A DENDRITIC NEURON WITH INTRA-BRANCH NONLINEAR MODULATION FOR SEQUENCE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dendritic neurons enrich sequence models with multiple compartment states that encode temporal memory. However, common branch readouts based on a weighted sum of compartment states do not explicitly distinguish the different functional and computational roles of compartments, restricting how internal state relationships shape the branch output. Inspired by location-dependent, asymmetric interactions within cortical pyramidal dendrites, we propose the distal-modulated proximal readout, which assigns proximal and distal roles to the compartments within each branch and uses their state differences to modulate the proximal response. Building on this readout, we construct the Nonlinear-Readout Dendritic Neuron (NR-DN), which introduces a role-specific nonlinear branch readout while retaining parallel computation of compartment states. The soma of NR-DN complements this temporal processing with a short, learnable causal window and bounded integer firing, allowing diverse frequency responses to the dendritic output. NR-DN preserves channel dimensions and can be integrated into different sequence backbones. Experiments on image, speech, and text sequence tasks demonstrate competitive performance with only one active branch per neuron. S4 equipped with NR-DN achieves 87.53% average accuracy on the Long Range Arena benchmark, demonstrating that NR-DN is an effective neuron model for sequence modeling.

1 INTRODUCTION

Vanilla point spiking neurons, such as the Leaky Integrate-and-Fire (LIF) neuron, aim to reduce the energy consumption of traditional activation neurons while retaining performance, but their effectiveness in sequence modeling is limited by oversimplified temporal dynamics (Zhang et al., 2024). Dendritic neuron models enrich sequence processing with multiple internal states that capture different temporal features. Recent multi-compartment and multi-branch dendritic designs have achieved strong performance on sequence tasks (Chen et al., 2026; Wang et al., 2025; Zhang et al., 2026), demonstrating the effectiveness of diverse compartment dynamics. However, common linear branch readouts aggregate these states without explicitly modeling the roles of and relationships among intra-branch compartments, limiting the expressive capacity of a branch as a computational unit.

State-space based neurons introduce coupling coefficients to model linear interactions between neighboring compartments and designate an output compartment, yet form their branch output through linear state aggregation (Chen et al., 2026; Wang et al., 2025; Zhang et al., 2026). Thus, distinct compartment state vectors with identical weighted sums remain indistinguishable at the branch readout, even when their inter-compartment relationships differ. A subsequent pointwise activation also acts only on the signal after readout. Other dendritic models introduce nonlinear aggregation of compartment states within a branch, but do not explicitly structure these interactions as role-specific thus compartments are still symmetric (Huang et al., 2024). Moreover, state-space neurons often introduce complex-valued computation into the compartment state update, increasing computational cost of dendrite module.

In cortical pyramidal neurons, local NMDA spikes support nonlinear integration within individual dendritic branches, which can function as computational subunits (Schiller et al., 2000; Polsky et al., 2004). This integration also varies within a branch: distal regions can exhibit higher input gain and broader temporal integration than proximal regions (Branco & Häusser, 2011). Moreover, the

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.

interactions are asymmetric: in basal branches, distal excitation lowers the local spike threshold for proximally driven responses, whereas proximal excitation can additionally increase the gain of distally driven responses (Behabadi et al., 2012). These findings inspire a proximal–distal branch readout, which assigns a response-generating role to one proximal compartment and modulates its response using the differences between distal and proximal compartment states.

For somatic processing, parallel spiking neurons learn temporal integration weights while supporting parallel computation (Fang et al., 2023), and integer-valued neurons provide multi-level outputs to mitigate quantization error (Luo et al., 2024). MMDEND combines parallel integration with scaling–shifting integer firing, but constrains its somatic kernel to a single exponential decay profile (Wang et al., 2025). Before firing, this positive-decay integrator performs low-pass smoothing and cannot implement temporal differencing.

To address the above limitations in branch readout and somatic integration, we propose the *Nonlinear-Readout Dendritic Neuron* (NR-DN). Its *Distal-Modulated Proximal Readout* (DMPR) assigns the proximal responsive role to one compartment and distal modulatory roles to the others. The differences between their states determine how the proximal response is modulated when forming the branch output. DMPR introduces nonlinear intra-branch interactions among compartments at readout while retaining parallel computation of compartment states. The soma combines a short, freely learnable causal window with bounded integer firing, enabling flexible temporal filtering of dendritic outputs. Shared branch templates provide heterogeneous dynamics across neurons in a layer, and the channel-preserving design allows NR-DN to be integrated into different sequence backbones.

Our contributions are threefold:

- We propose NR-DN, a channel-preserving dendritic neuron whose DMPR module uses distal–proximal compartment state differences to modulate the proximal response at branch readout while retaining parallel compartment updates.
- We develop a short free-kernel soma with bounded integer firing, supporting temporal responses beyond exponential low-pass smoothing. For bounded inputs, we theoretically establish bounds on membrane potentials and their sensitivity to parameter perturbations.
- Experiments on image, speech, and text sequence tasks demonstrate NR-DN's general modeling ability across backbones. With only one active branch per neuron, S4-NRDN achieves 87.53% average accuracy on LRA, outperforming S4 and other dendritic neurons.

2 Related Work

2.1 Dendritic Models

Dendritic models enrich neuronal computation through multiple compartments and branches, drawing on biological evidence for local dendritic integration (Spruston, 2008; Polsky et al., 2004). TC-LIF couples dendritic and somatic states for long-term temporal processing, while TS-LIF develops dual-compartment dynamics with distinct frequency responses for time-series forecasting (Zhang et al., 2024; Feng et al., 2025). PMSN formulates coupled hidden-compartment dynamics as temporal convolutions to enable parallel computation (Chen et al., 2026). MMDEND extends this state-space formulation to a multi-compartment, multi-branch model with adjustable input grouping (Wang et al., 2025). DRF instead equips each branch with a complex-valued resonant state, allowing branches to specialize in different frequency bands (Zhang et al., 2026).

Beyond state dynamics, these models differ in how compartment states form the branch output. PMSN rectifies a linear projection of hidden compartment states before output, whereas MMDEND applies a branch point activation function after a linear state readout (Chen et al., 2026; Wang et al., 2025). DRF linearly combines the real components of its branch states before firing (Zhang et al., 2026). DendSN introduces nonlinear intra-branch aggregation through a wavelet activation of a pooled compartment-state norm (Huang et al., 2024). However, none of these readouts explicitly models role-specific asymmetric modulation among compartments within a branch.

2.2 SOMA MODELS

Soma models have developed along two complementary directions: parallel temporal computation and richer firing representations. The PSN family removes reset-dependent recurrence, and the sliding PSN learns a finite causal window shared across time (Fang et al., 2023). SpikingSSMs instead uses a pretrained surrogate dynamic network to approximate reset-dependent LIF dynamics during parallel training (Shen et al., 2025). DRF uses a causal convolution over preliminary spikes to adapt firing thresholds, encouraging sparse firing while preserving parallel computation (Zhang et al., 2026).

Integer-valued neurons reduce quantization error through integer outputs during training and expand them into binary spike sequences at inference (Luo et al., 2024), while ternary neurons introduce signed three-level outputs (Guo et al., 2024). MMDEND combines parallel exponential integration with scaling–shifting integer firing to accommodate negative potentials and asymmetric membrane-potential distributions (Wang et al., 2025). Its temporal kernel, however, remains tied to a single exponential decay profile, which prevents independent learning of different temporal features.

