MULTI-RATE DENDRITIC MEMORY SPIKING NEURAL NETWORKS FOR MULTI-TIMESCALE NEURAL SIGNAL DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Invasive neural recordings capture signals with distinct timescales, including action potentials, local field potentials, and dopamine concentrations. Decoding behavior from these multimodal streams requires combining signals with different timescales, sampling rates, and temporal information densities while preserving modality-specific discriminative information. Existing methods often struggle to explicitly and independently control the temporal resolution at which history is updated and the span of history that is retained, making this integration difficult. Inspired by the proximal-to-distal functional gradient of biological dendrites, we introduce MRD-SNN, a spiking network in which each neuron maintains a direct somatic pathway and multiple sparse dendritic memory branches. Each branch refreshes at its own rate and uses a Legendre representation to summarize history over a specified nominal physical horizon. We evaluate the model on three cross-modal neural decoding tasks. MRD-SNN achieves the highest accuracy on all three tasks. Ablations on multimodal neural recordings and a spiking speech benchmark show that temporal resolution and memory horizon jointly determine what a branch can capture: the update interval controls the temporal granularity of input aggregation, while the memory parameter controls the nominal span of represented history. In multimodal neural decoding, branches need sufficient coverage to capture discriminative trends spanning several seconds. In tasks that depend on local timing patterns, finer resolution preserves transient structure. Branch-combination ablations further show that, when a task requires both fine temporal detail and longer context, multiple update rates can provide complementary information and outperform the tested single-scale configurations.

1 INTRODUCTION

Invasive neural recordings simultaneously contain signals such as action potentials, LFPs, and dopamine that evolve on different timescales (Schwerdt, 2025; Choi et al., 2025). Their rates of change can differ by several orders of magnitude, so aligning them to a common clock does not by itself resolve how histories at different rates should be retained and combined. Effective decoding must account for both rapid events and slower contextual changes (Figure 1A).

Spiking neural networks (SNNs) are a natural substrate for neural signal processing, as their event-driven dynamics share the communication vocabulary of the recorded tissue (Maass, 1997; Tavanaei et al., 2019; Eshraghian et al., 2023). Leaky integrate-and-fire (LIF) neurons retain recent input through membrane potential decay, and extensions such as learnable time constants (Fang et al., 2021), firing-rate adaptation (Bellec et al., 2018), and coupled compartments (Zhang et al., 2024; Zheng et al., 2024) widen the temporal window to varying degrees. Yet membrane dynamics alone impose a single exponential horizon on each neuron: information decays at a fixed rate irrespective of its origin or relevance. Structured memory modules such as the Legendre Memory Unit (Voelker et al., 2019) and the dual-memory pathway (Sun et al., 2026) provide richer history representations, but existing spiking realizations attach them at the layer level, forcing an entire population to share one temporal summary and obscuring the input-specific context that individual neurons might exploit. Recurrent networks (GRU, LSTM) learn temporal dependencies flexibly (Cho et al., 2014;

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.

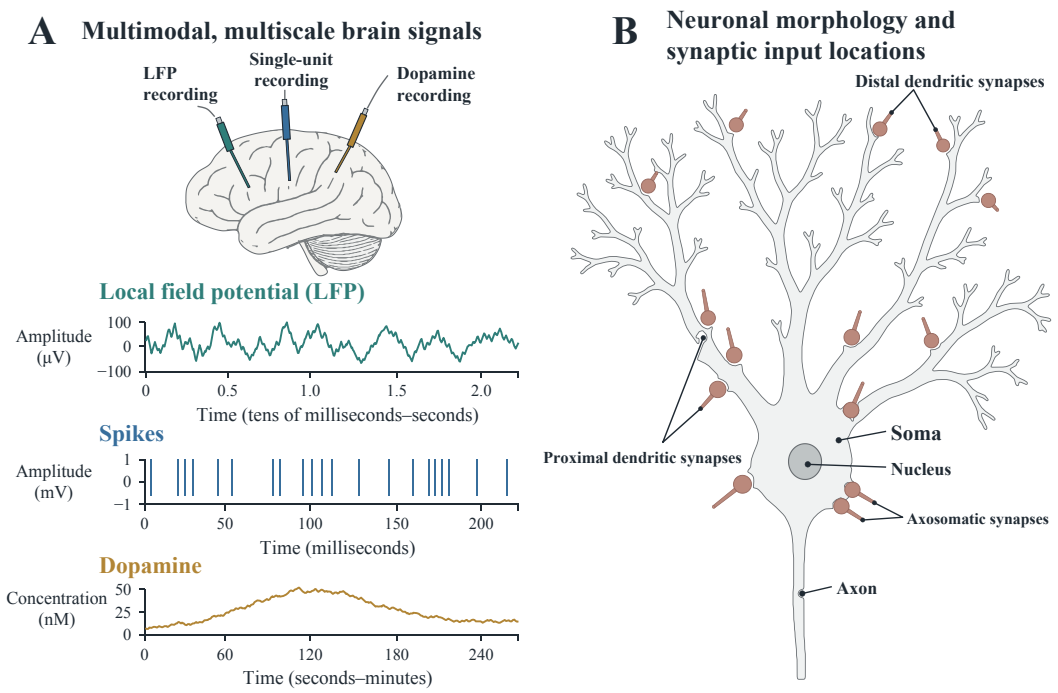

Figure 1: **Biological motivation. (A)** Invasive neural recordings contain LFP, spikes, and dopamine signals that evolve on distinct timescales. **(B)** Biological neurons integrate slow, fast, and direct inputs through distal dendritic, proximal dendritic, and axosomatic synapses. MRD-SNN maps this spatial–temporal organization onto a trainable multi-rate dendritic architecture.

Hochreiter & Schmidhuber, 1997), yet they lack inductive biases that reflect the physical structure of multi-scale neural signals.

