 0$, a neuron that releases its whole membrane and keeps no state. A spectrum written into the bias as $b_{\eta,c} = -\log(\tau_c - 1)$ therefore contracts as

$$\tau_c - 1 \;\mapsto\; (\tau_c - 1)^{0.7211},$$

which turns $\tau_c = 100, 30, 10$ into $28.5, 12.3, 5.9$ tokens and leaves $\tau_c = 2$ fixed; recall time constants over $[4, 256]$ become $[2.72, 54.5]$, and matched thresholds shrink by the same factor 0.7211. With the gradient, AdamW balances the decay $\mathrm{lr}\, \lambda_{\mathrm{wd}} |p|$ against a push of size $\mathrm{lr}\, \bar{c}$ for a share $\bar{c}$ of consistent steps, so $|p^*| \leq \bar{c}/\lambda_{\mathrm{wd}}$. The relaxation time $1/(\mathrm{lr}\, \lambda_{\mathrm{wd}}) = 33{,}333$ updates exceeds the schedule, so within training the contraction dominates. □

Lemma A.5 (Row constant of a linear read)**.** *Let the input have in-row mean $\boldsymbol{\mu}$. A linear read $\boldsymbol{M}\boldsymbol{x}_t$ in the gate logit adds $\boldsymbol{M}\boldsymbol{\mu}$ on every token of the row, so the constant of the row can settle in $\boldsymbol{M}$ instead of b_η; a read of the centered input $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t - \frac{1}{t-1} \sum_{s<t} \boldsymbol{x}_s$, with $\tilde{\boldsymbol{x}}_1 = \boldsymbol{0}$, has zero mean on a row whose input is stationary and carries only the per-token part.*

Proof. Write $\boldsymbol{x}_t = \boldsymbol{\mu} + \boldsymbol{\delta}_t$, where the deviations $\boldsymbol{\delta}_t$ sum to zero over the row. With a linear read $\boldsymbol{M}\boldsymbol{x}_t$ in the logit, the token projection gate being $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}$, the logit is

$$\big(b_\eta + w_\eta \mu_c + (\boldsymbol{M}\boldsymbol{\mu})_c\big) + \big(w_\eta \delta_{t,c} + (\boldsymbol{M}\boldsymbol{\delta}_t)_c\big),$$

and the second bracket has zero in-row mean, so the constant of the row is the first bracket and can settle in $\boldsymbol{M}$. With the centered input, $\tilde{\boldsymbol{x}}_t = \boldsymbol{\delta}_t - \frac{1}{t-1} \sum_{s<t} \boldsymbol{\delta}_s$ for $t \geq 2$, so $\boldsymbol{\mu}$ cancels exactly, and for a stationary input with $\mathbb{E}[\boldsymbol{\delta}_t] = \boldsymbol{0}$ also $\mathbb{E}[(\boldsymbol{M}\tilde{\boldsymbol{x}}_t)_c] = 0$. The read then has no path to the in-row constant, which can only be carried by $b_\eta + w_\eta \mu_c$. □

A.4.6 The time-constant spectrum

Lemma A.2 rules out tuning a single static retention, so the FFN input neurons launch from a time-constant spectrum with τ_c log-uniform over $[2, 100]$. Their retentions $a_c = 1 - 1/\tau_c$ then span $[0.5, 0.99]$, which contains the best constant retention $\sqrt{n/(n+1)}$ for every lag $n \leq 32$, and a fraction $\ln 10 / \ln 50 = 0.589$ of the channels has $a_c \geq 0.9$. The spectrum has to be placed rather than learned, since for a retention logit ℓ

$$\frac{\partial\big[(1-a)a^L\big]}{\partial \ell} = (1-a)\, a^L \big(L(1-a) - a\big),$$

which equals 2.0×10^{-2}, 5.7×10^{-5} and 1.8×10^{-9} at $L = 6, 16, 32$ for $a = \frac{1}{2}$. A spectrum alone silences the slow channels. At a fixed threshold the bipolar release rate without reset is

$$2\,\Phi\Big(-\frac{\vartheta}{\sigma\sqrt{\eta/(2-\eta)}}\Big),$$

which falls with η; for $\vartheta = 0.275$ and $\sigma = 1$ it is 0.634, 0.231, 0.026 and 10^{-4} at $\eta = 0.5, 0.1, 0.03, 0.01$. Matched thresholds therefore set each ϑ_c so that the channel releases at the rate of the fastest quartile of its group. Exempting one-dimensional parameters from weight decay removes the factor 0.7211 of Lemma A.4 from b_η and ϑ, so the spectrum and the matched thresholds are fixed points of the regularizer.

Figure 12 shows the bound of Proposition 4.3 against the lag, the launch distribution of the time constants under the spectrum, and the half-lives of the recall trace, which training with weight decay shrinks from $[4, 256]$ to $[2.5, 46]$ tokens, a logarithmic factor of 0.66–0.69 against the 0.72 of Lemma A.4.

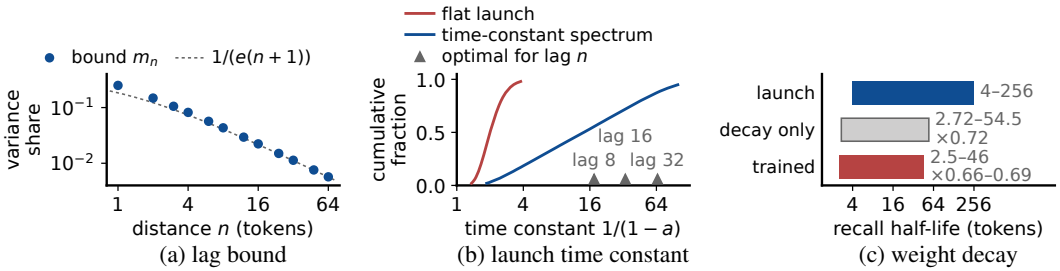


Figure 12: **Write-link derivation quantities.**

A.4.7 Scope

The bound and the release rates assume independent Gaussian inputs, a constant gate and no reset. The survival $(1 - f)^k$ is an estimate for a hard gate with perfect readout along one direction, not a bound for soft gates or higher ranks. Lemma A.4 omits the gradient, and the interaction of the projection gate with the spectrum is not derived.

A.5 Release Link: Proofs of Proposition 4.4

By Lemma 4.2, the membrane of a unit reaches the output only through the release slope Ψ'_t and the unit's readout column, and the unit's parameters are credited only through the transposed product. We prove the two statements of Proposition 4.4, that a unit that never releases closes both for good and that the bipolar release has fewer such units and keeps the energy that the one-sided release discards. We then prove Lemma A.6, by which either soft reset keeps the parallel scan that trains the block.

A.5.1 Setting

A unit follows $u_t = a_t v_{t-1} + I_t$, $S_t = R_\vartheta(u_t)$ and $v_t = u_t - S_t$ with the one-sided or the bipolar rule,

$$R^{(1)}_\vartheta(u) = (u - \vartheta)_+, \qquad R^{(2)}_\vartheta(u) = (u - \vartheta)_+ - (-u - \vartheta)_+; \tag{17}$$

the one-sided rule releases as an integrate-and-fire neuron does, and the bipolar rule as bidirectional spikes do (Xing et al., 2024). The block reads its channels through $\boldsymbol{y}_t = \sigma(\boldsymbol{W}_{\text{gate}}\boldsymbol{s}_t) \odot \boldsymbol{W}_{\text{out}}\boldsymbol{S}_t + \boldsymbol{W}_{\text{skip}}\boldsymbol{s}_t$, and the spiking gated FFN through $\boldsymbol{W}_{\text{down}}(\boldsymbol{S}^{\text{gate}}_t \odot \boldsymbol{S}^{\text{up}}_t)$. The backward pass replaces $\partial S_t/\partial u_t$ by $g^+_t = \sigma(\alpha(u_t - \vartheta))$ for the one-sided rule and by $g_t = g^+_t + g^-_t$ with $g^-_t = \sigma(\alpha(-u_t - \vartheta))$ for the bipolar rule. The reset uses the same S_t, so $\partial v_t/\partial u_t = 1 - \partial S_t/\partial u_t$ in the backward pass as well.

A.5.2 A SILENT UNIT STAYS SILENT

Proof of the first statement. Let unit j release $S_{t,j} = 0$ on every token of a batch, and let $\boldsymbol{\delta}_t = \partial\mathcal{L}/\partial \boldsymbol{y}_t$. Summed over the batch, the gradients of its readout column in the block and in the FFN are

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\text{out}}[:, j]} = \sum_t \big(\sigma(\boldsymbol{W}_{\text{gate}} \boldsymbol{s}_t) \odot \boldsymbol{\delta}_t\big)\, S_{t,j}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{\text{down}}[:, j]} = \sum_t \boldsymbol{\delta}_t\, S^{\text{gate}}_{t,j} S^{\text{up}}_{t,j}. \tag{18}$$

Every term is zero times a finite vector, which is exactly zero in IEEE arithmetic, so the sum is exactly zero in every precision and every order of summation. The surrogate enters only $\partial S/\partial u$ and not this gradient, so the statement holds for every surrogate. In the FFN the same exact zero arises when the partner unit is silent.

With a zero gradient, AdamW updates the moments as $m \leftarrow \beta_1 m$ and $v \leftarrow \beta_2 v$. If the column never had a gradient, then $m = v = 0$, the Adam direction $\hat{m}/(\sqrt{\hat{v}} + \epsilon)$ is zero, and the only change is the contraction

$$\boldsymbol{W}[:, j] \;\leftarrow\; (1 - \mathrm{lr}_k \lambda_{\text{wd}})\, \boldsymbol{W}[:, j]$$

along a fixed ray, which accumulates to $\prod_k (1 - \mathrm{lr}_k \lambda_{\text{wd}}) = 0.721$ over the schedule. If the column had gradients before the unit fell silent, its Adam direction decays element-wise as $(\beta_1/\sqrt{\beta_2})^k = 0.923^k$, to below 2% of its last value after 50 updates, and from then on the column only shrinks.