3 METHOD

In this section, we describe the detailed structure of NR-DN. Sec. 3.1 provides an overview of NR-DN; Sec. 3.2 describes the branch templates, active branches, and parallel update of compartment states; Sec. 3.3 introduces DMPR, the nonlinear intra-branch modulated proximal readout; and Sec. 3.4 theoretically analyzes the advantages of our soma.

3.1 OVERVIEW

NR-DN consists of a dendrite and a soma. Given an input sequence $X \in \mathbb{R}^{T\times N\times C}$, it returns an output of the same shape, where T, N, and C denote the sequence length, batch size, and hidden channel width, respectively. Since every channel in a layer corresponds to a neuron in our model, we use *channel* and *neuron* interchangeably in the rest of the paper. The channel-preserving property of NR-DN makes it a direct replacement for any point activation neuron in a backbone model.

As illustrated in Figure 1(b), the dendrite and soma are two separate temporal processing modules with different dynamics. A total of B distinct learnable branch templates are shared across all neurons in a layer, and each neuron is assigned d of them ($1 \le d \le B$). Each branch has K compartments. The dendrite first computes the compartment states Z in parallel; DMPR then obtains the branch readouts R of all d branches, which are averaged into the somatic input I. The soma then applies a learnable causal window of length W, followed by bounded integer firing. The computation of the dendrite and soma modules follows Eq. 1:

$$\begin{gathered} X \in \mathbb{R}^{T\times N\times C} \xrightarrow{parallel} Z \in \mathbb{R}^{T\times N\times d\times C\times K} \xrightarrow{\text{DMPR}} R \in \mathbb{R}^{T\times N\times d\times C} \xrightarrow{mean} I \in \mathbb{R}^{T\times N\times C}, \\ I \in \mathbb{R}^{T\times N\times C} \xrightarrow{\text{convolution}} V \in \mathbb{R}^{T\times N\times C} \xrightarrow{\text{int spike}} S \in \mathbb{Z}^{T\times N\times C}. \end{gathered} \tag{1}$$

3.2 COMPARTMENT UPDATES IN ACTIVE BRANCHES

Pyramidal dendrites exhibit heterogeneous temporal integration, both across cell types and along individual branches (Dembrow et al., 2015; Branco & Häusser, 2011). Thin basal and tuft dendrites of pyramidal neurons also share local integration mechanisms (Larkum et al., 2009). Inspired by this heterogeneity and the shared temporal dynamics across branches, we create a shared library of B branch templates with diverse temporal processing dynamics and assign d of them to each neuron in a layer. Each branch contains K compartments, so the dendrite can be regarded as a module with d branches and K compartments per branch.

For channel $c \in \{0, \dots, C-1\}$, the active templates are selected by cyclic assignment (Fig. 2(a)):

$$r(c) = \big\{(c+l) \bmod B\big\}_{l=0}^{d-1}, \qquad 1 \le d \le B.$$

This assignment distributes channels in balance across the template library while preserving channel correspondence. At fixed d and K, increasing B provides a larger pool of learnable templates with-

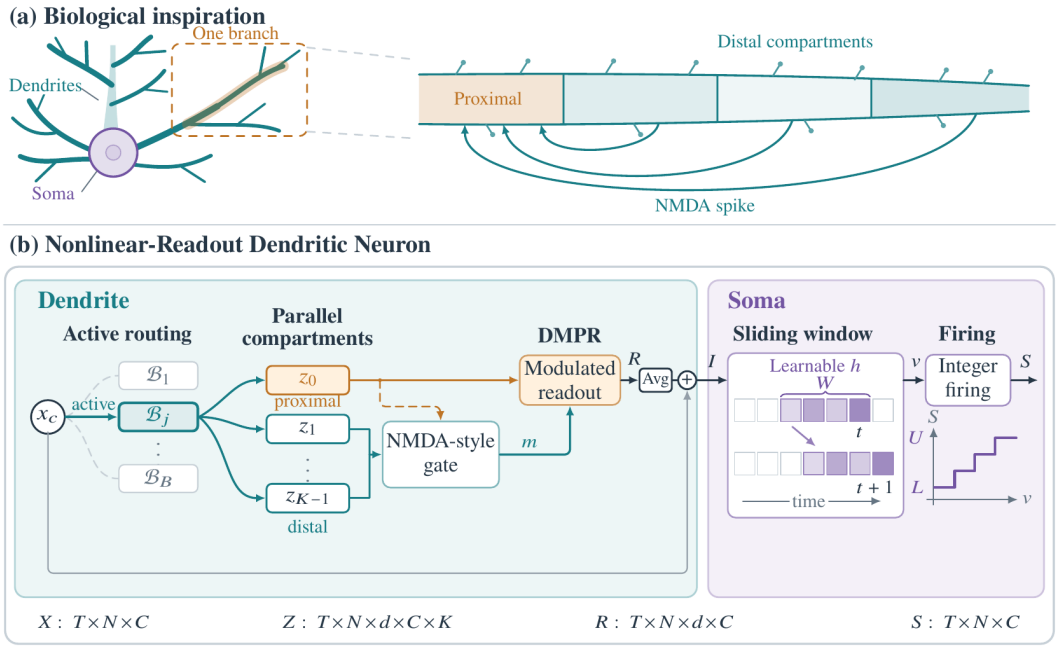


Figure 1: (a) A schematic pyramidal neuron and an enlarged dendritic branch illustrate local interactions between proximal and distal regions through NMDA spikes. (b) **Overview of NR-DN.** Input x_c is assigned d active branches from B shared templates ($d = 1$ shown). The K compartment states are computed in parallel and modulated by DMPR. The soma applies a learnable causal window of length W, followed by bounded integer firing. The output retains the same shape $T \times N \times C$.

out increasing the dK compartment states maintained per channel. Thus, the temporal processing ability is improved while the required memory remains unchanged. We find that $d = 1$ is sufficient to achieve competitive results on various datasets, so we mainly focus on this setting in the rest of the paper; see Appendix A.1 for $d > 1$. When $d = 1$, let $\mathbf{z}_{t,n,c} \in \mathbb{R}^K$ denote the compartment states driven by input $X_{t,n,c} \in \mathbb{R}$. Starting from $\mathbf{z}_{-1,n,c} = \mathbf{0}$, the compartment states are updated as in Eq. 2:

$$\begin{bmatrix} z_{t,n,c,0} \\ z_{t,n,c,1} \\ \vdots \\ z_{t,n,c,K-1} \end{bmatrix} = \overline{A}_j \begin{bmatrix} z_{t-1,n,c,0} \\ z_{t-1,n,c,1} \\ \vdots \\ z_{t-1,n,c,K-1} \end{bmatrix} + \overline{\mathbf{b}}_{j,c} X_{t,n,c}, \qquad j = r(c), \tag{2}$$

where

$$\overline{A}_j = \begin{bmatrix} 1-\tau_{j,0}^{-1} & 0 & \cdots & 0 \\ 0 & 1-\tau_{j,1}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1-\tau_{j,K-1}^{-1} \end{bmatrix} \in \mathbb{R}^{K\times K}, \quad \overline{\mathbf{b}}_{j,c} = g_{j,c} \begin{bmatrix} \tau_{j,0}^{-1} q_{j,0} \\ \tau_{j,1}^{-1} q_{j,1} \\ \vdots \\ \tau_{j,K-1}^{-1} q_{j,K-1} \end{bmatrix}. \tag{3}$$