Biological neurons offer a compelling alternative. Pyramidal cells in the cortex and hippocampus possess elaborate dendritic trees that are active computational elements (Poirazi et al., 2003; Mel, 1994) (Figure 1B). Proximal dendrites, situated close to the soma, receive fast-arriving synaptic inputs and relay them with minimal delay. Distal dendrites, extending far from the soma, integrate inputs over longer timescales; their cable properties and local ion-channel dynamics sustain currents for hundreds of milliseconds, supplying a slow contextual drive that modulates somatic firing (Larkum, 2013). In parallel, certain axo-somatic synapses bypass the dendritic tree entirely, providing an immediate pathway for time-critical signals. This architecture decomposes information by temporal scale at the level of a single neuron. Importantly, each dendrite samples only a sparse subset of the available synaptic inputs rather than receiving the full population signal—a structural constraint that encourages local, modality-aligned feature processing and reduces the wiring cost of maintaining many parallel branches (Guerguiev et al., 2017; Richards & Lillicrap, 2019).

Guided by this biological principle, we introduce **MRD-SNN** (*Multi-Rate Dendritic Spiking Neural Network*). Each neuron contains a direct somatic pathway and K sparse dendritic memory branches; each branch has its own update interval, Legendre state, and physical memory horizon, while its cached current remains available to the soma between refreshes. Different neurons maintain their own memory states.

We evaluate MRD-SNN on three cross-modal neural decoding tasks, Spiking Heidelberg Digits, and sequential MNIST. Our contributions are:

1. A spiking architecture with sparse, independently timed dendritic memory branches.
2. Ablation-based separation of update resolution and memory coverage.
3. Evidence that multi-rate branches provide complementary information across cross-modal neural recordings, speech, and sequential vision.

2 Related Work

Temporal dynamics in spiking neural networks. Existing SNNs extend neuronal timescales through learnable membrane constants, firing adaptation, or coupled compartments, including PLIF, ALIF, TC-LIF, DH-LIF, and TS-LIF (Fang et al., 2021; Bellec et al., 2018; Zhang et al., 2024; Zheng et al., 2024; Feng et al., 2025). These mechanisms enrich the response of individual neurons and allow past activity to influence subsequent integration and firing.

Structured memory and multi-rate computation. Structured memory modules, including LMU, LMUFormer, and DMP-SNN, extend history representations through polynomial states or dual pathways (Voelker et al., 2019; Liu et al., 2024; Sun et al., 2026). These representations supplement recurrent neuronal dynamics with an explicit summary of past inputs. Clockwork RNN and Phased LSTM reduce recurrent computation through periodic or time-gated updates (Koutník et al., 2014; Neil et al., 2016).

Dendritic computation in artificial networks. Dendritic computation studies show that multi-branch structures can process different input subspaces before somatic integration, improving representation quality, robustness, or parameter efficiency (Poirazi et al., 2003; Chavlis & Poirazi, 2025). Recent compartmental spiking models further introduce heterogeneous temporal dynamics (Zhang et al., 2024; Zheng et al., 2024). Together, these studies motivate treating a neuron as a collection of local processing units rather than a single weighted sum followed by a nonlinearity.

Building on this foundation, MRD-SNN extends structured temporal memory from layer-level sharing to soma-specific multi-branch representations, enabling independent control of history coverage and update resolution and providing a new architecture for processing multi-timescale signals with spiking neurons.

3 Method

3.1 Observation Arrival on a Physical Timeline

Let a trial contain timestamped observations $\{(t_i^m, z_i^m)\}$ for modality m. A dataset-specific base interval Δ defines processing ticks $t = 1, \ldots, L$. Observations are assigned to their causal availability ticks; a binned count or windowed feature becomes available at the endpoint of its window. Let $\mathcal{O}_t$ denote the feature coordinates that receive a new observation at tick t. For a first-layer weight vector $\mathbf{w}$, the observation-dependent projection is

$$P_t(\mathbf{w}) = \sum_{i \in \mathcal{O}_t} w_i \hat{z}_{t,i}, \qquad \hat{z}_{t,i} = \frac{z_{t,i} - \mu_i}{\max(\sigma_i, \epsilon)}. \tag{1}$$

Statistics μ_i, σ_i are computed from observed training values only. A feature without a new observation makes no contribution to this sum; an absent observation is *not* interpreted as a measured physiological zero or a repeated measurement. Internal membrane states, branch accumulators, and cached currents continue to evolve between observations. For hidden layers, $P_t(\mathbf{w}) = \mathbf{w}^\top \mathbf{h}_t^{l-1}$, where $\mathbf{h}_t^{l-1}$ is the preceding layer's spike vector after inter-layer dropout.

3.2 Dendritic Branches with Independent Update Schedules

Figure 2 summarizes the architecture. For soma j in layer l, the **direct somatic current** (analogous to the axo-somatic pathway) and the **dendritic branch drive** are:

$$I_{t,j}^{F,l} = P_t(\mathbf{W}_{F,j}^l) + b_{F,j}^l, \tag{2}$$

$$d_{t,j,k}^l = \mathrm{ReLU}\big(P_t(\mathbf{W}_{j,k}^l \odot \mathbf{R}_{j,k}^l) + b_{j,k}^l\big), \tag{3}$$

where $\mathbf{R}_{j,k}^l$ is a fixed binary connectivity mask that restricts each branch to a *sparse*, contiguous subset of input features. With a default receptive fraction of 0.25, each branch observes only one quarter of the input dimension, while the direct somatic pathway in Eq. (2) retains full-rank access to all features. The sparse mask restricts each branch to a local feature subset and encourages

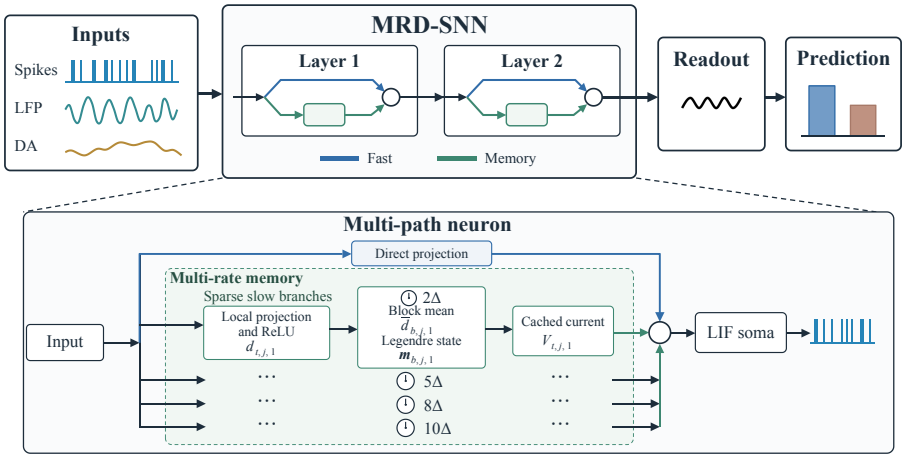

Figure 2: **MRD-SNN architecture.** The available input feeds a direct somatic projection and multiple sparse dendritic branches. Each branch computes a local block-averaged drive, updates a Legendre memory at its own interval, and produces a cached current that is fused with the direct pathway in the LIF soma.

complementary branch-specific histories. The direct somatic pathway retains access to all input features, so a branch can still combine multiple modalities within its receptive field.