For the input side, the unit contributes $\boldsymbol{W}[:, j] S_{t,j} = \mathbf{0}$ to the output whatever the column is, so $\boldsymbol{\delta}_t$ does not depend on the column, and

$$\frac{\partial \mathcal{L}}{\partial S_{t,j}} = \boldsymbol{W}_{\text{out}}[:, j]^\top \big(\sigma(\boldsymbol{W}_{\text{gate}} \boldsymbol{s}_t) \odot \boldsymbol{\delta}_t\big),$$

or $\boldsymbol{W}_{\text{down}}[:, j]^\top \boldsymbol{\delta}_t S^{\text{up}}_{t,j}$ for an FFN gate unit, is linear in it. The carried state enters only the next membrane, so

$$\frac{\partial \mathcal{L}}{\partial v_{t,j}} = a_{t+1} \frac{\partial \mathcal{L}}{\partial u_{t+1,j}}, \qquad \frac{\partial \mathcal{L}}{\partial u_{t,j}} = g_t \frac{\partial \mathcal{L}}{\partial S_{t,j}} + (1 - g_t) \frac{\partial \mathcal{L}}{\partial v_{t,j}},$$

with g_t^+ in place of g_t for the one-sided rule. Backward induction from the last token shows that every $\partial\mathcal{L}/\partial u_{t,j}$, and with it the gradient of every input parameter of the unit (the weights that drive it, its threshold and its gate bias), is a linear function of the column. Since the column only shrinks along a fixed ray, these gradients are proportional to its norm. □

Each of these paths also carries a subthreshold surrogate factor of at most $2\sigma(-\alpha\Delta)$, where Δ is the smallest distance between the membrane and a threshold. At $\alpha = 96$ and $\Delta = 0.19$ this factor is 1.2×10^{-8}, and since the FP32 logistic function returns exactly zero below -88.72, the input-side gradient is exactly zero once $\Delta > 0.924$.

A.5.3 THE BIPOLAR RELEASE KEEPS MORE UNITS AND MORE ENERGY

Proof of the second statement. Give both rules the same drive (a_t, I_t, ϑ_t) and the same initial state, and let the bipolar unit be silent, so that $|u_t| \le \vartheta_t$ and $v_t = u_t$ on every token. If the two carried states agree at $t - 1$, the membranes agree at t, the condition $u_t \le \vartheta_t$ gives $R^{(1)}_\vartheta(u_t) = 0$, and the carried states agree at t. By induction the one-sided unit is silent as well. The converse fails at any token with $u_t < -\vartheta_t$, where the bipolar unit releases $u_t + \vartheta_t < 0$.

For the energy, let $u \sim \mathcal{N}(\mu, \sigma^2)$, $z = (\vartheta - \mu)/\sigma$ and $z' = (\vartheta + \mu)/\sigma$. Substituting $u = \mu + \sigma x$ and using $\int_z^\infty x\phi(x)\,dx = \phi(z)$ and $\int_z^\infty x^2\phi(x)\,dx = \Phi(-z) + z\phi(z)$ gives

$$\mathbb{E}\big[(u - \vartheta)_+^2\big] = \sigma^2\, T_1(z), \qquad T_1(z) = (1 + z^2)\,\Phi(-z) - z\,\phi(z), \tag{19}$$

and the negative branch gives $\mathbb{E}[(-u - \vartheta)_+^2] = \sigma^2 T_1(z')$, since $-u \sim \mathcal{N}(-\mu, \sigma^2)$. For $\vartheta \ge 0$ the two branches have disjoint supports, so the energy of the bipolar release is the sum of the two, and

the one-sided release, which keeps only the first branch, discards the fraction

$$\lambda(m, z) = \frac{T_1(z')}{T_1(z) + T_1(z')}, \qquad m = \mu/\sigma. \tag{20}$$

A symmetric drive has $m = 0$ and $z' = z$, so $\lambda(0, z) = \frac{1}{2}$ for every z, that is, for every threshold and every scale. □

For a drive of mean $-|\mu|$ and standard deviation s, silence under the one-sided rule is governed by $z_+ = (\vartheta + |\mu|)/s$ and under the bipolar rule by $z_- = (\vartheta - |\mu|)/s$. For independent Gaussian membranes a batch of $N = 8{,}184$ tokens is silent with probability $\frac{1}{2}$ at $z = 3.76$ for the one-sided rule and at $z = 3.93$ for the bipolar rule with a symmetric drive, so a mostly negative drive silences one-sided units and leaves bipolar ones active. At $\vartheta = 0.275$ and $\sigma = 0.65$, for example, the one-sided and bipolar rules transmit 0.2419 and 0.4839 of the drive energy and are silent on 0.6639 and 0.3278 of the tokens.

A.5.4 THE PARALLEL SCAN

The bipolar reset is a clamp, and the next lemma shows that clamps keep the parallel scan that trains the block.

Lemma A.6 (Clamp-affine scan)**.** *For $\vartheta \geq 0$ the reset of either release, $\min(u, \vartheta)$ or $\mathrm{clamp}(u, -\vartheta, \vartheta)$, lies in the family $v \mapsto \mathrm{clamp}(av + b, \ell, h)$ with $a \geq 0$, which is closed under composition with an associative rule.*

Proof. The one-sided reset is $v = u - (u - \vartheta)_+ = \min(u, \vartheta) = \mathrm{clamp}(u, -\infty, \vartheta)$. For the bipolar reset and $\vartheta \geq 0$, the three cases $u \geq \vartheta$ ($S = u - \vartheta$, $v = \vartheta$), $|u| < \vartheta$ ($S = 0$, $v = u$) and $u \leq -\vartheta$ ($S = u + \vartheta$, $v = -\vartheta$) give $v = \mathrm{clamp}(u, -\vartheta, \vartheta)$. One token therefore maps the carried state by

$$v \mapsto \mathrm{clamp}(a_t v + I_t, -\vartheta_t, \vartheta_t),$$

or by $v \mapsto \mathrm{clamp}(a_t v + I_t, -\infty, \vartheta_t)$ for the one-sided reset, a member of the family $\mathcal{F}$ of maps $f_{a,b,\ell,h}(v) = \mathrm{clamp}(av + b, \ell, h)$ with $a \geq 0$ and $\ell \leq h$.

Affine maps and clamps. For $a \geq 0$,

$$a\,\mathrm{clamp}(w, \ell, h) + b = \mathrm{clamp}(aw + b,\ a\ell + b,\ ah + b),$$

because a nondecreasing affine map commutes with $\min$ and $\max$; for $a = 0$ both sides equal b.

Nested clamps. For $L \leq H$ and $\ell \leq h$,

$$\mathrm{clamp}\big(\,\mathrm{clamp}(y, L, H), \ell, h\big) = \mathrm{clamp}(y, L', H')$$

with $L' = \mathrm{clamp}(L, \ell, h) \leq H' = \mathrm{clamp}(H, \ell, h)$. For $L \leq y \leq H$ the left side is $\mathrm{clamp}(y, \ell, h)$, and so is the right side, since if $y < \ell$ then $L' = \ell$ and both sides are ℓ, if $y > h$ then $H' = h$ and both are h, and otherwise $L' \leq y \leq H'$. For $y < L$ the left side is L', and so is the right side, since either $y \leq L'$, or $L' < y < L$, which forces $L > h$ and hence $L' = H' = h$. The case $y > H$ is symmetric.

Composition. By the two identities,

$$\begin{aligned} f_2 \circ f_1(v) &= \mathrm{clamp}\big(\,\mathrm{clamp}(a_2a_1v + a_2b_1 + b_2,\ a_2\ell_1 + b_2,\ a_2h_1 + b_2), \ell_2, h_2\big) \\ &= f_{a_2a_1,\ a_2b_1+b_2,\ f_2(\ell_1),\ f_2(h_1)}(v), \end{aligned}$$

with $a_2a_1 \geq 0$ and $f_2(\ell_1) \leq f_2(h_1)$ because f_2 is nondecreasing, so $\mathcal{F}$ is closed under composition. The rule on quadruples is associative, since the first two components compose as affine maps and both bracketings of three maps give the endpoints $f_3(f_2(\ell_1))$ and $f_3(f_2(h_1))$. A reset token is $(0, I_t, -\vartheta_t, \vartheta_t)$ and the identity is $(1, 0, -\infty, \infty)$, so a parallel prefix scan (Blelloch, 1990) computes every carried state of the block. □

All four fields are needed, since in the symmetric subfamily $(a, b, -\varrho, \varrho)$ the endpoints of a composition sum to $\mathrm{clamp}(b_2 - a_2\varrho_1, -\varrho_2, \varrho_2) + \mathrm{clamp}(b_2 + a_2\varrho_1, -\varrho_2, \varrho_2)$, which has magnitude $2\varrho_2$

once $|b_2| \geq \varrho_2 + a_2\varrho_1$, because the current shifts the band. For $\vartheta < 0$ the bipolar reset is a reflection, $v = -u$ on $|u| < |\vartheta|$, and leaves $\mathcal{F}$, so the block keeps $\vartheta_t \geq \vartheta_{\min} = 0.02$.

A.5.5 SCOPE

The statements are exact for single units; they do not predict how many units fall silent, which depends on the drive of each unit. The inclusion of the second statement compares the two rules on one drive, and models trained with the two rules have different drives, so their counts test the direction of the inclusion and not the inclusion itself. The energy share assumes a Gaussian drive; where a trained drive is skewed, the one-sided rule discards less. The bipolar rule does not remove the exact zeros of the FFN product caused by a silent partner, which the first statement also covers, and Lemma A.6 holds for $\vartheta \geq 0$ only.

A.6 CREDIT LINK: PROOF OF PROPOSITION 4.5

The forward pass of every neuron is exact, and the surrogate slope α (Neftci et al., 2019) only decides how the transposed product of Lemma 4.2 enters a membrane below threshold and how far it travels back. We prove the adjoint recursion that the link rests on (Lemma A.7) and Proposition 4.5, evaluate its trade-off on a slow channel, and prove the matched slope (Lemma A.8).

A.6.1 THE ADJOINT RECURSION

Lemma A.7 (Adjoint recursion)**.** *The gradient $\lambda_t = \partial\mathcal{L}/\partial u_t$ satisfies*

$$\lambda_t = g_t e_t + \rho_t \lambda_{t+1}, \qquad \rho_t = a_{t+1}(1 - g_t), \tag{21}$$

with the demand $e_t = \partial\mathcal{L}/\partial S_t$ and the backward retention ρ_t, so every gradient of the retention and threshold parameters passes through a factor bounded by some g_{t+k} and a product of the ρ_t.