Here, $\tau_{j,k} > 1$ is a learnable time constant that characterizes the unique dynamics of each compartment, $g_{j,c}$ denotes the channel–branch gain, and $q_{j,k}$ denotes the compartment–branch gain. These gains are constructed as in Eq. 4, which keeps $g_{j,c} \in (0.5, 1.5)$ and $\sum_k q_{j,k} = K$. Together, these parameters determine the unique temporal processing dynamics of the compartments.

$$g_{j,c} = 1 + 0.5\tanh(\widetilde{g}_{j,c}), \qquad q_{j,k} = K \frac{\mathrm{softplus}(\widetilde{q}_{j,k}) + 10^{-4}}{\sum_{r=0}^{K-1}[\mathrm{softplus}(\widetilde{q}_{j,r}) + 10^{-4}]}, \tag{4}$$

It is worth noting that Eq. 2 is a real-valued diagonal state-space model (S4D-Real), which is a subfamily of S4D (Gu et al., 2022). Replacing the original complex-valued state matrices used

in S4D and other primary dendritic neurons with real-valued ones reduces the computational cost. Unrolling Eq. 2 yields

$$\mathbf{z}_{t,n,c} = \sum_{\ell=0}^{t} \overline{A}_j^{\ell} \overline{\mathbf{b}}_{j,c} X_{t-\ell,n,c}, \tag{5}$$

so all compartment states can be computed in parallel using FFT. The decay coefficients $a_{j,k} = 1 - 1/\tau_{j,k}$ lie in $(0, 1)$, ensuring stable and non-oscillatory integration. Rather than using the original real-valued initialization of S4D, we initialize diverse short time constants geometrically spaced between 1.25 and 5.0 to obtain a more stable model. See Appendix A.1 for more details.

3.3 Distal-Modulated Proximal Readout

The compartment updates in Section 3.2 provide multiple temporal states within each branch. To form the branch output, the common linear readout in state-space based multi-compartment neurons uses a weighted sum of compartment states (Chen et al., 2026; Wang et al., 2025). With readout weights $\mathbf{w}$ and a subsequent pointwise activation f, the output depends only on this scalar projection:

$$\begin{gathered} y(\mathbf{z}) = f(\mathbf{w}^\top \mathbf{z}), \\ if\ \mathbf{w}^\top \mathbf{z}^{(1)} = \mathbf{w}^\top \mathbf{z}^{(2)}, \quad then \quad y(\mathbf{z}^{(1)}) = y(\mathbf{z}^{(2)}). \end{gathered} \tag{6}$$

Thus, distinct compartment-state vectors with the same weighted sum remain indistinguishable, which limits how intra-branch compartment relationships can shape the branch response. Furthermore, the compartment update in Eq. 2 and this weighted sum are both linear and symmetric, which constrains the expressive ability of compartments within a branch.

Within cortical pyramidal branches, local NMDA spikes support nonlinear integration (Schiller et al., 2000; Polsky et al., 2004). Interactions also depend on location: distal excitation can lower the threshold for proximally driven responses, whereas proximal excitation can additionally increase the gain of distally driven responses (Behabadi et al., 2012). Inspired by these local, asymmetric interactions (illustrated in Fig. 1(a)), we introduce DMPR as a proximal–distal assigner for branch readout. In each branch, DMPR assigns the proximal responsive role to one compartment and distal modulatory roles to all other compartments. The differences between the distal and proximal states determine how the proximal compartment is modulated. We designate $z_{t,n,c,0}$ as the proximal state and $z_{t,n,c,k}$ ($k \geq 1$) as the distal states, with differences

$$\delta_{t,n,c,k} = z_{t,n,c,k} - z_{t,n,c,0}, \qquad k = 1, \ldots, K-1. \tag{7}$$

For the assigned template $j = r(c)$, DMPR computes independently at each time step (omitting t, n, c for simplicity) following Eq. 8 to Eq. 11. An NMDA-inspired sigmoid gate p_k responds to the magnitude of each difference, with $\alpha_{j,k} > 0$ and $\theta_{j,k} \geq 0$ controlling its sensitivity and threshold, respectively. $\tanh$ retains the sign of the difference, and learnable weights $\pi_{j,k}$ combine the gated differences into a modulation signal m. This signal scales the proximal response to form R, where s_j controls the branch strength and γ_j controls the modulation strength. Positive weights satisfying $\sum_k \pi_{j,k} = 1$ ensure $|m| < 1$; when $m = 0$, the readout reduces to the modulation-free proximal response. Equation 11 combines the branch output R with the raw input X to form the somatic input I. Appendix A.2 presents the boundedness properties and the extension to $d > 1$.

$$p_k = \sigma(\alpha_{j,k}|\delta_k| - \theta_{j,k}) \tag{8}$$

$$m = \tanh\left(\sum_{k=1}^{K-1} \pi_{j,k} p_k \tanh(\delta_k)\right), \tag{9}$$

$$R = s_j z_0 \left(1 + \gamma_j m\right). \tag{10}$$

$$I_{t,n,c} = \eta R_{t,n,c} + \xi X_{t,n,c}. \tag{11}$$

This modulation also ensures that the temporal computation of our neuron goes beyond a fixed convolution. With a linear branch readout, the compartment filters and a linear causal somatic window compose into a single convolution before firing:

$$I_c = \mathcal{K}_c * X_c, \qquad v_c = b + (h * \mathcal{K}_c) * X_c. \tag{12}$$

Appendix A.3 establishes sufficient conditions under which DMPR prevents this reduction, i.e., the resulting pre-firing mapping cannot be represented by a single fixed convolution plus a bias.

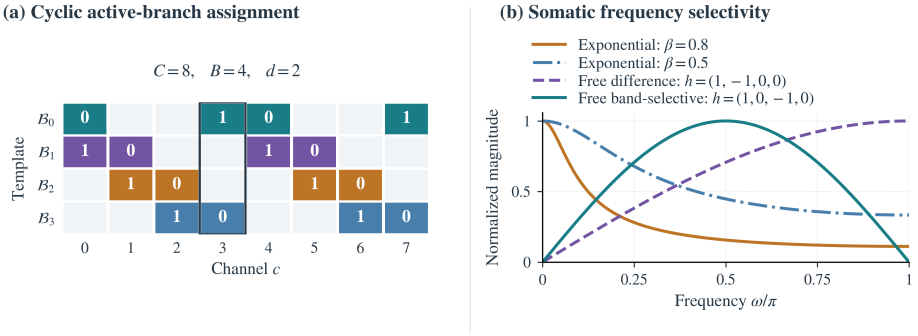


Figure 2: (a) **Cyclic routing.** For $C = 8$, $B = 4$, and $d = 2$, colored cells indicate the template indices assigned to each channel. (b) **Normalized frequency responses before firing.** The exponential kernel $h_\ell = (1-\beta)\beta^\ell$ ($\ell \geq 0$) is always low-pass, whereas the two free windows ($W = 4$) implement differencing and intermediate-frequency selection.