Branch k updates once every n_k ticks, so its physical update interval is $\delta_k = n_k\Delta$. At the endpoint of block b, it consumes the completed block mean:

$$\bar{d}^l_{b,j,k} = \frac{1}{n_k} \sum_{t=(b-1)n_k+1}^{bn_k} d^l_{t,j,k}. \tag{4}$$

This time-averages the internal drive rather than averaging raw sensor samples. An online accumulator avoids storing individual per-tick values; its contents are cleared after each update. An unfinished final block is not flushed early, preserving the declared refresh schedule.

3.3 History Representation with Explicit Physical Horizons

Each branch maintains a state vector $\mathbf{m}^l_{b,j,k} \in \mathbb{R}^q$. Let $H_k > 0$ be its nominal memory horizon in seconds. Denoting the order-q Legendre matrices by $\mathbf{A}_0, \mathbf{B}_0$ (Appendix A), the continuous system $\dot{\mathbf{m}} = (\mathbf{A}_0\mathbf{m} + \mathbf{B}_0 d)/H_k$ is discretized over interval δ_k:

$$\mathbf{A}_k = \exp\Big(\frac{\delta_k}{H_k}\mathbf{A}_0\Big), \quad \mathbf{B}_k = \int_0^{\delta_k} \exp\Big(\frac{s}{H_k}\mathbf{A}_0\Big)\frac{\mathbf{B}_0}{H_k}\, ds, \tag{5}$$

$$\mathbf{m}^l_{b,j,k} = \mathbf{A}_k\, \mathbf{m}^l_{b-1,j,k} + \mathbf{B}_k\, \bar{d}^l_{b,j,k}. \tag{6}$$

Defining $\theta_k = H_k/\delta_k$ recovers the horizon in branch-update units, so $H_k = n_k\Delta\theta_k$. A learned readout $\mathbf{c}^l_k \in \mathbb{R}^q$ and soma-specific gain $g^l_{j,k}$ produce the dendritic current:

$$V^l_{t,j,k} = g^l_{j,k}\, (\mathbf{c}^l_k)^\top \mathbf{m}^l_{\lfloor t/n_k \rfloor, j, k}. \tag{7}$$

The initial memory and cached current are zero. A new current becomes available at the block endpoint and is held until the next update. Between refreshes, the state and its cached readout remain fixed, but this dendritic current still enters every somatic update—analogous to the sustained plateau potentials generated by distal biological dendrites.

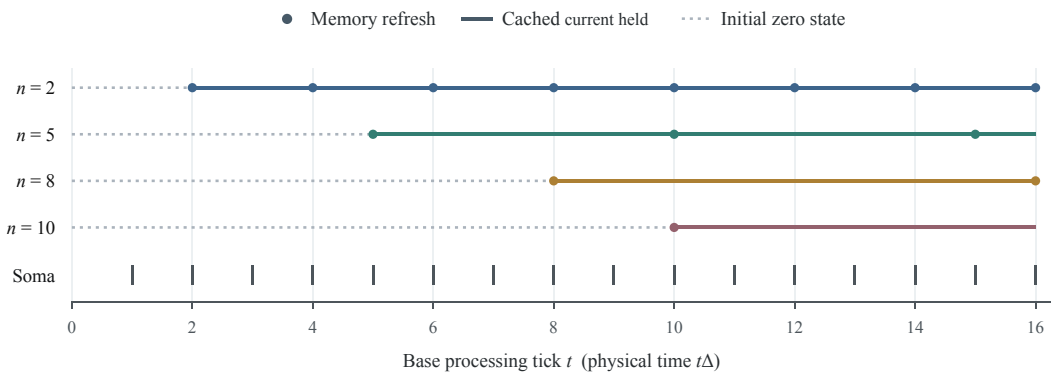

Figure 3: **Independent refresh schedules and persistent cached currents.** Branches with different update intervals refresh their memories at different base ticks, while each cached current is held between refreshes and remains available to the soma at every base processing tick.

3.4 Somatic Fusion and Trial Prediction

Learned coefficients fuse the direct pathway and dendritic currents:

$$I^l_{t,j} = \alpha_F \, I^{F,l}_{t,j} + \sum_{k=1}^{K} \alpha_k \, V^l_{t,j,k}, \tag{8}$$

$$\tilde{u}^l_{t,j} = \beta \, u^l_{t-1,j} + (1-\beta) \, I^l_{t,j}, \qquad s^l_{t,j} = \mathbf{1}[\tilde{u}^l_{t,j} \geq v_{\text{th}}], \tag{9}$$

$$u^l_{t,j} = (1 - s^l_{t,j}) \, \tilde{u}^l_{t,j}. \tag{10}$$

A single learned $(K{+}1)$-vector $\boldsymbol{\alpha}$, shared across layers and initialized to ones, sets the relative contribution of each pathway. Firing resets the membrane potential while preserving all dendritic memory states—a critical distinction from networks where a reset erases accumulated context. All trial-specific states reset between trials.

Two hidden layers of 128 neurons each feed a linear classifier. For a trial of valid length L_r, logits are averaged before softmax:

$$p(y \mid X_r) = \text{softmax}\Big(\frac{1}{L_r} \sum_{t=1}^{L_r} (\mathbf{W}_o \mathbf{s}^2_t + \mathbf{b}_o) \Big). \tag{11}$$

Training uses cross-entropy with backpropagation through time and a surrogate gradient for the spike threshold (Neftci et al., 2019).