Proof. For a PLIF channel with $u_t = a_t v_{t-1} + I_t$, $a_t = 1 - \eta_t$, $I_t = \eta_t x_t$ and the bipolar release, u_t affects the loss through S_t and through $v_t = u_t - S_t$, and v_t affects only u_{t+1}, whose retention and current depend on x_{t+1} alone. With $g_t = \sigma(\alpha(u_t - \vartheta)) + \sigma(\alpha(-u_t - \vartheta))$ in place of $\partial S_t/\partial u_t$ and e_t collected from the direct consumers of the release,

$$\lambda_t = g_t e_t + (1 - g_t)\,\frac{\partial\mathcal{L}}{\partial v_t} = g_t e_t + (1 - g_t)\,a_{t+1}\lambda_{t+1},$$

and unrolling with $\rho_t = a_{t+1}(1 - g_t)$ gives

$$\lambda_t = \sum_{k \geq 0} g_{t+k}\, e_{t+k} \prod_{j=0}^{k-1} \rho_{t+j}.$$

Every gradient that reaches the gate logit, $\partial\mathcal{L}/\partial z_t = \lambda_t \eta_t (1 - \eta_t)(x_t - v_{t-1})$, and hence b_η, w_η and $\boldsymbol{U}$, carries an entry factor g_{t+k} and a backward retention ρ per token. For the threshold the backward pass gives $\partial S_t/\partial\vartheta = -\partial v_t/\partial\vartheta = \sigma(\alpha(-u_t - \vartheta)) - \sigma(\alpha(u_t - \vartheta))$, whose magnitude is at most g_t, so

$$\frac{\partial\mathcal{L}}{\partial\vartheta} = \sum_t \big(\sigma(\alpha(-u_t - \vartheta)) - \sigma(\alpha(u_t - \vartheta))\big)\,\big(e_t - a_{t+1}\lambda_{t+1}\big)$$

passes through the same factor. □

A.6.2 THE FLOOR AND THE CONSERVATION

Proof of Proposition 4.5. The surrogate is even in u, and for $u > 0$

$$g'(u) = \alpha\big[\sigma'(\alpha(u - \vartheta)) - \sigma'(\alpha(u + \vartheta))\big] > 0,$$

because σ' is even and decreasing in $|x|$ and $|u - \vartheta| < u + \vartheta$. Its minimum is therefore at $u = 0$, where $g_{\min} = 2\sigma(-\alpha\vartheta)$.

Truncate the sums of the proposition at the end T of the row, so that

$$E_t = g_t + \rho_t E_{t+1}, \qquad H_t = 1 + \rho_t H_{t+1}, \qquad E_{T+1} = H_{T+1} = 0.$$

For a constant retention a, $\rho_t = a(1 - g_t)$, and $D_t = 1 - aE_t - (1 - a)H_t$ satisfies

$$D_t = 1 - ag_t - (1 - a) - \rho_t\big(aE_{t+1} + (1 - a)H_{t+1}\big) = \rho_t - \rho_t(1 - D_{t+1}) = \rho_t D_{t+1},$$

so $D_t = \prod_{j=t}^{T} \rho_j$ from $D_{T+1} = 1$, for every sequence g_t. □

For $g_t \equiv \bar{g}$ and a long row the two quantities are

$$H = \frac{1}{(1 - a) + a\bar{g}}, \qquad E = \bar{g}H = \frac{\bar{g}}{(1 - a) + a\bar{g}}. \tag{22}$$

At $\alpha = 8$ the floor exceeds 0.05 unless $\vartheta > \ln 39/8 = 0.458$, at $\alpha = 4$ unless $\vartheta > 0.916$, and at $\alpha = 96$ only for $\vartheta < 0.038$. An exact zero in FP32 needs $\alpha(\vartheta - |u|) \geq 88.72$, that is $\vartheta \geq 0.924$ at $\alpha = 96$.

A.6.3 The trade-off on a slow channel

In the stationary case equation 22 the entry rises with $\bar{g}$,

$$\frac{dE}{d\bar{g}} = \frac{1 - a}{\big((1 - a) + a\bar{g}\big)^2} > 0,$$

and is half saturated at $\bar{g} = (1 - a)/(2 - a)$, while H falls and the credit beyond lag k, a fraction ρ^{k+1}, collapses. Since $\bar{g}$ grows as α decreases, a smaller slope buys entry at the price of horizon, and the conservation makes the exchange exact, since the entry gained equals the backward horizon lost, in units of the forward horizon $1/(1 - a)$. Table 3 evaluates both for a slow channel with $a = 0.98$ and $\vartheta = 0.27$ at its floor. From $\alpha = 96$ to 16 the entry rises from 0 to 0.57 and the horizon falls from 50 to 22 tokens; from 16 to 8 the entry reaches 0.93 and the horizon 4.5; from 8 to 4 the entry gains only 5% while the horizon falls by another factor of 2.3 and the far credit ρ^{11} by a factor of 190. Above $\alpha = 8$ a slow channel is limited by entry and below it by horizon, so the loss should first fall and then rise as the slope decreases, with the turn between 16 and 4. Appendix A.9.1 checks the conservation and the stationary forms in simulation.

Table 3: **Entry and horizon of a slow PLIF channel.**

α	$\bar{g} = 2\sigma(-\alpha\vartheta)$	$\rho = a(1 - \bar{g})$	$H = 1/(1 - \rho)$	$E = \bar{g}H$	ρ^{11}
96	1.1×10^{-11}	0.980	50.0	5.5×10^{-10}	0.80
32	3.5×10^{-4}	0.980	49.1	0.017	0.80
16	0.026	0.954	21.9	0.57	0.60
8	0.207	0.777	4.49	0.93	0.063
4	0.507	0.483	1.93	0.98	3.3×10^{-4}

Stationary approximation for $a = 0.98$ and $\vartheta = 0.27$.

A.6.4 The matched slope

Lemma A.8 (Matched slope)**.** *Under Gaussian membrane noise of scale s, the expected derivative of the bipolar release is*

$$\mathbb{E}\big[\mathbf{1}\{|u + \xi| > \vartheta\}\big] = \Phi\Big(\frac{u - \vartheta}{s}\Big) + \Phi\Big(\frac{-u - \vartheta}{s}\Big), \tag{23}$$

which the surrogate matches term by term within 0.0095 at $\alpha^* = 1.702/s$.

Proof. Under Gaussian noise ξ of scale s, the release slope $\mathbf{1}\{|u+\xi| > \vartheta\}$ has the expectation equation 23. The logistic function matches the Gaussian distribution function to

$$\max_x \big|\sigma(1.702\,x) - \Phi(x)\big| = 0.0095,$$

so each term of the surrogate matches the corresponding term to within 0.0095 at $\alpha^* = 1.702/s$. $\square$

Taking the membrane standard deviation averaged over the PLIF neurons of the network trained at $\alpha = 96$, $\bar{\sigma}_V = 0.2125$, gives $\alpha^* = 1.702/0.2125 = 8.0$. On slow channels this slope is a deliberately wide surrogate ($\alpha s \approx 0.3$ for $s \approx 0.04$), valued for its entry rather than for matching the expected release. The block keeps $\alpha = 96$, because with thresholds near 0.75 its subthreshold surrogate $2\sigma(-72) \approx 10^{-31}$ already leaves the backward retention $\rho_t = a_{t+1}(1-g_t)$ equal to a_{t+1}, which a wider surrogate can only shorten.

A.6.5 SCOPE

The derivation explains a loss that first falls and then rises as the slope decreases and places the turn between 16 and 4. It does not locate the minimum, and $\alpha^* = 8.0$ depends on taking the membrane standard deviation as the noise scale and on averaging over sites.

A.7 READOUT LINK: PROOFS OF PROPOSITION 4.6

A token that the blocks write and release reaches the prediction only through the readout of the memory layers, the last factor of Lemma 4.2. We prove that the readout of a wide head is dominated by a component that is constant over rows and tokens and so carries no token information (Proposition 4.6). We then prove that a path added to the contents with a coefficient launched at zero does not start, whereas one launched at an anchor carries gradients from the first update and, exempt from weight decay, leaves the anchor only under a consistent gradient (Lemma A.9); the proof is written for the value residual and uses no property of the added contents.

A.7.1 SETTING

A head reads

$$\boldsymbol{r}_t = \sum_{j \le t} w_{t,j}\, \tau_j\, \boldsymbol{m}_j, \qquad w_{t,j} \ge 0, \qquad \sum_j w_{t,j} = 1, \tag{24}$$

as in equation 2. The contents are $\boldsymbol{m}_j = \mathrm{SiLU}(\boldsymbol{\xi}_j)$ (Ramachandran et al., 2017), where $\boldsymbol{\xi}_j$ is the convolved synaptic projection of the layer input. After the two heads are concatenated, a zero-centered RMSNorm, a gate and an output projection carry $\boldsymbol{r}_t$ to the residual stream. The value residual (Zhou et al., 2025) of the n-th later memory layer replaces $\boldsymbol{m}_j^{(n)}$ by $\boldsymbol{m}_j^{(n)} + \boldsymbol{\lambda}_n \odot \boldsymbol{m}_j^{(0)}$, where $\boldsymbol{m}_j^{(0)}$ are the contents of the first memory layer; AdamW applies the weight decay $\omega = \lambda_{\mathrm{wd}} = 0.1$ to matrices only, so $\boldsymbol{\lambda}_n$, a vector, is not decayed.

A.7.2 THE CONSTANT SHARE

Proof of Proposition 4.6. Take the tagging constant over the events, $\tau_j = \tau$; otherwise the argument applies to $\tau_j \boldsymbol{m}_j$. Decompose the contents of row b at position j as

$$\boldsymbol{m}_{b,j} = \boldsymbol{\mu} + \boldsymbol{\rho}_b + \boldsymbol{\epsilon}_{b,j},$$

with $\boldsymbol{\mu}$ the mean over rows and tokens, $\boldsymbol{\rho}_b$ the mean of row b minus $\boldsymbol{\mu}$, and $\boldsymbol{\epsilon}_{b,j}$ the deviation of the token from its row mean. The rows average $\boldsymbol{\rho}_b$ to zero and each row averages $\boldsymbol{\epsilon}_{b,j}$ to zero, so the mean square of the contents splits as

$$\mathbb{E}\|\boldsymbol{m}\|^2 = \|\boldsymbol{\mu}\|^2 + \mathbb{E}\|\boldsymbol{\rho}_b\|^2 + \mathbb{E}\|\boldsymbol{\epsilon}\|^2,$$

and dividing by $\mathbb{E}\|\boldsymbol{m}\|^2$ gives the shares $\pi_{\mathrm{g}} + \pi_{\mathrm{r}} + \pi_{\mathrm{t}} = 1$ of the proposition. Because the weights sum to one, $\boldsymbol{r}_t = \tau(\boldsymbol{\mu} + \boldsymbol{\rho}_b + \boldsymbol{d}_t)$ with $\boldsymbol{d}_t = \sum_j w_{t,j}\boldsymbol{\epsilon}_{b,j}$. Assume that the weights are uncorrelated with the deviations, and that the deviation at every position has zero mean and is uncorrelated with the row mean. Then $\mathbb{E}\boldsymbol{d}_t = \boldsymbol{0}$ and $\boldsymbol{d}_t$ is uncorrelated with $\boldsymbol{\mu}$ and $\boldsymbol{\rho}_b$, so