3.4 Free Somatic Kernel and Bounded Integer Firing

The sliding PSN enables efficient parallel temporal integration via a sliding window but suffers from quantization error in the membrane potential (Fang et al., 2023). The Integer LIF neuron provides more precise quantization but cannot be computed in parallel due to its reset mechanism (Luo et al., 2024). The existing MMDEND soma combines parallel integration with integer firing, but constrains its convolution kernel to a single exponential profile, as shown in Eq. 13 (Wang et al., 2025):

$$v_{t,n,c} = \sum_{\ell=0}^{t} \alpha\beta^l I_{t-\ell,n,c}, \qquad I_{t,n,c} = 0 \text{ for } t < 0. \tag{13}$$

Drawing inspiration from parallel and integer neurons, our soma combines a short, learnable causal sliding window with bounded integer firing. Given the dendritic output $I \in \mathbb{R}^{T\times N\times C}$, we compute the membrane potential and integer output as

$$v_{t,n,c} = b + \sum_{\ell=0}^{W-1} h_\ell I_{t-\ell,n,c}, \qquad I_{t,n,c} = 0 \text{ for } t < 0. \tag{14}$$

$$S_{t,n,c} = Q_{L,U}(v_{t,n,c}) := \lfloor \mathrm{clip}(v_{t,n,c}, L, U) \rfloor, \qquad L, U \in \mathbb{Z}, \quad L \leq 0 < U. \tag{15}$$

Here, $h \in \mathbb{R}^W$ denotes a learnable sliding window, b is a scalar bias, and L, U are the integer firing bounds. Both h and b are shared across samples and channels within a layer, resulting in $W + 1$ trainable parameters in total. The convolution in Eq. 14 can be computed in parallel using FFT. We initialize $h_\ell^{(0)} = 2^{-\ell}$ and $b^{(0)} = 0$. To overcome the non-differentiability of the floor and clip functions, we use the straight-through estimator $\widetilde{Q}'_{L,U}(v) = \mathbf{1}_{\{L\leq v\leq U\}}$ as the surrogate function (Bengio et al., 2013).

The independent sliding window broadens the temporal expressivity of the soma. For the same window length $W \geq 2$, the exponential-kernel family $\{\alpha\beta^l\}_{l=1}^W$ is a strict subset of the free-kernel family $\{h_l\}_{l=1}^W$. In the frequency domain, as shown in Fig. 2(b), the exponential kernel exhibits low-pass smoothing, whereas free kernels also support high-pass and band-selective responses, enabling temporal differencing. Details are provided in Appendix B.1.

Additionally, the membrane potential is stable. For bounded input $\|I\|_\infty \leq M$ and bounded perturbations $\Delta h, \Delta b$ of the kernel and bias,

$$\|v\|_\infty \leq |b| + M\|h\|_1, \qquad \|\Delta v\|_\infty \leq M\|\Delta h\|_1 + |\Delta b|. \tag{16}$$

Thus, the membrane potential remains bounded and depends Lipschitz-continuously on the learned kernel and bias. Our initialization satisfies $\|h^{(0)}\|_1 = 2(1 - 2^{-W}) < 2$, ensuring a controlled membrane potential over time. The detailed proof is provided in Appendix B.2. Empirically, a

kernel length as short as 4 suffices to achieve results competitive with a length-T exponential kernel, which we analyze as follows: for a soma with an exponential kernel as in Eq. 13, let $v^{[W]}$ denote the sum of its first W terms. Over the first T positions,

$$\|v - v^{[W]}\|_2^2 \leq \beta^{2W} \|v\|_2^2. \tag{17}$$

With the commonly used initialization $\beta = 1/2$, a four-step window yields a relative error of at most $1/256$ at initialization. Together with Eq. 16, this links the required window length to the decay rate rather than to the sequence length. See Appendix B.3 for more details.

At the inference stage, the soma follows the method of Luo et al. (2024) to convert integer values into binary spikes, thus reducing the total MACs introduced by integer spikes.

4 Experiments

4.1 General Sequence Modeling with NR-DN

Table 1: Comparison of neuron models on S-CIFAR and SSC.

Dataset	Method	Dendritic	Nonlinear Branch	Parameters	Accuracy
S-CIFAR10 & S-CIFAR100 (32)	LIF	✗	✗	0.51M	81.50 & 55.45
	GLIF (Yao et al., 2022)	✗	✗	0.51M	83.66 & 58.92
	KLIF (Jiang & Zhang, 2023)	✗	✗	0.51M	83.26 & 57.37
	PSN (Fang et al., 2023)	✗	✗	0.52M	88.45 & 62.21
	SPSN (Fang et al., 2023)	✗	✗	0.51M	86.70 & 62.11
	MPSN (Fang et al., 2023)	✗	✗	0.52M	85.81 & 60.69
	PMSN (Chen et al., 2026)	✓	✗	0.54M	90.97 & 66.09
	MMDEND (Wang et al., 2025)	✓	✗	0.51M	92.71 & 67.65
	NR-DN(Ours)	✓	✓	0.52M	**93.86** & **70.71**
SSC	SRNN (Cramer et al., 2020)	✗	✗	0.11M	50.9
	ALIF (Yin et al., 2021)	✗	✗	0.73M	74.20
	PSN (Fang et al., 2023)	✗	✗	0.32M	43.71
	SPSN (Fang et al., 2023)	✗	✗	0.13M	71.50
	MPSN (Fang et al., 2023)	✗	✗	0.32M	68.04
	TC-LIF (Zhang et al., 2024)	✓	✗	0.11M	63.46
	MMDEND (Wang et al., 2025)	✓	✗	0.13M	75.63
	NR-DN(Ours)	✓	✓	0.13M	**76.12**
S-CIFAR10 (1024)	PMSN (Chen et al., 2026)	✓	✗	0.21M	82.14
	DRF (Zhang et al., 2026)	✓	✗	0.21M	**84.30**
	NR-DN(Ours)	✓	✓	0.26M	83.26

To demonstrate the sequence modeling ability of our neuron, we compare NR-DN with point neurons and other dendritic neurons on four sequence tasks: 32-step column-by-column sequential CIFAR10/100 (Fang et al., 2023), pixel-by-pixel sequential CIFAR10 (Chen et al., 2026), and the speech dataset SSC (Cramer et al., 2020). The backbone models for these four datasets are relatively simple, so the performance depends heavily on the neuron model.

As shown in Tab. 1, NR-DN achieves strong performance among existing neuron models with fewer or comparable parameters. It consistently outperforms point neurons, including LIF variants and parallel neurons, showing the impact of NR-DN's dendrite module. Moreover, it surpasses the MMDEND and PMSN dendritic neurons on all datasets with the same backbone model, including improvements of 1.1% and 3% over MMDEND on S-CIFAR10 and S-CIFAR100, respectively. These results demonstrate its strong general sequence modeling ability. Notably, among these neurons, NR-DN is the only dendritic model that introduces nonlinear interactions within a branch.

4.2 LONG-RANGE SEQUENCE MODELING

To further test the temporal processing capability of NR-DN, we integrate it into S4 (Gu et al., 2021), a more complex state-space model that has proven effective on sequence tasks. We evaluate our model on LRA (Tay et al., 2020), which contains six image and text sequence tasks ranging from 1K to 16K steps; the average accuracy across these six tasks serves as the final score. As shown in Tab. 2, our model achieves the best performance on every task. It not only surpasses the S4 baseline by 1.5%, but also outperforms the dendritic model MMDEND and SpikingSSM, which use S4 as the backbone, by 1.4% and 3.2%, respectively. Although the comparison with neurons such as DRF is not entirely fair due to different backbones, their parameter counts are close, so it still demonstrates the effectiveness of NR-DN (see Appendix C for details). These findings demonstrate the value of integrating NR-DN in strong baseline models for further improving their temporal processing ability.