3.5 Separating History Span, Refresh, and Somatic Use

The temporal properties of each dendritic branch are determined by its update interval n_k and nominal memory horizon $H_k = n_k \Delta \theta_k$. The update interval controls the temporal resolution and refresh frequency, whereas the memory horizon controls history coverage. A smaller n_k preserves fast changes more accurately but covers a shorter physical span; a larger n_k provides longer context but smooths transient fluctuations through block averaging. Increasing θ_k extends coverage but may weaken the representation of short-lived changes when the memory dimension is fixed.

Cached dendritic currents are used at every somatic tick even when the corresponding branch updates its memory less frequently. Multi-rate branches can therefore provide both rapid local responses and slowly varying long-range context without requiring memory refresh and somatic use to occur at the same frequency.

3.6 Why Multi-Rate Branches Provide Complementary Information

A memory branch with update interval n and horizon H can be viewed as a bandpass filter: block averaging attenuates frequencies above π/n, while the fixed Legendre basis limits representation of

components slower than approximately $1/H$. Single-rate branches face a fundamental trade-off: a smaller n preserves transients but limits the look-back span (see Appendix G); a larger n covers a longer history but smooths fine temporal details.

Multi-rate branches with update intervals $n_1 < n_2 < \cdots < n_K$ capture information in distinct frequency bands. Fast branches detect rapid events, while slow branches track gradual drifts. The learned fusion weights $\boldsymbol{\alpha}$ (Eq. (8)) form a *task-adaptive filter bank*, assigning greater weight to the most discriminative timescales. From an information-theoretic perspective, when the latent process decomposes into fast and slow components,

$$\mathbf{h}_t = \mathbf{h}_t^{\text{fast}} + \mathbf{h}_t^{\text{slow}},$$

branches tuned near these timescales achieve higher joint mutual information, provided they capture orthogonal temporal structure:

$$I\big(\mathbf{h}_t; \mathbf{z}_t^1, \ldots, \mathbf{z}_t^K\big) > \max_k I\big(\mathbf{h}_t; \mathbf{z}_t^k\big)\,.$$

4 Experimental Setup

4.1 Datasets

We construct three cross-modal neural decoding benchmarks from publicly available recordings (Even-Chen et al., 2019; 2022; Shin & Jadhav, 2025; Tang et al., 2023). Dataset details are summarized in Table 8 in Appendix D. The tasks are all binary classification problems, but differ in modality count, sampling interval, and temporal structure. Spike–LFP–DA combines spikes, LFP, and dopamine and has the richest temporal diversity; DANDI 000121 contains spikes and LFP at two sampling rates; and DANDI 000447 uses a single 50 ms rate for both modalities, providing a control in which multi-rate memory should matter least.

In addition to these cross-modal benchmarks, we evaluate MRD-SNN on two single-modality tasks to test whether multi-rate memory generalizes beyond neural recordings. We select SHD and sMNIST because their discriminative information is distributed unevenly over time, with informative events or structures occurring at different temporal densities. **Spiking Heidelberg Digits (SHD)** (Cramer et al., 2022) is a 20-class spiking speech-recognition task containing fast phonetic transitions and slower prosodic variation. **Sequential MNIST (sMNIST)** (LeCun et al., 1998) presents an image as a 784-step temporal stream, where local stroke information and broader digit structure contribute at different points in the sequence.

4.2 Experimental Protocol

Experiments use five-fold cross-validation with three training seeds (2026–2028), 100 epochs with AdamW (lr 0.005, weight decay 0.01, gradient clip 1.0), and model selection based on validation-set accuracy. The three seed results are first averaged within each fold, and the mean $\pm$ sample standard deviation is then computed across the five fold averages.

For the branch-combination experiments, SHD uses weight decay 0.1; sMNIST uses learning rate 0.001 and weight decay 0.01, with the learning rate halved after 50 epochs. Both retain the official test set and partition the official training set into five training/validation folds. SHD uses time-averaged logits, whereas sMNIST reads the final time step. MRD-SNN in Table 1 uses learnable positive $\theta_k = \exp(\rho_k)$, shared across layers. Its eight horizon parameters are initialized as $\theta_{\text{base}} 2^{-1+2k/7}$ for $k = 0, \ldots, 7$. The baseline values for the three datasets are set to 40, 5, and 16, respectively, based on their sample durations. The branch-combination and refresh-scheduling controls use fixed horizons.

5 Results

5.1 Cross-Modal Decoding Performance

We compare against eight baseline models spanning feedforward and adaptive spiking, dual-compartment and memory-augmented spiking, and gated recurrent architectures (Table 1). MRD-SNN achieves the highest mean accuracy on all three tasks. On Spike–LFP–DA, it reaches 77.67%,

Table 1: Test accuracy (%) on three neural decoding tasks.

Model	Parameters	Spike–LFP–DA	DANDI 000121	DANDI 000447
FSNN (Maass, 1997)	57,087 / 60,437	66.47±4.54	80.21±3.15	72.66±2.09
PLIF (Fang et al., 2021)	57,089 / 59,948	65.95±6.08	79.19±3.39	72.07±2.28
ALIF (Bellec et al., 2018)	57,087 / 59,946	65.87±4.89	72.92±2.35	63.53±3.07
TC-LIF (Zhang et al., 2024)	57,091 / 59,950	64.79±6.75	80.60±3.46	71.87±1.82
DH-LIF (Zheng et al., 2024)	56,982 / 59,498	69.30±7.61	82.79±1.34	71.69±2.75
DMP-SNN (Sun et al., 2026)	57,066 / 60,360	66.55±5.18	81.09±2.20	72.47±2.24
LMU (Voelker et al., 2019)	57,100 / 59,908	65.23±5.37	81.15±1.96	73.40±3.88
GRU (Cho et al., 2014)	56,289 / 60,595	64.27±6.13	83.07±2.16	73.25±3.14
MRD-SNN (K=8)	57,523 / 60,339	**77.67±4.89**	**87.34±1.09**	**77.91±3.12**

Table 2: Memory-horizon comparison with a fixed update interval on the three neural decoding tasks.