$$\mathbb{E}\|\boldsymbol{r}_t\|^2 = \tau^2\big(\|\boldsymbol{\mu}\|^2 + \mathbb{E}\|\boldsymbol{\rho}_b\|^2 + \mathbb{E}\|\boldsymbol{d}_t\|^2\big), \qquad \mathbb{E}\|\boldsymbol{d}_t\|^2 = \sum_{j,k} w_{t,j} w_{t,k}\, \mathbb{E}[\boldsymbol{\epsilon}_j^\top \boldsymbol{\epsilon}_k] = \kappa\, \mathbb{E}\|\boldsymbol{\epsilon}\|^2,$$

where $\kappa = \sum_{j,k} w_{t,j} w_{t,k} \varrho_{jk}$ and $\varrho_{jk} = \mathbb{E}[\boldsymbol{\epsilon}_j^\top \boldsymbol{\epsilon}_k]/\mathbb{E}\|\boldsymbol{\epsilon}\|^2$ is the correlation of the deviations of events j and k. The component $\tau\boldsymbol{\mu}$ is the same for every row and token, and its share of the mean square of the readout is

$$c = \frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\mu}\|^2 + \mathbb{E}\|\boldsymbol{\rho}_b\|^2 + \kappa\,\mathbb{E}\|\boldsymbol{\epsilon}\|^2} = \frac{\pi_{\mathrm{g}}}{\pi_{\mathrm{g}} + \pi_{\mathrm{r}} + \pi_{\mathrm{t}}\kappa}. \tag{25}$$

If neighboring deviations are not negatively correlated, the off-diagonal part of κ is nonnegative and $\kappa \geq \sum_j w_{t,j}^2 = 1/N_2$, and since c decreases in κ,

$$c \leq \frac{\pi_{\mathrm{g}}}{\pi_{\mathrm{g}} + \pi_{\mathrm{r}} + \pi_{\mathrm{t}}/N_2}.$$

The contents have $\pi_{\mathrm{g}} > 0$ even for centered inputs. Write

$$\mathrm{SiLU}(\xi) = \xi\,\sigma(\xi) = \xi\big(\sigma(\xi) - \tfrac{1}{2}\big) + \tfrac{1}{2}\,\xi.$$

The first term is nonnegative for every ξ and vanishes only at $\xi = 0$, because $\sigma(\xi) - \frac{1}{2}$ has the sign of ξ. Hence $\mathbb{E}[\mathrm{SiLU}(\xi)] > 0$ whenever $\mathbb{E}\xi = 0$ and ξ is not identically zero, so every channel of the contents has a nonzero mean and $\boldsymbol{\mu} \neq \mathbf{0}$. □

The constant leaves the Jacobian $\partial \boldsymbol{r}_t/\partial \boldsymbol{m}_j = w_{t,j}\tau \boldsymbol{I}$ unchanged, but after the RMSNorm it occupies the share c of a fixed mean square, so the part of the readout that varies over rows and tokens keeps the amplitude $\sqrt{1-c}$. For $\xi \sim \mathcal{N}(0, s^2)$ the mean share $(\mathbb{E}\,\mathrm{SiLU}(\xi))^2/\mathbb{E}\,\mathrm{SiLU}(\xi)^2$ is 0.048, 0.078, 0.120 and 0.173 at $s = 0.5, 0.7, 1, 1.5$, whereas a linear content only inherits the constant of its input. Appendix A.9.3 compares the bound with the constant share measured on trained wide heads.

A.7.3 Paths launched at zero

Lemma A.9 (Paths launched at zero)**.** *Let a path enter the contents of a later layer with a coefficient, $\boldsymbol{m}^{(n)} \leftarrow \boldsymbol{m}^{(n)} + \boldsymbol{\lambda}_n \odot \boldsymbol{m}'$. While $\boldsymbol{\lambda}_n = \mathbf{0}$, every other parameter follows the trajectory of the network without the path, so nothing aligns the per-class terms of $\partial\mathcal{L}/\partial\boldsymbol{\lambda}_n$, which may cancel. With $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_n^0 + \boldsymbol{\delta}_n$, $\boldsymbol{\delta}_n$ launched at zero and exempt from weight decay, AdamW gives*

$$\boldsymbol{\delta}_n(T) = \bar{\boldsymbol{p}}\,\Sigma_T, \tag{26}$$

with $\bar{\boldsymbol{p}}$ the sign consistency of the Adam steps and Σ_T the summed learning rate, so the coefficient leaves the anchor $\boldsymbol{\lambda}_n^0$ only as far as its steps agree in sign, whereas a decay at the rate ω would shrink $\boldsymbol{\lambda}_n^0$ itself by the factor $P_T = \prod_{k<T}(1 - \mathrm{lr}_k\omega)$.

Proof. The mixing is element-wise, so the gradient of the mixing weight is exactly

$$\frac{\partial\mathcal{L}}{\partial\lambda_{n,c}} = \sum_{t,b} \frac{\partial\mathcal{L}}{\partial m_{t,b,c}^{(n)}}\, m_{t,b,c}^{(0)}.$$

If $\boldsymbol{\lambda}_n = \mathbf{0}$ at every update, every activation equals that of the network without the residual, and so does the gradient of every other parameter θ, because the only new path, $\boldsymbol{\lambda}_n \odot \boldsymbol{m}^{(0)}$, has the derivative $\boldsymbol{\lambda}_n \odot \partial\boldsymbol{m}^{(0)}/\partial\theta = \mathbf{0}$ and changes no activation. With the same data and the same optimizer states, every downstream parameter then follows the trajectory of the network without the residual, and the output projections and the contents of deeper layers never adapt to $\boldsymbol{m}^{(0)}$. Splitting the loss by target class k writes the gradient as a sum of class terms,

$$\frac{\partial\mathcal{L}}{\partial\lambda_{n,c}} = \sum_k \sum_{t\in k} \frac{\partial\mathcal{L}}{\partial m_{t,c}^{(n)}}\, m_{t,c}^{(0)},$$

evaluated on a network that makes no use of $\boldsymbol{m}^{(0)}$, so nothing aligns these terms and they may cancel. With $\boldsymbol{\lambda}_n^0 \neq \mathbf{0}$, instead, the downstream parameters receive gradients through $\boldsymbol{\lambda}_n^0 \odot \boldsymbol{m}^{(0)}$ from the first update and adapt to it.

With $\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_n^0 + \boldsymbol{\delta}_n$ and no decay on $\boldsymbol{\delta}_n$, the update of one element is

$$\delta_{k+1} = \delta_k + \mathrm{lr}_k\, p_k, \qquad \delta_0 = 0,$$

with the Adam step $p_k = -\hat{m}_k/(\sqrt{\hat{v}_k} + \epsilon)$, for which Cauchy–Schwarz on the two moment sums gives $|p_k| \le (1-\beta_1)/\sqrt{(1-\beta_2)(1-\beta_1^2/\beta_2)} = 1.165$ in the steady state. Its solution is $\delta_T = \sum_{k<T} \mathrm{lr}_k p_k$, and with $\Sigma_T = \sum_{k<T} \mathrm{lr}_k$ and the weighted mean of the steps $\bar{p} = \sum_k \mathrm{lr}_k p_k / \Sigma_T$, which measures their sign consistency, it is $\delta_T = \bar{p}\,\Sigma_T$, which is equation 26. A decoupled decay at the rate ω would act on the whole coefficient,

$$\lambda_{k+1} = (1 - \mathrm{lr}_k\omega)\,\lambda_k + \mathrm{lr}_k\, p_k, \qquad \lambda_T = \lambda^0 P_T + \sum_{k<T} q_k p_k, \qquad q_k = \mathrm{lr}_k \prod_{l=k+1}^{T-1} (1 - \mathrm{lr}_l\omega),$$

so the launch value would shrink by the factor $P_T = e^{-\omega\Sigma_T}(1 + O(\omega^2 \sum_k \mathrm{lr}_k^2))$. □

On the schedule Σ_T is 1.3047 at update $4{,}976$ and 3.2694 at the end, and a decay at $\omega = \lambda_{\mathrm{wd}} = 0.1$ would give $P_T = 0.721$, so a decayed launch value would lose 0.14, 0.28 and 0.56 of $\lambda^0 = 0.5, 1, 2$ to decay alone. When the direction of the steps does not persist, the Adam direction behaves as unit-variance noise and the sum over the updates has standard deviation $(\sum_k \mathrm{lr}_k^2)^{1/2}$, so a launch at zero stays near the noise floor $\mathbb{E}|\lambda_{n,c}| \approx \sqrt{2/\pi}\,(\sum_k \mathrm{lr}_k^2)^{1/2} = 0.015$ at update $4{,}976$, and an anchored coefficient stays as close to its anchor.

A.7.4 Scope

Proposition 4.6 gives an upper bound and an order, not a point value, because κ contains the correlation of the deviations within a row and any content selection of the weights, and the bound needs nonnegative off-diagonal correlations. The readout also has a larger row share than the contents, because causal averaging weights early events more; the derivation treats the constant share only. Lemma A.9 shows that $\boldsymbol{\lambda}_n = \mathbf{0}$ is a near-stationary point of the mixing weights along the trajectory of the network without the residual; it does not show that it is a saddle of the joint objective. The lemma leaves out the per-token gate, whose weights are decayed as matrices, and the depth profile of the anchors is not an output of the derivation.

A.8 Secondary Priors

The secondary components of NeuronSpark (Section 4.3) each act on one factor of Lemma 4.2. The recall trace and the power-law history current add write paths to the retention product of the block, the recall readout gate acts on the release slope, and the extra patterns and the parameterization of token projections act on the readout.

A.8.1 The block horizon

The block retention is $\boldsymbol{a}_t = \sigma(\boldsymbol{W}_{\mathrm{decay}}\boldsymbol{s}_t) \odot \sigma(\boldsymbol{b})$ with $\sigma(b_c) = 2^{-1/\tau_c}$, $\tau_c \in [1, 64]$, and a gate without bias (Appendix A.1). Let channel c have the gate logit z_t and the mean gate $\bar{g}_c = \mathbb{E}_t\,\sigma(z_t)$. At launch $\bar{g}_c = \frac{1}{2}$, because the gate weights are drawn symmetrically and $\sigma(z) + \sigma(-z) = 1$. Holding the gate near 0.9 would need a logit of about 2.2 on every token, a constant that without a bias only the mean input direction can supply. Since $\ln$ is concave, $\bar{g}_c \le \frac{1}{2}$ gives $\mathbb{E}\ln\sigma(z_t) \le \ln\bar{g}_c \le -\ln 2$, so the geometric-mean retention over d tokens is at most $(\bar{g}_c\, 2^{-1/\tau_c})^d$ and the half-life obeys

$$h_c = \frac{\ln 2}{-\mathbb{E}\ln a_{t,c}} \le \frac{\ln 2}{\ln 2/\tau_c - \ln\bar{g}_c}, \tag{27}$$

which equals $\tau_c/(\tau_c+1) < 1$ at $\bar{g}_c = \frac{1}{2}$.