Table 2: Performance comparison of models on LRA.

Method	Spiking	ListOps	IMDB	AAN	CIFAR	Pathfinder	Path-X	Avg.
Transformer(Vaswani et al., 2017)	✗	36.37	64.27	57.46	42.44	71.40	/	53.66
S4(Gu et al., 2021)	✗	59.60	86.82	90.90	88.65	94.20	96.35	86.09
SpikingLMU (Liu et al., 2024)	✓	37.30	65.80	79.76	55.65	72.68	/	60.20
SD-TCM (Huang et al., 2026)	✓	59.20	86.33	89.88	84.77	91.76	/	76.99
DRF (Zhang et al., 2026)	✓	60.02	86.52	90.02	85.32	92.36	82.80	82.86
SpikingSSM (Shen et al., 2025)	✓	60.23	80.41	88.77	88.21	93.51	94.82	84.33
S4-SPSN(Fang et al., 2023)	✓	56.80	85.70	86.40	85.60	92.60	50.30	76.20
S4-MMDEND (Wang et al., 2025)	✓	59.90	88.60	90.00	87.80	94.30	96.30	86.10
S4-NRDN(Ours)	✓	**62.00**	**89.32**	**91.66**	**89.08**	**95.70**	**97.42**	**87.53**

4.3 NEURON COMPUTATIONAL EFFICIENCY ANALYSIS

In this section, we compare the computational cost of NR-DN with that of an existing parallel point neuron (Fang et al., 2023) and two primary multi-compartment dendritic neurons (Wang et al., 2025; Zhang et al., 2026). The computational efficiency of NR-DN benefits from the real-valued computing in compartment update and the short kernel length in the soma. As shown in Tab. 3, we compute the FLOPS of all three components of each neuron. Following their configurations on LRA tasks, the total FLOPS of SPSN, MMDEND, and DRF are $2TCW_s$, $66TC$, and $81TC$, respectively. Our NR-DN requires only around $57TC$ FLOPS, which is higher than SPSN with the same kernel length but about 15% and 31% lower than MMDEND and DRF, respectively.

Table 3: Neuron FLOPs per sample. Every entry is to be multiplied by TC.

Neuron	Compartment update	Branch readout	Soma
SPSN	0	0	$2W_{\mathrm{S}}$
MMDEND	$10\bar{M}K_{\mathbb{C}}$	$\bar{M}(4K_{\mathbb{C}}+1)$	$2\bar{M}+7$
DRF	$8R$	$2R-1$	$2W_{\mathrm{D}}+2$
NR-DN(Ours)	$3dK$	$d(9K-3)+3$	$2W$

K and $K_{\mathbb{C}}$ denote the numbers of real compartments and complex modes, respectively; d, $\bar{M}$, and R denote the active, average, and DRF branch counts per channel, respectively. W_{S}, W, and W_{D} are the window lengths of SPSN, NR-DN, and DRF, respectively.

4.4 ABLATION STUDIES

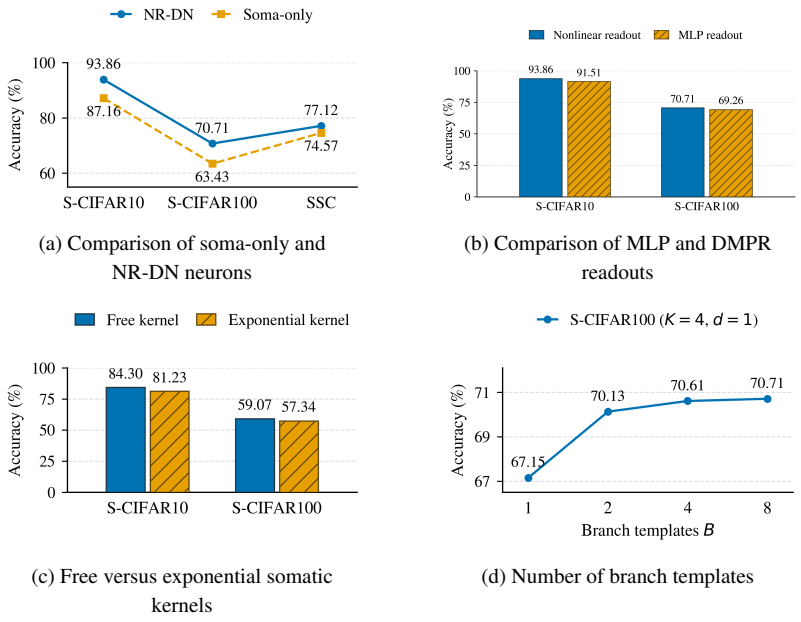
Soma-only neuron. To demonstrate that the dendrite is a necessary module contributing to NR-DN's strong performance, we conduct experiments on S-CIFAR10, S-CIFAR100, and SSC using only the soma. To reach the full potential of the soma, we set its kernel length to the number of

time steps T of each dataset. As shown in Fig. 3(a), NR-DN consistently outperforms the soma-only neuron on all three datasets, by over 5% on average. This confirms the essential temporal processing ability of dendrite module.

Branch readout method. To demonstrate the effectiveness of DMPR, we replace it with a one-layer MLP with sigmoid activation as the branch readout and evaluate it on S-CIFAR10 and S-CIFAR100. For a fair parameter comparison, we set the hidden dimension of the MLP to 2. As shown in Fig. 3(b), the neuron with DMPR outperforms the MLP readout by over 2% on both datasets, demonstrating the effectiveness of DMPR as an role-specific assigning module.

Exponential kernel. We replace the independent kernel of our soma with the exponential kernel in Eq. 13 and compare the resulting exponential-kernel soma with our free-kernel soma, whose kernel length is set to 4. As shown in Fig. 3(c), the kernel replacement causes accuracy drops of 3% and 2% on S-CIFAR10 and S-CIFAR100, respectively. This provides evidence for our hypothesis in Sec. 3.4 that a short free kernel has better expressive ability than a long exponential kernel.

Number of branch templates. We test the impact of the number of branch templates B, which accounts for the total temporal branch dynamics of neurons in a layer. As shown in Fig. 3(d), with the compartment number K and active branch number d fixed, the accuracy of NR-DN on S-CIFAR100 increases monotonically as B grows from 1 to 8, while the memory consumption remains unchanged. The 3% gap between $B = 1$ and $B = 2$ shows the importance of dynamic templates.



(a) Comparison of soma-only and NR-DN neurons

(b) Comparison of MLP and DMPR readouts

(c) Free versus exponential somatic kernels

(d) Number of branch templates

Figure 3: Ablation studies. (a) NR-DN versus soma-only neurons on S-CIFAR10, S-CIFAR100, and SSC. (b) MLP versus DMPR branch readouts. (c) Free versus exponential somatic kernels. (d) Effect of the number of branch templates B on S-CIFAR100.

5 Conclusion

We presented NR-DN, a channel-preserving dendritic neuron that combines a role-specific nonlinear branch readout with flexible somatic processing. DMPR uses distal–proximal compartment state differences to modulate the proximal response when forming the branch output, while retaining parallel computation of compartment states. The soma pairs a short, freely learnable causal window with bounded integer firing, providing flexible temporal responses and multi-level output encoding. Experiments on image, speech, and text sequence tasks show competitive performance across backbones with only one active branch per neuron. These findings support NR-DN as a transferable dendritic neuron model with effective sequence modeling ability.