Dataset	n	Horizon H (s)	Accuracy (%)
Spike–LFP–DA	2	0.8	65.27±6.15
Spike–LFP–DA	2	4.0	73.73±4.25
DANDI 000121	2	0.1	82.45±2.65
DANDI 000121	2	0.5	86.12±1.23
DANDI 000447	2	1.6	75.24±1.59
DANDI 000447	2	8.0	75.88±1.93

exceeding the strongest baseline, DH-LIF, by 8.37 percentage points. It reaches 87.34% on DANDI 000121 and 77.91% on DANDI 000447, exceeding the strongest respective baselines by 4.27 and 4.51 points. Relative to FSNN, the gains are 11.20, 7.13, and 5.25 points on Spike–LFP–DA, DANDI 000121, and DANDI 000447, respectively.

The gain over the strongest baseline is largest on Spike–LFP–DA and smallest on DANDI 000121. MRD-SNN achieves the highest mean on all three tasks, with its clearest advantage on the benchmark containing the most diverse input timescales.

5.2 Update Resolution and Memory Horizon

Each dendritic branch has two temporal parameters: the update interval n_k, which controls how finely within-block fluctuations are preserved (*resolution*), and the memory horizon $H_k = n_k\theta_k\Delta$, which controls how far back the branch remembers (*coverage*). The default architecture varies both simultaneously. Here we disentangle their contributions.

The role of memory horizon. At the same update interval, extending the memory horizon produces larger gains on datasets with richer temporal scales. Spike–LFP–DA improves from 65.27% to 73.73%, a gain of 8.46 percentage points, while DANDI 000121 improves from 82.45% to 86.12%, a gain of 3.67 points.

In Table 3, we next examine update resolution while keeping the nominal horizon fixed.

At the fixed horizon, the effect of the update interval likewise depends on the temporal structure of the dataset. On Spike–LFP–DA and DANDI 000121, the faster update exceeds the slower update by 1.08 and 0.60 percentage points, respectively, indicating that finer temporal resolution can provide a measurable benefit. On DANDI 000447, the two settings differ by only 0.33 points.

These results indicate that when the input contains richer temporal scales or more temporally heterogeneous information, extending historical coverage is more likely to capture discriminative long-term changes, while varying the update resolution can provide additional benefits. Under the present evaluation protocol, the three-modality dataset contains the richest temporal-scale information and therefore shows the largest improvement. DANDI 000447 contains fewer exploitable differences in temporal scale, so the resulting benefit is limited.

Table 3: Update-interval comparison at a fixed nominal memory horizon.

Dataset	n	Horizon H (s)	Accuracy (%)
Spike–LFP–DA	2	4.0	73.73±4.25
Spike–LFP–DA	10	4.0	72.65±3.91
DANDI 000121	2	0.5	86.12±1.23
DANDI 000121	10	0.5	85.52±1.48
DANDI 000447	2	8.0	75.88±1.93
DANDI 000447	10	8.0	76.21±2.17

Table 4: Branch combinations with fixed θ (4 for SHD, 49 for sMNIST). Fast and slow branches use $n = 2$ and $n = 20$, respectively. Dual-branch configurations are parameter-matched within each dataset. Accuracy (%).

Branch configuration	SHD	sMNIST
Single fast	78.82±0.36	54.11±0.23
Single slow	76.46±0.85	84.93±0.58
Dual fast+slow	**82.68±0.65**	**93.47±0.10**
Dual slow+slow	78.02±0.39	87.19±0.06
Dual fast+fast	80.41±0.68	58.81±0.14

5.3 Multi-Rate Complementarity

Branches operating at different update rates can capture complementary information. We test this by comparing a mixed-rate pair against same-rate pairs at matched parameter counts.

The mixed-rate pair reaches 82.68% on SHD and 93.47% on sMNIST. On SHD, it exceeds the single-fast branch by 3.86 points and the parameter-matched dual-fast and dual-slow pairs by 2.27 and 4.66 points. On sMNIST, the single-slow branch substantially outperforms the single-fast branch (84.93% versus 54.11%), yet adding a fast branch yields a higher accuracy than adding another slow branch: 93.47% versus 87.19%, a gain of 6.28 points at matched capacity. Thus, the preferred single-branch rate differs across tasks, but the mixed-rate pair performs best on both. These comparisons support complementary contributions from the two branch configurations rather than an explanation based solely on parameter count.

5.4 Design Choices

Soma-specific memory outperforms shared memory. In the default architecture, each of the 128 neurons per layer maintains its own set of dendritic memory states. We test whether this private context is necessary by pooling the dendritic drives into groups before memory integration: $G{=}1$ (all neurons share a single state per branch), $G{=}8$ (groups of 16), or $G{=}128$ (independent, the default). Table 5 shows that independent memory consistently outperforms layer-wide sharing, with gains of 4.75, 5.79, and 6.36 percentage points on Spike–LFP–DA, DANDI 000121, and DANDI 000447, respectively. Different neurons receive different feature subsets through the sparse connectivity masks and accumulate different temporal contexts; pooling erases this input-specific history.

5.5 Branch Count with Learnable Memory Horizons

We ask whether distributing a fixed memory-state budget across different numbers of learnable-horizon branches ($K \in \{1, 2, 4, 8\}$) improves decoding. We set $q = 80/K$, so every configuration has the same total memory-state count $Kq = 80$, allowing a controlled comparison of how this budget is allocated across branches.

Each branch learns a positive horizon parameter $\theta_k = \exp(\rho_k)$, from which its Legendre transition matrices are recomputed differentiably. Branches are initialized at distinct horizons while preserving the dataset-specific baseline at the level of their geometric mean; the initialization details are given in the appendix.

Table 5: Independent versus shared memory (K=4). All configurations have two hidden layers of 128 neurons.

Memory organisation	Spike–LFP–DA	DANDI 000121	DANDI 000447
G=1 (all shared)	70.45±6.20	80.23±1.75	69.06±3.38
G=8 (groups of 16)	74.12±5.79	83.54±2.47	69.93±4.10
G=128 (independent)	75.20±5.68	86.02±1.79	75.42±2.46

Table 6: Branch-count comparison with learnable θ and fixed $Kq = 80$.