Let $S_t^{(h)}$ be the release recomputed from a zero state at token $t-h+1$. The membranes differ there by $a_{t-h+1}v_{t-h}$ with $|v_{t-h}| \le \vartheta_{t-h}$, and the clamp and the release are 1-Lipschitz, so the difference

is multiplied by at most a_s at every later token s and

$$\left|S_t - S_t^{(h)}\right| \le \Big(\prod_{j=t-h+1}^{t} a_j \Big)\, \vartheta_{t-h};$$

likewise $\partial u_t / \partial I_{t-d} = \prod_{j=t-d+1}^{t} a_j$ in a window without releases. With the gate at $\frac{1}{2}$ every retention is at most $a_{\max} = 2^{-1}2^{-1/64} = 0.4946$, and $a_{\max}^d$ is 0.060, 3.6×10^{-3} and 1.3×10^{-5} at $d = 4, 8, 16$.

A.8.2 The recall trace

The recall trace keeps $\boldsymbol{M}_t = \operatorname{diag}(\boldsymbol{\lambda})\boldsymbol{M}_{t-1} + \hat{\boldsymbol{k}}_t \boldsymbol{c}_t^\top$ with unit keys $\hat{\boldsymbol{k}}_t \in \mathbb{R}^r$ and per-slot decays $\lambda_n = 2^{-1/\tau_n}$, τ_n from 4 to 256 tokens, and its readout $\boldsymbol{o}_t = \boldsymbol{M}_t^\top \hat{\boldsymbol{k}}_t$ enters the current through a map $\boldsymbol{W}_{\mathrm{rec}}$ initialized at zero. Unrolling gives

$$\boldsymbol{o}_t = \sum_{s \le t} \psi_{t,s}\, \boldsymbol{c}_s, \qquad \psi_{t,s} = \sum_{n=1}^{r} \hat{k}_{t,n}\, \lambda_n^{t-s}\, \hat{k}_{s,n}, \qquad |\psi_{t,s}| \le \lambda_{\max}^{t-s}, \tag{28}$$

where the bound follows from Cauchy–Schwarz for unit keys. A past content is therefore weighted by the agreement of its key with the current one, and the factor can reach 0.978 at lag 8 and 0.917 at lag 32, against 3.6×10^{-3} at lag 8 for the block membrane. The gradient of the zero-initialized map, $\sum_t (\partial \mathcal{L}/\partial \boldsymbol{I}_t)\boldsymbol{o}_t^\top$, is nonzero from the first update, because $\boldsymbol{o}_t$ does not depend on $\boldsymbol{W}_{\mathrm{rec}}$.

A.8.3 The slope of a weak current and the recall readout gate

Write the release as $S_\vartheta(u) = u - \operatorname{clamp}(u, -\vartheta, \vartheta)$. Since the clamp has slope 1 inside the band and 0 outside,

$$S_\vartheta(u+\delta) - S_\vartheta(u) = \int_0^\delta \mathbf{1}\{|u+s| > \vartheta\}\, ds$$

almost everywhere. Let u have a density ρ symmetric about zero and let δ be independent of u. Taking expectations and exchanging the integrals gives

$$\mathbb{E}\big[S_\vartheta(u+\delta) - S_\vartheta(u)\big] = \int_0^\delta \Pr(|u+s| > \vartheta)\, ds = p\,\delta + O(\delta^3), \qquad p = \Pr(|u| > \vartheta),$$

because $\frac{d}{ds}\Pr(|u+s| > \vartheta) = \rho(\vartheta - s) - \rho(-\vartheta - s)$ vanishes at $s = 0$. For a Gaussian current e of any size, independent of the rest w of the membrane $u = w + e$, Stein's lemma $\mathbb{E}[e\, f(w+e)] = \mathbb{E}[e^2]\,\mathbb{E}[f'(w+e)]$ with $f = S_\vartheta$ gives exactly

$$\frac{\mathbb{E}\big[(S_\vartheta(u) - S_\vartheta(u-e))\, e\big]}{\mathbb{E}[e^2]} = \Pr(|u| > \vartheta).$$

By Lemma 4.2 the readout of the trace thus reaches the output only as

$$\frac{\partial \boldsymbol{y}_t}{\partial \boldsymbol{o}_t} = \operatorname{diag}\big(\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t)\big)\, \boldsymbol{W}_{\mathrm{out}}\, \operatorname{diag}(\boldsymbol{\Psi}_t')\, \boldsymbol{W}_{\mathrm{rec}},$$

masked by $\boldsymbol{\Psi}_t'$ with mean p, whereas the block input also has the unmasked path $\boldsymbol{W}_{\mathrm{skip}}$. The recall readout gate adds the readout to the gate logit, $\sigma(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t + \boldsymbol{R}\boldsymbol{o}_t)$ with $\boldsymbol{R}$ initialized at zero, which adds $\operatorname{diag}\big(\sigma'(\boldsymbol{W}_{\mathrm{gate}}\boldsymbol{s}_t + \boldsymbol{R}\boldsymbol{o}_t) \odot \boldsymbol{W}_{\mathrm{out}}\boldsymbol{S}_t\big)\boldsymbol{R}$ to this Jacobian, a per-channel rescaling of the block's own output that vanishes when the block does not release.

A.8.4 Wide heads and extra patterns

The eligibility trace accumulates $\bar{\ell}_s = \frac{1}{256}\sum_c \log \bar{a}_{s,c}$ per token,

$$b_{t,j} = \gamma_h \sum_{j < s \le t} \bar{\ell}_s.$$

At launch the decay projection of the memory layer is zero, so $\log \bar{a}_{s,c} = -\ln 2 \cdot e^{A_c}$ does not depend on the token, and $\bar{\ell}_s = -0.1118$ for the time constants e^{-A_c} log-spaced over $[1, 512]$, a mean dominated by the shortest channels. Ignoring the pattern match, the weights of the wide head, $\gamma_h = 0.03$, then fall with an e-folding lag of about 300 tokens in every memory layer. Over heads and tokens the log-decay $\gamma_h \bar{\ell}_s$ has rank one, so a wide head changes its horizon only through the shared per-token decay, which moves the sharp head as well, and γ_h, a frozen launch value plus a decayed offset, returns to 0.03 unless the loss pushes it away. By Proposition 4.6 the deep wide heads then read a constant, a row-level component and a token term reduced by κ. Extra patterns act at this point. Four further patterns per wide head form their own weights over the same keys, tagged contents and eligibility trace and add their readouts through a projection initialized at zero, so the layer equals the base layer at launch and gains four readout directions per token.

A.8.5 Zero-initialized token projections learn a constant first

Let a scalar carry a zero-initialized token projection, $s_t = \lambda + \boldsymbol{w}^\top \boldsymbol{z}_t$, as the per-token gate of the value residual does, and let $\boldsymbol{\mu} = \mathbb{E}\boldsymbol{z}_t \neq \mathbf{0}$, since RMSNorm outputs are not centered. With $\chi_t = \partial \mathcal{L} / \partial s_t$,

$$\frac{\partial \mathcal{L}}{\partial \lambda} = T\bar{\chi}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = T\bar{\chi}\,\boldsymbol{\mu} + \sum_t \chi_t (\boldsymbol{z}_t - \boldsymbol{\mu}),$$

whose first term copies the gradient of the scalar along $\boldsymbol{\mu}$. Adam moves every element by lr in its first update, so $\Delta\lambda = -\mathrm{lr}\ \mathrm{sign}(\bar{\chi})$ while

$$\Delta(\boldsymbol{w}^\top \boldsymbol{\mu}) = -\mathrm{lr} \sum_i \mathrm{sign}\left(\frac{\partial \mathcal{L}}{\partial w_i}\right) \mu_i,$$

which is $-\mathrm{lr}\ \mathrm{sign}(\bar{\chi}) \|\boldsymbol{\mu}\|_1$ when the constant term dominates every element; with $\|\boldsymbol{z}_t\|^2 = D = 512$ and a global share of 0.096, $\|\boldsymbol{\mu}\|_1 \approx \sqrt{2D/\pi}\,\|\boldsymbol{\mu}\|_2 = 127$. If the loss depends on $(\lambda, \boldsymbol{w})$ only through $\lambda + \boldsymbol{w}^\top \boldsymbol{\mu}$, the least-norm solution is $\boldsymbol{w} = \nu\boldsymbol{\mu}$, $\lambda = \nu$, and the scalar carries only $1/(1 + \|\boldsymbol{\mu}\|^2) = 0.02$ of the constant. The projection therefore first learns $\boldsymbol{w}^\top \mathbb{E}[\boldsymbol{z}_t]$, a faster copy of the scalar. Causal centering, $\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t - \frac{1}{t}\sum_{j<t} \boldsymbol{z}_j$, removes the term $T\bar{\chi}\boldsymbol{\mu}$ on stationary rows and leaves the constant to the scalar, and at $\boldsymbol{w} = \mathbf{0}$ both inputs give the same forward pass; the per-token gate of the value residual reads $\tilde{\boldsymbol{z}}_t$ for this reason (Appendix A.1).

A.8.6 The power-law history current

The block adds $\boldsymbol{Q}\boldsymbol{h}_t$ to its current, with

$$\boldsymbol{h}_t = \sum_{d=1}^{t} w_{t,d}\, \boldsymbol{P}\boldsymbol{s}_{t-d}, \qquad w_{t,d} = \frac{d^\alpha - (d-1)^\alpha}{t^\alpha}, \tag{29}$$

for a learnable exponent $\alpha \in (0, 1]$, not the surrogate slope. The weights are nonnegative and telescope to $\sum_d w_{t,d} = 1$, so $\boldsymbol{h}_t$ is a convex combination of strictly past inputs, the row mean at $\alpha = 1$ and the previous token as $\alpha \to 0$. The most recent n tokens hold the weight $(n/t)^\alpha$, so half the weight lies within $t\,2^{-1/\alpha}$ tokens, and summation by parts gives the mean lag

$$t - t^{-\alpha} \sum_{m=1}^{t-1} m^\alpha \ \to\ \frac{\alpha}{\alpha + 1}\, t,$$

so the horizon grows with the position and has no time constant. At $\alpha = \frac{1}{2}$ and $t = 1{,}023$ the half-weight lag is 256 tokens and $\sum_d w_{t,d}^2 = O(\log t / t)$ gives $N_2 = 375$. If the inputs have a constant-plus-row share of 0.12 of their mean square, the token share of $\boldsymbol{h}_t$ is $0.88 \sum_d w_{t,d}^2 / (0.12 + 0.88 \sum_d w_{t,d}^2)$, which is 6.4%, 3.5% and 1.9% at $t = 256, 512, 1{,}023$. The current is therefore mostly a row-level bias with a horizon as long as the row, and it enters before the release, so by Section A.8.3 it reaches the output with the slope p.