AI USE STATEMENT

We mainly used generative AI tools to assist with mathematical conclusions and proofs, discussions of experimental design, and the interpretation of results. We did not use generative AI tools for our novel ideas, method implementation, or experimental results. Additional uses included literature search, outline drafting, and figure preparation. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops a bio-inspired computational dendritic neuron model and evaluates it on existing, publicly available image, speech, and text benchmarks. Its neuroscience inspiration is a functional abstraction rather than a claim of physiological fidelity. Responsible downstream use requires task-specific assessment of dataset bias, privacy, and potential misuse, together with respect for dataset licenses and usage restrictions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the complete NR-DN architecture, including the state updates, branch readout, dendritic output, and somatic integration and firing. Appendices A and B provide parameterization and initialization details, as well as the theoretical proofs and analyses supporting the key statements in the paper. Appendix C documents the backbone models and the neuron configuration for each task, and Table 4 summarizes the training hyperparameters.

REFERENCES

Bardia F Behabadi, Alon Polsky, Monika Jadi, Jackie Schiller, and Bartlett W Mel. Location-dependent excitatory synaptic interactions in pyramidal neuron dendrites. *PLoS computational biology*, 8(7):e1002599, 2012.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

Tiago Branco and Michael Häusser. Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron*, 69(5):885–892, 2011.

Xinyi Chen, Jibin Wu, Chenxiang Ma, Yinsong Yan, Hanwen Liu, Yujie Wu, and Kay Chen Tan. Pmsn: A parallel multi-compartment spiking neuron for multiscale temporal processing. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.

Nikolai C Dembrow, Boris V Zemelman, and Daniel Johnston. Temporal dynamics of l5 dendrites in medial prefrontal cortex regulate integration versus coincidence detection of afferent inputs. *The Journal of Neuroscience*, 35(11):4501–4514, 2015.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. *Advances in Neural Information Processing Systems*, 36:53674–53687, 2023.

Shibo Feng, Wanjin Feng, Xingyu Gao, Peilin Zhao, and Zhiqi Shen. Ts-lif: a temporal segment spiking neuron network for time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 28064–28084, 2025.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in neural information processing systems*, 35:35971–35983, 2022.

Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12244–12252, 2024.

Yifan Huang, Wei Fang, Zhengyu Ma, Guoqi Li, and Yonghong Tian. Scalable dendritic modeling advances expressive and robust deep spiking neural networks. *arXiv preprint arXiv:2412.06355*, 2024.

Yulong Huang, Zunchang Liu, Xiaopeng Lin, Hongwei Ren, Hongxiang Huang, Haotian Fu, Yue Zhou, Hong Xing, Zeke Xie, and Bojun Cheng. Prf: Parallel resonate and fire neuron for long sequence learning in spiking neural networks. *Pattern Recognition*, pp. 114968, 2026.

Chunming Jiang and Yilei Zhang. Klif: An optimized spiking neuron unit for tuning surrogate gradient slope and membrane potential. *arXiv preprint arXiv:2302.09238*, 2023.

Matthew E Larkum, Thomas Nevian, Maya Sandler, Alon Polsky, and Jackie Schiller. Synaptic integration in tuft dendrites of layer 5 pyramidal neurons: a new unifying principle. *Science*, 325 (5941):756–760, 2009.

Zeyu Liu, Gourav Datta, Anni Li, and Peter Beerel. Lmuformer: Low complexity yet powerful spiking model with legendre memory units. In *International conference on learning representations*, volume 2024, pp. 12108–12122, 2024.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals and Systems*. Prentice Hall, 2 edition, 1997.

Alon Polsky, Bartlett W Mel, and Jackie Schiller. Computational subunits in thin dendrites of pyramidal cells. *Nature neuroscience*, 7(6):621–627, 2004.

Jackie Schiller, Guy Major, Helmut J Koester, and Yitzhak Schiller. Nmda spikes in basal dendrites of cortical pyramidal neurons. *Nature*, 404(6775):285–289, 2000.

Shuaijie Shen, Chao Wang, Renzhuo Huang, Yan Zhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, and Luziwei Leng. Spikingssms: Learning long sequences with sparse and parallel spiking state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20380–20388, 2025.

Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206–221, 2008.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kexin Wang, Yuhong Chou, Di Shang, Shijie Mei, Jiahong Zhang, Yanbin Huang, Man Yao, Bo Xu, and Guoqi Li. Mmdend: Dendrite-inspired multi-branch multi-compartment parallel spiking neuron for sequence modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27459–27470, 2025.

Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. Glif: A unified gated leaky integrate-and-fire neuron for spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 32160–32171, 2022.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10): 905–913, 2021.

Dehao Zhang, Malu Zhang, Shuai Wang, Jingya Wang, Wenjie Wei, Zeyu Ma, Guoqing Wang, Yang Yang, and Haizhou Li. Dendritic resonate-and-fire neuron for effective and efficient long sequence modeling. *Advances in Neural Information Processing Systems*, 38:107568–107594, 2026.

Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Tc-lif: A two-compartment spiking neuron model for long-term sequential modelling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16838–16847, 2024.

A Analysis of the Dendrite

A.1 Real-Valued Diagonal State Dynamics, Initialization, and Stability

We assume a zero initial state and zero resting voltage. Our model clamps each learned time constant to $\tau_{j,k} \geq 1.001$ to avoid division by zero. The input gains follow Eq. (4).

Eq. (2) has an exact continuous-time representation. Denote $a_{j,k} = 1 - \frac{1}{\tau_{j,k}}$ and

$$\Lambda_j = \operatorname{diag}(\log a_{j,0}, \ldots, \log a_{j,K-1}), \qquad \mathbf{b}^{\mathrm{ct}}_{j,c} = -g_{j,c}\Lambda_j \mathbf{q}_j. \tag{18}$$

Without loss of generality, let the time interval be $\delta = 1$. Following S4D (Gu et al., 2022), applying ZOH discretization to the state model $\dot{\mathbf{z}} = \Lambda_j \mathbf{z} + \mathbf{b}^{\mathrm{ct}}_{j,c} x$ gives

$$e^{\Lambda_j} = \overline{A}_j, \qquad \Lambda_j^{-1}(e^{\Lambda_j} - I_K)\mathbf{b}^{\mathrm{ct}}_{j,c} = g_{j,c}(I_K - \overline{A}_j)\mathbf{q}_j, \tag{19}$$

which exactly matches Eq. 3. Thus, the parallel update of our compartments is a subset of S4D. We name it S4D-Real since $\overline{A}_j$ and $\overline{b}_{j,c}$ are real-valued. For K stored states, our real-valued dynamics require only K real state coordinates instead of the $2K$ coordinates of K complex modes.

S4D provides both a real-valued initialization $\Lambda_{kk} = -(k+1)$ and a complex-valued initialization $\Lambda_{kk} = -\frac{1}{2} + i\pi k$ (Gu et al., 2022). Our NR-DN instead uses a short-timescale initialization: the time constants $\tau_{j,k}$ are geometrically spaced from 1.25 to 5, so the resulting $a_{j,k}$ lie in $[0.2, 0.92]$. Their e-folding times, $\frac{-1}{log a_{j,k}}$, range from approximately 1 to 12, indicating diverse initial memory of recent inputs. Our experimental results show that this relatively small initial memory window is effective.