K	q	Spike–LFP–DA	DANDI 000121	DANDI 000447	SHD
1	80	67.62±6.00	81.02±3.62	71.64±2.96	78.42±0.84
2	40	74.00±6.18	83.78±2.14	71.83±2.57	81.13±1.02
4	20	75.75±5.13	85.78±2.46	74.09±3.09	81.56±0.71
8	10	**77.67±4.89**	**87.34±1.09**	**77.91±3.12**	**82.33±0.52**

Table 6 shows that every tested multi-branch configuration exceeds the single-branch mean on all four datasets. Relative to $K = 1$, the best multi-branch configuration improves accuracy by 10.05, 6.32, 6.27, and 3.91 points on Spike–LFP–DA, DANDI 000121, DANDI 000447, and SHD, respectively.

On Spike–LFP–DA, accuracy rises from 67.62% at $K = 1$ to 75.75% at $K = 4$ and 77.67% at $K = 8$; the highest mean occurs at $K = 8$ on all four datasets. Thus, distributing the fixed memory-state budget across more branches generally improves performance, with diminishing gains at larger branch counts.

6 Conclusion

MRD-SNN equips spiking neurons with sparse, independently clocked dendritic memory branches whose horizons are set in physical time. On three cross-modal neural decoding benchmarks, it achieves the highest accuracy, with an 8.37-point advantage on the most temporally diverse task. The controlled comparisons show that horizon, refresh resolution, and branch diversity play distinct roles. Longer coverage helps when the discriminative signal evolves over seconds, finer updates help when local timing matters, and mixed rates are most useful when both forms of evidence are present. The architecture is therefore most effective when branch schedules are matched to the temporal structure of the signals being decoded.

Reproducibility Statement

The recurrences, discretisation, update timing, and readout are fully specified in Section 3. Source datasets and their preprocessing are detailed in Section 4 and Appendix D.

Ethics Statement

This study reuses publicly released neural recordings and benchmark datasets. No new animal or human data were collected.

AI Use Statement

Generative AI tools were used for language polishing, translation, and literature searches. The researchers reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *NeurIPS*, 2018.

S. Chavlis and P. Poirazi. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. *Nature Communications*, 16:943, 2025.

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

J. Choi et al. Aseptic, semi-sealed cranial chamber implants for chronic multi-channel neurochemical and electrophysiological neural recording in nonhuman primates. *J. Neuroscience Methods*, 420:110467, 2025.

B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke. The Heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 33(7):2744–2757, 2022.

J. K. Eshraghian et al. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

N. Even-Chen, B. Sheffer, S. Vyas, S. I. Ryu, and K. V. Shenoy. Structure and variability of delay activity in premotor cortex. *PLOS Comput. Biol.*, 15:e1006808, 2019.

N. Even-Chen et al. Structure and variability of delay activity in premotor cortex. DANDI Archive, version 0.220124.2156, 2022.

W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *ICCV*, 2021.

W. Fang et al. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Science Advances*, 9:eadi1480, 2023.

S. Feng, W. Feng, X. Gao, P. Zhao, and Z. Shen. TS-LIF: A temporal segment spiking neuron network for time series forecasting. In *ICLR*, 2025.

J. Guerguiev, T. P. Lillicrap, and B. A. Richards. Towards deep learning with segregated dendrites. *eLife*, 6:e22901, 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber. A Clockwork RNN. In *ICML*, PMLR 32(2):1863–1871, 2014.

M. E. Larkum. A cellular mechanism for cortical associations: An organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151, 2013.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Z. Liu, G. Datta, A. Li, and P. A. Beerel. LMUFormer: Low complexity yet powerful spiking model with Legendre Memory Units. In *ICLR*, 2024.

W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

B. W. Mel. Information processing in dendritic trees. *Neural Computation*, 6(6):1031–1085, 1994.

E. O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Proc. Mag.*, 36(6):51–63, 2019.

D. Neil, M. Pfeiffer, and S.-C. Liu. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *NeurIPS*, 2016.

P. Poirazi, T. Brannon, and B. W. Mel. Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999, 2003.

B. A. Richards and T. P. Lillicrap. Dendritic solutions to the credit assignment problem. *Current Opinion in Neurobiology*, 54:28–36, 2019.

H. Schwerdt. FSCV dopamine, spike, and LFP data used for validating chronic electrode implanted in aseptic semi-sealed chamber. Zenodo, 2025.

J. D. Shin and S. P. Jadhav. Novel-familiar-novel WTrack (CA1-PFC). DANDI Archive, version 0.250724.2224, 2025.

P. Sun et al. Algorithm–hardware co-design of neuromorphic networks with dual memory pathways. *Nature Machine Intelligence*, 8:901–912, 2026.

W. Tang, J. D. Shin, and S. P. Jadhav. Geometric transformation of cognitive maps for generalization across hippocampal-prefrontal circuits. *Cell Reports*, 42:112246, 2023.

A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

A. R. Voelker, I. Kajić, and C. Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In *NeurIPS*, 2019.

S. Zhang et al. TC-LIF: A two-compartment spiking neuron model for long-term sequential modelling. In *AAAI*, 38(15):16838–16847, 2024.

H. Zheng et al. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15:277, 2024.

A Legendre Recurrence

For zero-based indices, $(\mathbf{A}_0)_{ij} = (2i+1)[-\mathbf{1}_{i<j} + (-1)^{i-j+1}\mathbf{1}_{i\geq j}]$ and $(\mathbf{B}_0)_i = (2i+1)(-1)^i$. The branch kernel $\kappa_k[r] = g_k\, \mathbf{c}_k^\top \mathbf{A}_k^r \mathbf{B}_k$ gives $V_{bn_k,k} = \sum_{a=1}^{b} \kappa_k[b-a]\, \bar{d}_{a,k}$.

B Modality Ablation

We compare MRD-SNN and FSNN under different input subsets on the Spike–LFP–DA benchmark. Both models use two hidden layers of 128 neurons. MRD-SNN uses eight dendritic branches with $q = 10$ and fixed $\theta_k = 40$ for every branch. These settings are held constant across all input subsets, including the full-modality condition. All conditions use the same five folds and validation-based checkpoint selection. Absent modalities are zeroed after training-set normalization while all input slots and model parameters are retained.

Table 7: Modality ablation on Spike–LFP–DA. Test accuracy (%) and the MRD-SNN minus FSNN difference.