A.8.7 THE CONTENT PRODUCT TERM

The components above change the weights $w_{t,j}$ or add readout directions, but none changes the contents. The contents are one nonlinearity of the layer input, whose context component varies slowly over the row. The content product term adds $\boldsymbol{U}_m(\mathrm{SiLU}(\boldsymbol{A}_m\boldsymbol{z}_j) \odot \boldsymbol{B}_m\boldsymbol{z}_j)$ to the contents of the wide head, a gated product of two projections of the layer input, so that the contents can carry products of token and context features.

A.8.8 SCOPE

The half-life bound needs a mean gate of at most $\frac{1}{2}$. Interference limits the trace. For independent isotropic keys, white contents and a write on every token, its steady-state signal-to-interference ratio at lag l is $r\big(\frac{1}{r}\sum_n \lambda_n^l\big)^2 / \big(\frac{1}{r}\sum_n (1-\lambda_n^2)^{-1}\big)$, linear in r and 0.29 at lag 8 for $r = 24$ at launch, so the trace serves recent inputs. Whether the extra patterns, the history current or the product term lower the loss is not predicted by these derivations.

A.9 EXTENDED EXPERIMENTAL PROTOCOL AND DIAGNOSTICS

This section gives the setup and the measured outcome of every test summarized in Table 2. The first experiments check the release identity and the four links in simulation; the later experiments test the links on 200M models and the full model across scales. Each result states its seeds or samples, and paired differences are read against the noise floor of Appendix A.9.6.

Panel index. Main-text panels map to their detailed checks as follows: Figures 3, 4 and 5 to Appendices A.9.1, A.2 and A.3; Figure 6(a) to Appendix A.9.2, and (b, c, d) to Appendices A.9.3 and A.4; Figure 7(a, b) to Appendices A.9.3 and A.5, and (c, d) to Appendices A.9.3 and A.6; Figure 8(a, b, c, d) to Appendices A.9.3 and A.7; and Figures 9 and 10 with Table 1 to Appendices A.9.4 and A.10; the energy results are in Appendix A.9.5.

A.9.1 EXPERIMENT I: THE RELEASE IDENTITY AND THE FOUR LINKS IN SIMULATION

Setup. Figure 3 simulates one neuron with threshold $\vartheta = 1$ and a membrane starting at zero (Appendix A.2): (a) $1{,}000$ neurons driven by Gaussian currents of unit variance around means drawn uniformly from $[-0.5, 0.5]$, with the largest rate error over the neurons; (b) a constant current held for $2{,}000$ steps; (c) $4{,}096$ steps of the currents $1.2, -0.4, -0.4, -0.4$, grouped in runs of 8 and 24, interleaved, or in 64 random orders, whose median is drawn. Figure 4 differentiates 60 random bipolar PLIF blocks with 8 inputs, 16 channels, 8 outputs and 48 tokens, Gaussian weights of variance $1/\text{fan-in}$ ($0.36/\text{fan-in}$ at the input), retentions uniform on $[0.2, 0.9]$, thresholds uniform on $[0.5, 1.2]$ per token and channel and scaled by a common factor from 4 to 0.45 to set the release rate, and standard Gaussian inputs, at the last token t and every $q < t$: (a) the share of the Jacobians $\partial \boldsymbol{y}_t / \partial \boldsymbol{s}_q$ that a product misses by more than 10^{-6} against central differences with step 10^{-6} through the current, skipping pairs with a membrane within 10^{-4} of its threshold, for the lemma's product, the product without the clamp factor at $r = q$ and the product without clamp factors; (b) the same shares against automatic differentiation of the surrogate block at slopes 1 to 128, and the share of pairs on which the surrogate Jacobian departs from the exact one; (c) the share of channel paths from q to t that no reset cuts at release rates 0.04, 0.21 and 0.41, against $(1-p)^{t-q}$ for independent releases at rate p. Figure 5 simulates (a) 4×10^5 steps of a static gate on independent Gaussian inputs, (b) 4×10^6 Gaussian drives per threshold, (c) $200{,}000$ tokens of a PLIF channel with retention 0.98, threshold 0.27 and a Gaussian current that gives the membrane a standard deviation of 0.04, whose exact backward recursions for E_t and H_t at slopes 2 to 128 are averaged over the tokens and drawn against the stationary forms with the mean surrogate of a Gaussian membrane of the same scale, and (d) $2{,}000$ rows of 512 events and 8 channels whose contents are SiLU or linear functions of Gaussian inputs, with no row offset or with row offsets of standard deviation 0.6 and event deviations of 0.8, each row read once with softmax weights on independent Gaussian scores.

Result. At retention one the rate error of the bipolar neuron stays below the bound $2\vartheta/T$ of Proposition 4.1, the leaky neurons keep rate errors of 0.06 at $a = 0.9$ and 0.26 at the median retention of the trained blocks, 0.43, the release follows the dead zone to 2×10^{-16}, and every order of the same currents releases the input rate at $a = 1$ (Figure 3). At every release rate from 0.002 to 0.52 the product has a largest error of 3×10^{-10} against central finite differences of the exact block,

dropping the clamp factor at the input token misses up to 7% of the Jacobians (23% with the surrogate), and dropping every clamp factor misses up to 46%; with the surrogate it matches automatic differentiation at every slope from 1 to 128, while the surrogate gradient itself departs from the exact derivative on 12 to 27% of the pairs; and the resets cut all but 4% of the channel paths over 8 tokens at a release rate of 0.41, fewer than independent releases would cut because quiet channels stay quiet (Figure 4). In the closed forms, a static retention keeps at most $m_n \approx e^{-1}/(n+1)$ of the variance for a token at lag n, $aE + (1-a)H = 1$ holds to 1.6×10^{-15}, and the constant share of SiLU contents reaches 0.99 at $N_2 = 512$ (Figure 5). On $2{,}000$ random clamp-affine triples including reset tokens, the closure of Lemma A.6 holds to 3×10^{-7} in FP32 and to 9×10^{-16} in FP64, and the rule on quadruples is associative.

A.9.2 Experiment II: where the difference forms

Setup. Figure 6(a) scores a 200M model without the write, credit and readout priors and the Transformer on the same $61{,}440$ targets at positions from 64 on, each with a cold window of its last 1 to 32 tokens, and keeps the 62% whose target does not occur in the window. By target class, copy targets are those whose bigram occurred earlier in the row, seen targets occurred earlier with another bigram, and new targets did not occur earlier in the row.

Result. The same probe splits the deficit into the order of the tokens 2 to 4 back, from which the Transformer draws 0.07 to 0.09 nats more, and the identity of the tokens 6 to 32 back, from which the spiking network draws 9 to 19% less per token. On seen targets this network trails the Transformer by 0.098 nats, a difference that the final model closes (Figure 10c).

A.9.3 Experiment III: the four links on 200M models

Setup. Figure 6(b) computes the cancellation ratio $1 - |\sum_t \gamma_t| / \sum_t |\gamma_t|$ of the retention bias of every channel on one batch of $2 \times 1{,}023$ tokens of two 200M models at update $5{,}000$, one with the static gate and one with the token projection gate, and draws one minus its median over the channels of the block and the FFN input neurons, averaged over layers 1–19 and 20–24, against the range for pure noise (Appendix A.4); (c) is a synthetic stream in which a marked token has to be kept across distractors, scored beyond 32 tokens; (d) reads the per-token standard deviation of the gates at update $4{,}976$ on the same fixed rows at every scale, with the plain projection input at 200M, 500M and 1.3B, the centered input at 500M, 1.3B and 2.0B, and the static gate of the model before the projection prior as reference. Figure 7(a) gives the share of the release energy on the negative side of the drive in layers 1, 4 and 24 of a 200M model with the one-sided release (Appendix A.5); (b) counts the units that never release and the readout columns whose gradient is exactly zero on paired 200M checkpoints with the one-sided and the bipolar release; (c) gives the share of PLIF channels whose mean surrogate is below 0.05 after $5{,}000$ updates of 200M models at slopes 96, 16, 8 and 4, against the share for a zero-mean Gaussian membrane, and (d) the change of their loss against slope 96 together with the change of the final loss of full runs at slope 8 on two seeds (Appendix A.6). Figure 8(a) decomposes the wide-head readout of each deep memory layer of a 200M model without the value residual (Appendix A.7); (b) and (c) read the mean absolute mixing weight and the mean output gate of the wide head in memory layers 2 to 6 of two 200M models at update $4{,}976$ with the value residual launched at zero and at 0.5, against the noise floor of a launch at zero; and (d) replaces, in a 200M model with the value residual launched at 0.5 at update $4{,}976$, the first-layer contents carried into the sharp or the wide heads of memory layers 2 to 6 by zero, by their per-channel mean or by the contents of the previous token, and gives the loss increase over all targets on 12 batches.

Result. The bias gradient is summed over one batch of $2{,}046$ tokens, on which its cancellation ratio is 0.928–0.956 for the input neurons and 0.906–0.916 for the gate neurons of the memory layers, against 0.978–0.985 for pure noise, and in the synthetic stream the static and element-wise gates recall the marked token at 0.083 and 0.079 against chance 0.078. Without centering the per-token variation of the block input gates falls with the width toward that of a static gate, from 0.060 at 200M, and at 1.3B 57.7% of these gates saturate, whereas with the centered input it rises with the width and none saturate. The negative side carries 25.5% and 11.8% of the release energy in layers 4 and 24, where the drive is skewed, and the bipolar release reduces the silent units from $1{,}413$ to 23 and the readout columns with exactly zero gradient from $4{,}297$ to 557, a smaller reduction because a silent partner still zeroes the product of the FFN. On trained checkpoints all $1{,}233$ frozen readout

columns of the FFN have a gradient of exactly 0, and 89% of them belong to units whose release rate is exactly zero; scaling the column of a silent unit by $1, 0.5, 0.25, 0.1, 0$ scales its input-side gradients by the same factors. After 5,000 updates the starved shares of 0.43, 0.21, 0 and 0 at slopes 96, 16, 8 and 4 compare with 0.50, 0.22, 0.005 and 0 for a Gaussian membrane, and the loss changes by -0.0020, -0.0063 and $+0.0002$ at slopes 16, 8 and 4 against slope 96. In memory layers 2 to 6 the constant stays below the bound in every layer; launched at zero the mixing weight stays within 1.2 to 2.3 times its noise floor, whereas launched at 0.5 it holds 0.46 to 0.48, and the output gates of the three deepest wide heads open from 0.18–0.35 to 0.34–0.73; shifting the carried contents by one token costs as much as removing them, 1.08 against 1.04 nats in the sharp heads, so the later layers use them token by token.