Expanding Eq. (2) yields

$$z_{t,n,c,k} = g_{j,c} q_{j,k}(1 - a_{j,k}) \sum_{\ell=0}^{t} a_{j,k}^{\ell} X_{t-\ell,n,c}. \tag{20}$$

For $|X_{t,n,c}| \leq M$, the geometric sum gives

$$|z_{t,n,c,k}| \leq |g_{j,c}| q_{j,k} M (1 - a_{j,k}^{t+1}) \leq |g_{j,c}| q_{j,k} M. \tag{21}$$

Hence, the compartment voltage is bounded independently of the sequence length.

For $d > 1$, active branch $e \in \{0, \ldots, d-1\}$ uses template $r(e,c) = (c+e) \bmod B$. Replacing $j = r(c)$ with $j = r(e,c)$ in Eq. (2) gives independent states $\mathbf{z}_{t,n,e,c} \in \mathbb{R}^K$, and all the conclusions above still hold.

A.2 Readout Parameterization and Bounded Modulation

Our default initialization is $\widetilde{\alpha}_{j,k} = 6$, $\widetilde{\theta}_{j,k} = 0.5$, $\widetilde{\pi}_{j,k} = 0$, $\gamma_j = 0.1$, $s_j = \eta = 1$, and $\xi = 0$. Thus, the initial mixing weights are $\pi_{j,k} = 1/(K-1)$. For finite states and parameters, each gated difference satisfies $|p_k \tanh(\delta_k)| < 1$. Since the mixing weights form a convex combination, $|m| < \tanh(1) < 1$. For bounded input $|X_{t,n,c}| \leq M$, Eq. (21) therefore yields

$$\begin{aligned} |R_{t,n,c}| &\leq |s_j| \left[(1 + |\gamma_j| \tanh(1))|z_{t,n,c,0}|\right] \\ &\leq |s_j| \left[(1 + |\gamma_j| \tanh(1))|g_{j,c}| q_{j,0} M\right]. \end{aligned} \tag{22}$$

Thus, the branch output remains bounded independently of the sequence length. If all distal states equal the proximal state, then $\delta_k = 0$ and $m = 0$, so R degenerates to $s_j z_0$.

For general d, the same readout is applied to each active branch slot e using template $j = r(e, c)$. The somatic input is then the weighted sum

$$I_{t,n,c} = \frac{\eta}{\nu_d} \sum_{e=0}^{d-1} R_{t,n,e,c} + \xi X_{t,n,c}. \tag{23}$$

The implementation supports mean, square-root, and sum aggregation, with $\nu_d = d$, $\sqrt{d}$, and 1, respectively; all three coincide when $d = 1$.

A.3 Linear Composition and Nonlinear Compartment Interactions

We assume fixed parameters, fixed routing, zero initial states, and zero resting potential.

Proposition 1 (Composition under linear readout)**.** *Suppose each branch uses an S4-style compartment update and the branch readout is linear. Then the combined dendritic and somatic computation is equivalent to a single channel-wise convolution plus the somatic bias.*

Proof. Omitting batch and branch indices, each compartment satisfies $z_{c,k} = \kappa_{c,k} * X_c$ by Eq. (5). A linear branch readout has the form

$$R_c = \sum_k w_{c,k} z_{c,k} + D_c X_c = \left(\sum_k w_{c,k} \kappa_{c,k} + D_c \delta_0 \right) * X_c,$$

where δ_0 is the unit impulse. A fixed branch aggregation and a linear input residual can be absorbed into an effective dendritic kernel $\mathcal{K}_c$. The associativity of causal convolution then gives $v_c = b + h * (\mathcal{K}_c * X_c) = b + (h * \mathcal{K}_c) * X_c$. □

Proposition 2 (DMPR prevents degeneration to a linear convolution)**.** *Consider $d = 1$ and a somatic window with $h_0 \neq 0$. If there exists an input sequence X for which $\eta s_{r(c)} \gamma_{r(c)} z_{\cdot,n,c,0}[X] m_{\cdot,n,c}[X]$ is not identically zero for some n, c, then the resulting combination of dendrite and soma cannot be represented by a linear convolution plus a bias.*

Proof. Simultaneously negate all input channels. The linearity of the zero-state dynamics gives $z[-X] = -z[X]$ and $\delta[-X] = -\delta[X]$. Each magnitude gate is even under this transformation, the mixing weights are input-independent, and both hyperbolic tangents in Eq. (9) are odd. Thus, $m[-X] = -m[X]$, and

$$R[X] + R[-X] = 2 s_j \gamma_j z_0[X] m[X]. \tag{24}$$

The unmodulated proximal contribution, the additive distal term $s_j \rho_j m$, and the raw-input residual cancel under this sign reversal. Writing $f = v - b$, the somatic voltage consequently satisfies

$$f_c[X] + f_c[-X] = 2\eta s_j \gamma_j \, h * \big(z_{c,0}[X] m_c[X] \big). \tag{25}$$

On a finite sequence, causal convolution by h corresponds to a lower-triangular matrix with diagonal entries h_0, which is invertible when $h_0 \neq 0$; hence, the right-hand side is nonzero under the stated condition. In contrast, every fixed linear convolution is odd and must satisfy $f[X] + f[-X] = 0$. Since $v[0] = b$, choosing a different constant bias cannot remove this contradiction. □

This property holds with the model's positive real decay modes. For example, consider one channel with $K = 2$, unit input gains, $\tau_0 = 2$, $\tau_1 = 4$, and a unit impulse input. At the first step, $z_0 = 1/2$ and $z_1 = 1/4$, so $\delta_1 = -1/4$ and $\widehat{\delta}_1 = (-1/4)/\sqrt{1/16 + \varepsilon}$. Since $\pi_1 = 1$, $m = \tanh[p_1 \tanh(\widehat{\delta}_1)] \neq 0$. Taking $s = \eta = 1$ and the initial value $\gamma = 0.1$ gives $\gamma z_0 m \neq 0$, and the initialized somatic window already satisfies $h_0 = 1$. Therefore, DMPR enables input-dependent interactions among compartment responses that a fixed composite convolution cannot express, without introducing a sequential nonlinear compartment state update.

B Analysis of the Soma

We analyze the somatic integrator for a fixed incoming feature sequence, under identical parameter-sharing assumptions when comparing kernel families. Sample and channel indices are omitted. Inputs are zero for negative time indices, and norms are taken over the evaluated sequence.

B.1 Temporal Expressivity and Frequency Response

For a fixed window length W, Eq. 13 and Eq. 14 define the kernel families

$$\mathcal{E}_W = \{a(1, \beta, \ldots, \beta^{W-1}) : a \in \mathbb{R},\ 0 < \beta < 1\}, \qquad \mathcal{F}_W = \mathbb{R}^W. \tag{26}$$

Every element of $\mathcal{E}_W$ belongs to $\mathcal{F}_W$, whereas for $W \geq 2$, the kernel $(1, -1, 0, \ldots) \in \mathcal{F}_W$ has $h_1/h_0 = -1$ and thus does not belong to $\mathcal{E}_W$. Hence, $\mathcal{E}_W \subsetneq \mathcal{F}_W$.