Input	MRD-SNN (%)	FSNN (%)	Difference (points)
Spike only	50.68±0.86	50.60±1.49	+0.08
DA only	52.95±2.31	49.92±2.42	+3.03
LFP only	76.35±6.38	66.75±4.31	+9.60
Spike + LFP	**77.87±5.58**	65.91±4.82	+11.96
LFP + DA	77.27±5.08	66.39±4.61	+10.88
Spike + LFP + DA	76.11±4.85	66.47±4.07	+9.64

The positive gaps for DA, LFP, and their combinations indicate that MRD-SNN extracts modality-specific information that the feedforward reference does not exploit as effectively. Spike–LFP is the strongest condition (77.87%), and its 11.96-point advantage is the largest in the table. Adding the spike stream to LFP improves MRD-SNN over LFP alone, consistent with the spike representation supplying rich event and frequency-channel information. Adding DA to Spike–LFP does not further improve the mean in this setting, so the result supports complementary modality use rather than a claim that every added stream must increase accuracy.

C Configuration

Two layers of 128 LIF neurons (v_{th}=1, hard reset). q=10 LMU order, intervals $(2, 5, 8, 10, 12, 15, 20, 30)$, contiguous 25% masks. Horizon parameter θ=40/5/16/4 for primary/000121/000447/SHD. AdamW with lr = 0.005, weight decay = 0.01, gradient clip = 1.0, 100 epochs, dropout = 0.2, batch 128. Five-fold CV, training seeds 2026–2028, validation-based selection. PyTorch 2.11, SpikingJelly 0.0.0.0.14 (Fang et al., 2023).

D Dataset Processing

The tasks, sources, modalities, sampling intervals, trial counts, and feature counts of the three neural decoding benchmarks are summarized below.

Table 8: Neural decoding benchmarks. Temporal diversity decreases from left to right.

	Spike–LFP–DA	DANDI 000121	DANDI 000447
Task	Reward (2-class)	Direction (2-class)	Trajectory (2-class)
Source	Pittsburgh	Stanford	Brandeis
Modalities	Spike, LFP, DA	Spike, LFP	Spike, LFP
Sampling intervals	10 / 50 / 100 ms	10 / 50 ms	50 / 50 ms
Trials	836	1,280	723
Features	8	14	14

Pittsburgh: Zenodo 14825992 (CC BY 4.0). LFP: notch at 50/100/150 Hz (Q=30), Butterworth bandpass at 1–4, 4–8, 8–13, 13–30, 30–70, 70–150 Hz, exponential smoother (τ=100 ms), log power every 50 ms. Spikes: 10 ms bin counts. Dopamine: forward-fill after first valid sample. Stanford: DANDI 000121 v0.220124.2156 (Even-Chen et al., 2022). Brandeis: DANDI 000447 v0.250724.2224 (Shin & Jadhav, 2025). SHD: official download (Cramer et al., 2022), 700 channels merged to 140 by 5-channel bins, binary event presence, 100 bins of 14 ms.

E Full-Rate Versus Multirate Refresh

This control separates memory refresh from somatic use. We hold each physical memory horizon fixed and compare four- and eight-branch schedules with full-rate updates, $n_k = 1$. The full-rate discretization uses $\theta_k = H_k/\Delta$, while trainable tensors and masks match the multirate reference. For a trial of L ticks, a soma performs $\sum_k \lfloor L/n_k \rfloor$ memory updates rather than KL; input projections and somatic updates still run at every tick.

Table 9: Full-rate ($n_k = 1$) versus multirate computation at fixed physical horizons and fixed θ. Update reduction uses actual test-trial lengths; speedup is the median across ten fold–seed runs, each measured on its median-length test trial.

Dataset	K	Full-rate accuracy	Multirate accuracy	Updates reduced (%)	Speedup
Spike–LFP–DA	4	74.72±4.45	75.20±5.68	76.88	1.56
Spike–LFP–DA	8	76.19±5.07	76.11±4.85	85.54	1.78
DANDI 000121	4	85.62±2.09	86.02±1.79	77.22	1.55
DANDI 000121	8	86.61±1.48	86.54±2.21	85.91	1.77
DANDI 000447	4	75.19±2.52	75.42±2.46	76.97	1.56
DANDI 000447	8	75.89±3.39	75.15±2.51	85.63	1.78

Across the six dataset–capacity pairs, multirate accuracy differs from full-rate computation by at most 0.74 percentage points. It is slightly higher in three configurations and lower in the other three. Memory updates decrease by 76.88–77.22% for $K = 4$ and 85.54–85.91% for $K = 8$, yielding streaming speedups of 1.55–1.56× and 1.77–1.78×, respectively. Timing uses batch size one, two warm-up runs, and five alternating measurements of each schedule on the same device; these measurements describe uncompiled PyTorch execution and do not estimate device energy consumption.

F Memory Horizon, Update Interval, and Impulse Retention

The horizon parameter θ sets the nominal memory span in branch updates, whereas the update interval n converts those updates into physical time: $H = n\theta\Delta t$. This appendix illustrates two points: the residual trace of an input becomes small beyond the nominal horizon in the tested dynamics, and the same number of updates spans more physical time in a slow branch. We examine the fixed LMU state before the learned temporal readout and the LIF soma, so retention here refers to the remaining input trace in that state.

Dynamics on the branch clock. For memory order q, let A_0 and B_0 denote the unscaled continuous-time LMU matrices. With indices $i, j = 0, \ldots, q-1$, the implementation uses

$$(A_0)_{ij} = (2i+1)\begin{cases} -1, & i < j, \\ (-1)^{i-j+1}, & i \geq j, \end{cases} \qquad (B_0)_i = (2i+1)(-1)^i. \tag{12}$$

Zero-order-hold discretization with a unit branch-clock step gives

$$\bar{A}_\theta = \exp(A_0/\theta), \qquad \bar{B}_\theta = \int_0^1 \exp(sA_0/\theta)\,\frac{B_0}{\theta}\,ds. \tag{13}$$

For a scalar dendritic drive x_t on the base clock, complete blocks are averaged before updating the state:

$$\bar{x}_j = \frac{1}{n} \sum_{t=(j-1)n+1}^{jn} x_t, \qquad m_j = \bar{A}_\theta m_{j-1} + \bar{B}_\theta \bar{x}_j. \tag{14}$$

The nominal physical horizon is $H = n\theta\Delta t$. Thus $(n, \theta) = (2, 400)$ and $(10, 80)$ both have $H = 800\Delta t$, or 8 s when $\Delta t = 10$ ms. At their common update boundaries, the autonomous state transitions satisfy

$$\bar{A}_{400}^{5} = \bar{A}_{80}. \tag{15}$$

This identity is exact in real arithmetic: five updates of the first branch and one update of the second span the same ten base-clock steps. It concerns propagation of an existing state without further input. The driven systems are not identical: the first receives five separate two-step averages, whereas the second receives one ten-step average. The latter cannot distinguish permutations of the scalar drive within a complete block. This distinction separates temporal aggregation resolution from the timescale of autonomous memory evolution.