Constant share of the wide heads. Table 4 compares the bound of Proposition 4.6 with the constant share measured on the wide heads of a trained 200M model without the value residual, whose tagging is nearly constant ($\tau_j = 0.972$–0.978 with a token standard deviation of 0.001–0.002). The global share of its SiLU contents is $\pi_{\mathrm{g}} = 0.05$–0.14, against 0.004–0.03 for the linear contents of the Transformer. The bound on N_2 comes from the measured read profile, in which a far part of mass f and entropy H and a near part of at most 16 events give $\sum_j w_{t,j}^2 \geq f^2 e^{-H} + (1-f)^2/16$, by $H_2 \leq H$ for the far part and Cauchy–Schwarz for the near part. The bound holds in all five layers, and the measured shares follow the order of π_{g}. Solving for $1/\kappa$ gives 10–17 in four layers and 4.4 in the layer whose far mass is only 0.47, so about 180 events are worth 10 to 16 independent draws and the token deviations of the contents are correlated over about ten tokens.

Table 4: **Constant share of the wide-head readout.**

memory layer	π_{g}	bound on N_2	bound on c	measured c	$1/\kappa$	$\sqrt{1-c}$
2	0.139	182	0.78	0.62	16.2	0.62
3	0.102	100	0.70	0.54	16.5	0.68
4	0.058	189	0.54	0.31	10.5	0.83
5	0.056	43	0.44	0.18	4.4	0.91
6	0.090	187	0.45	0.37	15.8	0.80

A.9.4 Experiment IV: scale, components and evaluation

Setup. Figure 9 uses the final checkpoints of every model and scale (Tables 6 and 1); Figure 10(a) adds the matched slope, the time-constant spectrum and the centered input in turn to a 200M network with the other priors, where the step to slope 8 also sets the two implementation details of Appendix A.1 and a linear up branch of the FFN that the next step removes; (b) gives each prior of Table 7 as the seed-paired change between runs with and without it, measured on the configuration to which it was added; and (c) compares a network without the write, credit and readout priors with the final model without the centered input, which leaves the loss at 200M unchanged.

Result. Mamba-2 differs from the Transformer by -0.006 nats at 200M and -0.029 at 500M, and without the centered input the difference of NeuronSpark is $+0.010$, $+0.030$, $+0.041$ and $+0.045$ from 200M to 2.0B. On the BabyLM tasks the grammatical acceptability of NeuronSpark does not fall with scale, and the BLiMP supplement and GlobalPIQA stay within 3.3 points of the Transformer. Appendix A.10 places the remainder on new targets in the silent spiking branch of the FFN and in the order of nearby tokens.

A.9.5 Experiment V: energy per token

Setup. End-to-end energy per token is estimated from register-transfer-level descriptions of every operator of each model, each checked bit by bit against a software reference in simulation and synthesized into logic cells; an operation costs the cell count of the unit that performs it, operations are counted at a context of 1,024 tokens, and from 64 to 8,192 tokens for the 2.0B models in Figure 13(b), which is relative to the Transformer of that size at 1,024 tokens, and memory traffic is not counted for any model. Every multiplication of every model, NeuronSpark included, runs on the same 16-bit fixed-point multiply unit and every exponential or sigmoid on one lookup unit, and no operand is requantized, so NeuronSpark is credited only with work that is exactly zero. A zero release skips its multiplications in a gated unit that is bit-identical to the dense one and 7% larger, and a pool entry whose trace makes its weight exactly zero in the fixed-point datapath is skipped by its

head, while the extra queries of the last head, whose score bound is not measured, read every entry. On the final checkpoints 31 to 37% of the release-fed multiplications are active, and neuron updates cost below 0.1% of the total. Scales above 2.0B are projections of the same designs, not trained models. The Transformer takes a 2.9B shape and the Llama 2 shapes from 7B to 70B (Touvron et al., 2023b), NeuronSpark the width of the Transformer with the depth that matches its parameter count, Mamba-2 its published shapes where these exist, and NeuronSpark keeps the release densities and pool horizons of the 1.3B model.

Result. At a context of 1,024 tokens the energy per token of NeuronSpark falls from 0.73 of the Transformer's at 200M to 0.46 at 2.0B, 23 to 55% below Mamba-2 (Figure 13a). At 2.0B the Transformer's energy per token grows by 35% from 1,024 to 8,192 tokens, while NeuronSpark's grows by 15% and stays the lowest at every length (b), and the tied LM head takes 47% of NeuronSpark's energy at 200M against 29% at 2.0B (c). Mamba-2 saves attention and NeuronSpark saves projections, so at 2.0B the Transformer's energy per token exceeds that of Mamba-2, which does not depend on the context, beyond about 1,200 tokens. At 200M, where the LM head dominates, Mamba-2 falls below NeuronSpark beyond about 4,200 tokens, to 0.53 of the Transformer against 0.68 at 8,192 tokens. In projection the advantage widens to 0.36 of the Transformer's energy per token at 70B.

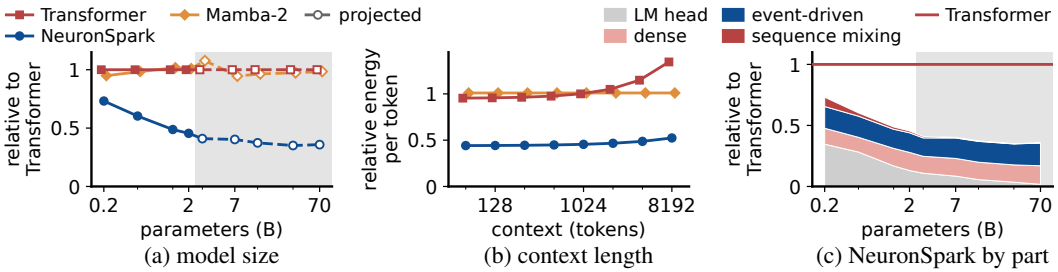


Figure 13: **End-to-end energy per token by scale and context.**

A.9.6 Controls, protocol and predictions

Shared training protocol. All models train on the BabyLM 2026 Strict corpus (Choshen et al., 2026), 100M words from child-directed speech, spoken and written British English, books, subtitles, simple encyclopedia text and telephone dialogue, tokenized with the Qwen3 tokenizer of 151,669 entries (Yang et al., 2025) and packed into rows of 1,024 tokens. One epoch of 159,232 rows is read in the same order by every model that uses the same seed, in batches of 8 rows for 19,904 updates. The optimizer is AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1 and gradient clipping at 1.0; the learning rate rises linearly to 3×10^{-4} over 1,000 updates and follows a cosine to 0.1 of the peak. Every model computes in bfloat16 and keeps FP32 master parameters and gradients, because bfloat16 storage rounds away every AdamW step that is smaller than half the spacing of the stored value, which leaves a weight above $512\,\mathrm{lr}$ in magnitude unchanged whenever its Adam ratio is at most one and would freeze most thresholds, retention biases and normalization gains. Seeds 2026, 2027 and 2028 fix the initialization and the data order, and all comparisons pair runs with the same seed. Because every row is read once, the loss of an update is measured on rows the model has not yet trained on; the final loss, the mean over every tenth update of the last 1,000 (100 updates, 800 rows), therefore estimates the loss on unseen text of the same distribution. For a parameter budget P the gap is $\Delta(P) = \mathcal{L}_{\mathrm{SNN}}(P) - \mathcal{L}_{\mathrm{TF}}(P)$, the difference of the final losses of a spiking network and a Transformer trained on the same tokens in the same order with the same optimizer and seed; a difference paired by seed has a noise floor of 0.007 nats for one seed.

Baselines. The Transformer follows Llama (Touvron et al., 2023a): pre-normalization with RMSNorm, rotary positions (Su et al., 2024), a SwiGLU MLP and tied embeddings; at 200M it has width 512, 24 layers, 8 heads and MLP width 2,624. Mamba-2 (Dao & Gu, 2024) uses heads of 64 channels with expansion 2, one group, a convolution of width 4 and state size 136; at 200M it has width 768 and 22 layers. It runs the flash-linear-attention implementation through the trainer of the Transformer, so data order, loss, optimizer, schedule and precision are those of the Transformer. Widths and depths of the baselines are set so that their parameter counts match the Transformer's

within 0.2%; NeuronSpark's width and depth are listed in Table 5, and its priors add 1–5% parameters to a backbone of the Transformer's size.

BabyLM evaluation. We evaluate the final pretrained checkpoints with the zero-shot tasks of the 2026 evaluation pipeline (Choshen et al., 2026). The zero-shot tasks compare the log-probabilities of minimal pairs or candidate continuations. BLiMP (Warstadt et al., 2020) and its supplement (Warstadt et al., 2023) test grammatical knowledge, entity tracking follows the state of objects through a description (Kim & Schuster, 2023), COMPS tests property inheritance (Misra et al., 2023), and GlobalPIQA tests physical commonsense (Chang et al., 2025). Table 6 gives the final loss and the zero-shot average of every model and scale with their seed deviations, and Table 1 in Section 5 reports every task.