The exponential kernel has the frequency response

$$H_{\mathrm{E}}(\omega) = \frac{\alpha}{1 - \beta e^{-i\omega}}, \qquad |H_{\mathrm{E}}(\omega)|^2 = \frac{\alpha^2}{1 + \beta^2 - 2\beta \cos \omega}. \tag{27}$$

Its magnitude decreases from $\omega = 0$ to π, resulting in a low-pass response. In contrast, our free kernel has $H_W(\omega) = \sum_{\ell=0}^{W-1} h_\ell e^{-i\omega\ell}$. For $W = 4$, $h = (1, -1, 0, 0)$ yields $|H_W(\omega)| = 2|\sin(\omega/2)|$, which removes the constant component, while $h = (1, 0, -1, 0)$ gives $|H_W(\omega)| = 2|\sin \omega|$, which suppresses both the zero and Nyquist frequencies. These responses demonstrate temporal operations beyond exponential smoothing.

B.2 Bounded and Continuous Membrane Potential

Proof of Eq. (16):

Proof. Let $v = b + h * I$ and $\|I\|_\infty \leq M$. Applying the triangle inequality gives the standard finite-convolution bound (Oppenheim et al., 1997)

$$|v_t| \leq |b| + \sum_{\ell=0}^{W-1} |h_\ell|\, |I_{t-\ell}| \leq |b| + M\|h\|_1. \tag{28}$$

For the same input and perturbed parameters, $\Delta v = \Delta b + \Delta h * I$; hence,

$$\|\Delta v\|_\infty \leq |\Delta b| + M\|\Delta h\|_1. \tag{29}$$

□

More generally, suppose $\|\Delta I\|_\infty \leq \epsilon_I$, $\|\Delta h\|_1 \leq \epsilon_h$, and $|\Delta b| \leq \epsilon_b$. Expanding the perturbed convolution gives

$$\begin{aligned} \Delta v &= \Delta b + h * \Delta I + \Delta h * (I + \Delta I), \\ \|\Delta v\|_\infty &\leq \epsilon_b + \|h\|_1 \epsilon_I + (M + \epsilon_I)\epsilon_h. \end{aligned} \tag{30}$$

At initialization, $\|h^{(0)}\|_1 = \sum_{\ell=0}^{W-1} 2^{-\ell} = 2(1 - 2^{-W}) < 2$ and $b^{(0)} = 0$. Thus, $\|v^{(0)}\| < 2M$, and the gain of Δv with respect to input perturbations is below 2 regardless of the window length. Together, these results demonstrate the stability of the somatic membrane potential.

B.3 Effective Memory and Short-Window Approximation

Proof of Eq. (17):

Proof. According to Eq. 13, for $1 \leq W \leq T$,

$$v_t = \alpha \sum_{\ell=0}^{t} \beta^\ell I_{t-\ell}, \qquad v_t^{[W]} = \alpha \sum_{\ell=0}^{\min(t, W-1)} \beta^\ell I_{t-\ell}. \tag{31}$$

The error is zero for $t < W$. For $t \geq W$,

$$\begin{aligned} v_t - v_t^{[W]} &= \alpha \sum_{\ell=W}^{t} \beta^{\ell} I_{t-\ell} \\ &= \beta^W \alpha \sum_{r=0}^{t-W} \beta^r I_{t-W-r} = \beta^W y_{t-W}. \end{aligned} \tag{32}$$

Summing over the first T positions yields

$$\begin{aligned} \sum_{t=0}^{T-1} |v_t - v_t^{[W]}|^2 &= \beta^{2W} \sum_{t=0}^{T-W-1} |v_t|^2 \\ &\leq \beta^{2W} \sum_{t=0}^{T-1} |v_t|^2. \end{aligned} \tag{33}$$

□

For a target relative squared error $\varepsilon \in (0, 1)$, a sufficient window length is

$$W \geq \left\lceil \frac{\log \varepsilon}{2 \log \beta} \right\rceil. \tag{34}$$

With the initialization $\beta = 1/2$, $W = 4$ yields an error of $\beta^{2W} = 1/256 \approx 0.39\%$. The corresponding bounds at $\beta = 0.9$ and 0.99 are 43.05% and 92.27%, respectively, illustrating the dependence on the decay rate. Note that β is learnable, so this error bound does not necessarily hold during training. Nevertheless, this analysis inspired us to use an independent kernel of length 4 to match or even surpass the exponential kernel.

C Experimental Setup

All experiments are conducted on NVIDIA GeForce RTX 5090 GPUs. The results on each LRA dataset are averaged over three random seeds. AAN and PATHX are trained on 4 and 8 GPUs in parallel, respectively, while all other datasets are trained on a single GPU. Unless otherwise stated, NR-DN uses 8 branch templates, 4 compartments, and 1 active branch per neuron in the dendrite, and a sliding window of length 4 in the soma. The firing range bounds L and U are set per task and specified below. Detailed training hyperparameters are listed in Tab. 4.

Long Range Arena. For the LRA tasks (Tay et al., 2020), we use bidirectional S4 (Gu et al., 2021) as the backbone model, with NR-DN replacing the GELU activation function after the FFT convolution. As described in the S4 paper, we use 8 blocks with a hidden dimension of 128 for Listops and 6 blocks for the other tasks. We use $L = 0, U = 4$ for PATHX and $L = -4, U = 4$ for the other tasks.

Column-by-Column sequential CIFAR-10/100. We follow the backbone model used by Fang et al. (2023), which has two stages, each containing three "Conv1d–BatchNorm–NRDN" blocks followed by one average pooling layer. We use $L = -4, U = 4$ for both datasets.

Pixel-by-Pixel sequential CIFAR-10. We follow the backbone model used by Chen et al. (2026), which has 6 residual blocks: (FC128-BN-NRDN)-RB128-RB128-AP4-(FC256-BN-NRDN)-RB256-RB256. We use $L = 0, U = 1$ for this dataset.

Spiking Speech Commands. We follow the backbone used in MMDEND (Wang et al., 2025), which has three hidden layers with a hidden dimension of 128. We use $L = -4, U = 4$ for this dataset.

Table 4: Training hyperparameters of each dataset.

Dataset	lr	neuron lr	wd	batch	Epochs
ListOps	0.01	0.001	0.05	32	40
IMDB	0.01	0.001	0.05	16	32
AAN	0.01	0.001	0.05	64	20
CIFAR	0.01	0.001	0.05	50	200
Pathfinder	0.004	0.001	0.05	64	200
PathX	0.001	0.001	0.05	16	50
S-CIFAR10 (32)	0.001	0.001	0	128	400
S-CIFAR100 (32)	0.001	0.001	0	128	400
S-CIFAR10 (1024)	0.01	0.01	0.01	64	200
SSC	0.005	0.005	5e-4	32	200

Model size on LRA. Table 5 compares the total parameter counts of the models on the six LRA tasks. The S4 and DRF counts are taken from Table 6 of Zhang et al. (2026). On ListOps, we use an S4 backbone with 8 blocks and a hidden dimension of 128 rather than the default 6 blocks with a hidden dimension of 256 in codebase, which substantially reduces the number of parameters while maintaining accuracy.

Table 5: Model parameter counts on LRA.

Method	ListOps	Text	Retrieval	Image	Pathfinder	PATHX
S4	815K	843K	1.1M	3.6M	1.3M	–
DRF	297K	841K	1.1M	3.2M	1.3M	–
S4-NRDN (Ours)	290K	856K	1.1M	3.6M	1.3M	1.3M

"–" indicates that the parameter count is not reported in the cited table.