How the input trace decays beyond the nominal horizon. We set $q = 10$, $m_0 = 0$, and apply $x_1 = 1$, $x_t = 0$ for $t > 1$ to the scalar drive before block averaging. At a complete-block boundary $t = jn$, the impulse contribution is

$$m_j = \bar{A}_\theta^{j-1} \frac{\bar{B}_\theta}{n}, \qquad R_{n,\theta}(t) = 100\, \frac{\|\bar{A}_\theta^{t/n-1} \bar{B}_\theta/n\|_2}{\|\bar{B}_\theta/n\|_2}. \tag{16}$$

Here R measures the residual state norm relative to that branch's first write. It is neither a fraction of recoverable information nor a measure of classification contribution. Table 10 uses the discretized matrices stored by the current implementation. No model is trained for this diagnostic.

Table 10: Relative impulse-state norm R (%) at matched nominal horizons, $q = 10$. Each column is normalized by its own first-write norm; the denominators differ. Base-clock time assumes $\Delta t = 10$ ms.

Base step t	Time (s)	$n = 2,\ \theta = 400$	$n = 10,\ \theta = 80$
100	1	29.64	39.61
200	2	25.80	34.73
400	4	22.17	30.01
600	6	18.07	24.57
800	8	9.68	13.45
1000	10	1.21	1.67
1200	12	0.262	0.368
1600	16	0.021	0.029

Why the same update count covers more time in a slow branch. For fixed θ and q, branches follow the same normalized impulse-retention curve as a function of the number of updates. However, j updates span $jn\Delta t$ in physical time. For example, with $\theta = 40$ and $\Delta t = 10$ ms, the fortieth update occurs at 0.8 s for $n = 2$ but at 4.0 s for $n = 10$. Both retain 19.70% of their own first-write norm at that update, so the slower branch stretches the same relative retention profile over five times as much physical time. This longer span comes with coarser input averaging.

This fixed-θ comparison differs from Table 10, where θ is adjusted to keep $n\theta$ equal.

G A Fast Branch Alone Does Not Ensure Reliable Long-Sequence Decoding

Task and comparison. We construct an 8192-step pulse-order XOR task to test whether one frequently updated, long-horizon memory can handle both distant information and local order. A binary variable A is encoded at the beginning of the sequence and must remain available for seven subsequent queries. At each query, B is encoded by the order of two channel-group pulses, $X \to Y$

or $Y \to X$, separated by two steps. The two orders have equal pulse counts. After both pulses, the model predicts A XOR B. Success therefore requires both remote retention and recent order discrimination.

We compare a single high-rate, long-horizon branch with a high-rate short-horizon branch paired with a low-rate long-horizon branch (Table 11). Both configurations have a maximum nominal span of 8192 input steps. The low-rate branch averages blocks of 64 inputs and holds its readout between updates. All branches use ReLU. The horizons are fixed in this experiment.

Table 11: Branch settings for the long-sequence XOR experiment. n is the update interval, q the memory order, and $n\theta$ the nominal span in input steps.

Configuration / branch	q	n	θ	$n\theta$
Single: high-rate, long-horizon	10	1	8192	8192
Single: matched state budget	20	1	8192	8192
Dual: high-rate, short-horizon	10	1	10	10
Dual: low-rate, long-horizon	10	64	128	8192

Model and evaluation. This diagnostic uses 40 input channels, one layer of 128 spiking neurons, and a linear binary readout.

The fixed split contains 512 training, 512 validation, and 2048 test sequences. We train from scratch with seeds 2026–2028 for 500 epochs, batch size 32, AdamW learning rate 0.005, gradient clipping at 1, and a learning-rate reduction by 0.5 every 100 epochs. The loss combines cross-entropy at all seven queries with a firing-rate penalty weighted by 0.1; no auxiliary A or B labels are used. Checkpoints are selected by final-query validation accuracy, breaking ties by seven-query cross-entropy. Selected checkpoints are independently rechecked and hashed before testing, without test-time recalibration or reselection. The reported metric is final-query test accuracy.

Table 12: Final-query test accuracy (%) on 2048 sequences. Mean $\pm$ sample SD is computed across three training seeds.

Configuration	2026	2027	2028	Mean $\pm$ SD
Single ($q = 10$)	51.56	53.27	52.10	52.31$\pm$0.87
Single ($q = 20$)	51.42	71.83	50.00	57.75$\pm$12.21
Short + long	100.00	100.00	99.95	**99.98$\pm$0.03**

Results and interpretation. The single high-rate, long-horizon branch with $q = 10$ remains near chance across all three seeds, whereas the short-plus-long configuration reaches 99.98$\pm$0.03% (Table 12), a gain of 47.67 percentage points. Selected-checkpoint validation accuracies are 51.82$\pm$0.41% and 100.00$\pm$0.00%, respectively. Increasing the single branch to $q = 20$ matches the dual configuration's 2,560 memory-state scalars but yields 57.75$\pm$12.21% test accuracy. The dual configuration is both more accurate and more consistent across seeds in this comparison; its total parameter count remains higher (8,857 versus 7,320).

Updating every step does not by itself ensure a useful representation of recent order. A single branch must represent both the initial bit and the recent pulse sequence in the same state; doubling its memory order does not close the observed gap. Separate states and readouts let the short branch represent recent order while the long branch retains distant context. The results support this division of memory roles as a way to reconcile local temporal discrimination with long-term retention.