Table 5: **Training protocol and model sizes.**

(a) Shared protocol

Corpus	BabyLM 2026 Strict, 100M words	Optimizer	AdamW, $\beta = (0.9, 0.95)$
Tokenizer	Qwen3 (151,669 entries)	Weight decay, clipping	0.1, 1.0
Row length	1,024 tokens	Learning rate	3×10^{-4}, 1,000 warmup, cosine
Batch, updates	8 rows, 19,904	Precision	bfloat16, FP32 master
Final loss	every 10th of last 1,000 updates	Seeds	2026, 2027, 2028

(b) Parameters (millions) and NeuronSpark geometry

Model	200M	500M	1.3B	2.0B
Transformer	199.6	501.4	1,312.0	1,984.5
NeuronSpark	209.4	514.5	1,343.1	2,012.3
Mamba-2	199.6	501.1	1,312.3	1,984.4
NeuronSpark width D	512	1,024	1,536	1,792
NeuronSpark layers L	24	17	24	28

Table 6: **Final loss and BabyLM average at four scales.**

	Final training loss				BabyLM average			
Model	200M	500M	1.3B	2.0B	200M	500M	1.3B	2.0B
Transformer	2.877±.033	2.803±.036	2.747±.035	2.751±.035	47.5	47.1	47.2	46.4
NeuronSpark	2.883±.040	2.821±.036	2.765±.035	2.769±.035	47.4	47.5	47.5	47.7
Mamba-2	**2.870**±.033	**2.774**±.036	**2.719**±.035	**2.722**±.035	**49.4**	**49.7**	**49.7**	**49.7**

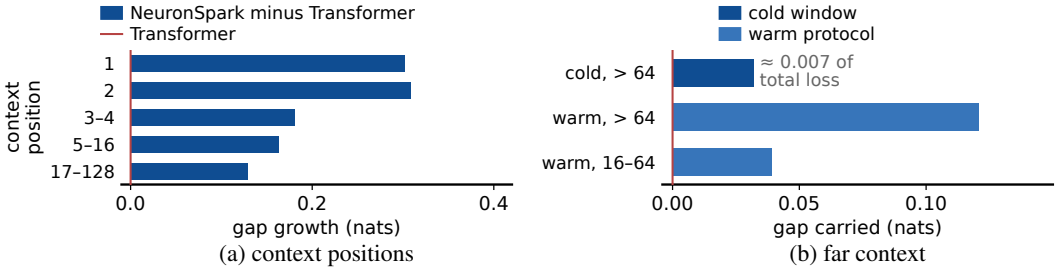
Loss in nats per token with the seed standard deviation, zero-shot accuracy in %; mean over three seeds; best of each scale in bold.

Predictions P1–P7. The decisive tests are (P1) at retention one the bipolar release matches the input rate up to $2\vartheta/T$, whereas with leak the release has a dead zone of half-width $(1-a)\vartheta$ and depends on the order of equal inputs, a past input reaches the block output only through the product of Lemma 4.2, and the closed forms of Propositions 4.3, 4.4, 4.5 and 4.6 match simulation; (P2) the difference to a Transformer vanishes when both models see only the last one or two tokens and grows with the context that a target needs; (P3) the gradient of the retention bias cancels across tokens, no static or element-wise gate keeps a token beyond a few steps, a token projection gate does, and without a centered input the constant of the projection grows with the width and saturates the gates; (P4) the one-sided release leaves units whose readout gradient is exactly zero, and the bipolar release removes them; (P5) at a steep surrogate slope a share of the PLIF channels close to the Gaussian prediction receives almost no gradient, and as the slope decreases this share vanishes while the loss first falls and then rises; (P6) the deep wide heads pass a constant share below the bound of Proposition 4.6, a mixing weight launched at zero stays at its noise floor (Lemma A.9) whereas one launched at 0.5 holds, and the value residual opens the gates of the wide heads with contents that the later layers use token by token; and (P7) with the four links open NeuronSpark lies within the Transformer's seed spread at every scale and scores at the Transformer's level on the BabyLM tasks, each prior lowers the loss, and the remaining difference lies on targets that do not occur earlier in the row. Table 2 reports each outcome.

A.10 Where the Remaining Difference Lies

The priors bring the final loss of NeuronSpark to the Transformer's, but not uniformly over targets (Figure 10c). This subsection summarizes the measurements that locate the remainder on the final 200M models. All measurements compare NeuronSpark with the Transformer on the same tokens; a single-seed paired difference below 0.007 nats is within the noise, since two seeds trained on the same tokens already differ by that much.

Target classes and positions. We call a target *copy* if the bigram that ends in it appeared earlier in the row, *seen* if the target token appeared earlier but not after the same token, and *new* otherwise (Arora et al., 2024). NeuronSpark leads on copy targets by 0.08–0.10 nats, ties on seen targets and trails on new targets by 0.13–0.17 nats on all three seeds, and the deficit on new targets grows with the position of the target in the row, from 0.03–0.08 at positions 1–16 to 0.15–0.19 at positions 257–1,024. The growth with position does not come from distant context, since keeping only the last 64 tokens changes the loss on these targets by an amount equal to about 0.007 of the total loss (Figure 14b).



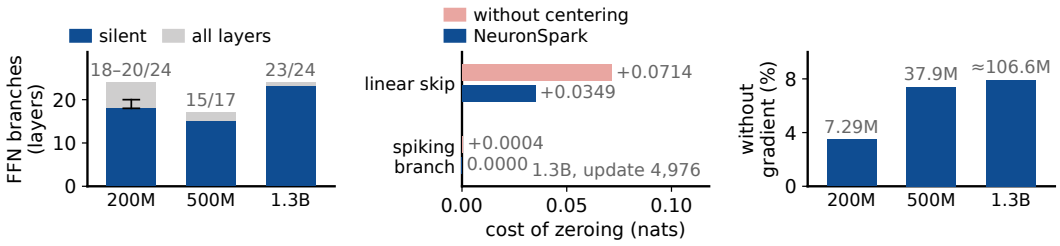
(a) context positions (b) far context

Figure 14: **Location of the remaining difference.**

Nearby word order. Keeping only the last 32 tokens and replacing one lag bucket at a time, NeuronSpark uses the current token better than the Transformer (-0.094 nats) and the tokens at lags 2, 3–4, 5–8, 9–16 and 17–32 worse ($+0.024$, $+0.061$, $+0.040$, $+0.037$ and $+0.026$). The Transformer's advantage is the order of the nearby tokens, which neither the block, whose membrane keeps about four tokens, nor the recall trace, whose kernel weighs lags without ordering them, represents.

Per-token consumption. Between update $4{,}976$ and the end of training, the deficit on frequent new targets grows most at context positions 1 and 2 ($+0.30$ nats) and less at positions 17–128 ($+0.13$), while the use of distant context by NeuronSpark catches up with the Transformer's (Figure 14a). The input neurons of the FFN are the one slowly integrating population of the model. About $4{,}400$ of their $12{,}300$ channels settle at $\eta < 0.02$, where they store a running mean of the row, and in the network without the centered input of Lemma A.5 the token projection gate places a constant $\boldsymbol{UV\mu}$ in the projection rather than in b_η, which silences these channels; centering removes this constant and at 500M returns the saturated share of the FFN input neurons from 67% to its launch level, and at 1.3B from 74% to 12%, below it. The spiking gated FFN they drive releases rarely in 18 to 20 of the 24 layers, because its gate and up neurons receive a drive far below threshold, and their product is an exact zero-gradient state of the kind analyzed in Proposition 4.4, which bipolar release removes for single units but not for products of two units. At 200M, 7.3M parameters (3.5%) receive no gradient for this reason; the other populations of the model, the six memory layers, the eighteen blocks, the input neurons of the blocks and the recall traces, carry the prediction, and replacing any of them by its running mean costs about as much as removing it. The silent populations grow with scale, to 23 silent FFN branches of 24 at 1.3B, and the parameters without gradient rise from 3.5% at 200M to 7.4% at 500M and 7.9% at 1.3B (Figure 15).

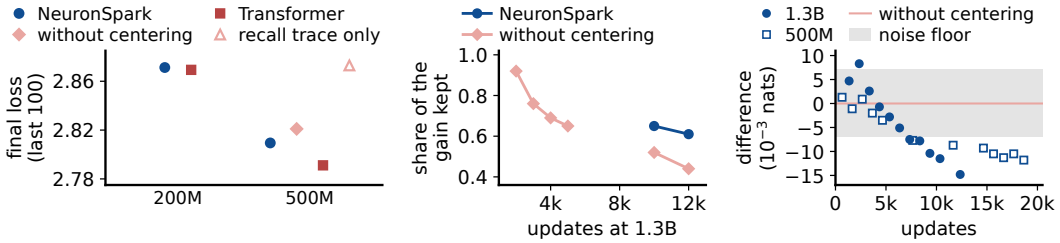
The centered projection across scales. Without the centered input, the constant $\boldsymbol{UV\mu}$ grows with the fan-in of the projection, and the share of saturated channels ($\eta < 0.02$ or $\eta > 0.98$) rises from 0.06% at 200M to 12.8% at 500M and 57.7% at 1.3B in the input neurons of the blocks and from 48% to 74% in the FFN input neurons, while the per-token variation of these gates falls toward that of a static gate (Figure 6d). The cost of the constant grows with the width. Without centering the final loss is higher by 0.0115 nats at 500M, and at 1.3B the paired difference widens by about 0.0017 nats per $1{,}000$ updates after the first $5{,}000$ (Figure 16c), so that the share of the gain over a



(a) FFN branches (b) skip burden (c) dead parameters

Figure 15: **Silent populations across scales.**

model with the recall trace alone that survives at 1.3B rises from 0.44 to 0.61 after 12,000 updates (Figure 16b).



(a) endpoints (b) gain retained (c) paired difference

Figure 16: **The centered projection at 500M and 1.3B.**

The difference that remains at 500M, +0.018 nats for seed 2026 (Figure 16a), is not in the projection gate, since with centering the spiking branch of the FFN stays silent at 1.3B, its removal costing nothing, while the load on its linear skip halves (Figure 15b). Matching the thresholds of the FFN neurons to their drive, as the time-constant spectrum does for the input neurons, targets these silent products; it is a consequence of Proposition 4.4 that NeuronSpark does not yet include.

Components kept. Table 7 lists the seed-paired loss change of every prior that NeuronSpark keeps, each measured by paired runs with and without it on the configuration to which it was added, trained for the full 19,904 updates.

Table 7: **Seed-paired loss change of each temporal prior at 200M.**

Link	Temporal prior	Loss change (nats), one entry per seed
Main priors		
Write	Token projection gate	−0.0203
Write	Time-constant spectrum, matched thresholds, no decay on 1-d	−0.0034 / −0.0031 / −0.0051
Release	Bipolar graded release	−0.1162[a]
Credit	PLIF surrogate slope 8	−0.0052 / −0.0058
Readout	Value residual launched at 0.5	−0.0210 / −0.0247
Readout	Depth anchors	−0.0047 / −0.0073
Readout	Per-token gate on the residual	−0.0039 / −0.0017 / −0.0020
Write	Centered projection input	−0.0007 / −0.0017
Secondary priors		
Write	Hebbian recall trace	−0.0055[b]
Release	Recall readout gate	0.0000
Readout	Extra patterns	−0.0043 / −0.0049
Readout	Power-law history current	−0.0038 / −0.0015
Readout	Content product term	−0.0070[c]

Seed-paired final-loss change at 200M. [a] with other implementation changes of the same run; [b] mean of three seeds; [c] never-seen targets at positions 257–1024.