Parallel Delayed Memory Units for Enhanced Temporal Modeling in Biomedical and Bioacoustic Signal Analysis

Pengfei Sun, Wenyu Jiang, *Senior Member, IEEE*, Paul Devos, and Dick Botteldooren, *Senior Member, IEEE*

***Abstract*—Advanced deep learning architectures, particularly recurrent neural networks (RNNs), have been widely applied in audio, bioacoustic, and biomedical signal analysis, especially in data-scarce environments. While gated RNNs remain effective, they can be relatively over-parameterised and less training-efficient in some regimes [1], [2], while linear RNNs tend to fall short in capturing the complexity inherent in bio-signals. To address these challenges, we propose the Parallel Delayed Memory Unit (PDMU), a delay-gated state-space module for short-term temporal credit assignment targeting audio and bioacoustic signals, which enhances short-term temporal state interactions and memory efficiency via a gated delay-line mechanism. Unlike previous Delayed Memory Units (DMU) that embed temporal dynamics into the delay-line architecture, the PDMU further compresses temporal information into vector representations using Legendre Memory Units (LMU). This design serves as a form of causal attention, allowing the model to dynamically adjust its reliance on past states and improve real-time learning performance. Notably, in low-information scenarios, the gating mechanism behaves similarly to skip connections by bypassing state decay and preserving early representations, thereby facilitating long-term memory retention. The PDMU is modular, supporting parallel training and sequential inference, and can be easily integrated into existing linear RNN frameworks. Furthermore, we introduce bidirectional, efficient, and spiking variants of the architecture, each offering additional gains in performance or energy efficiency. Experimental results on diverse audio and biomedical benchmarks demonstrate that the PDMU significantly enhances both memory capacity and overall model performance.**

***Index Terms*—bio-signals, EEG, State-space model, Legendre Memory Units, Delay lines, Spiking neural networks, Short-term memory, Parallel training, Real-time inference**

I. Introduction

AUDIO, bioacoustic, and biomedical signals contain valuable latent information that is essential for understanding and monitoring individual health conditions [3], [4]. Recent advances in machine learning have greatly improved the processing and analysis of audio and biomedical signals [3]–[6]. Gated RNNs [7], designed specifically for sequential data, offer a low-complexity solution with a time complexity of O(N). Gated RNNs process input signals by feeding them into memory cells that aggregate and redistribute information over time through complex network dynamics. Another widely used deep learning model is the Transformer [8], [9], which employs a self-attention mechanism to assign importance to different input elements, effectively capturing long-term dependencies in the data. Unlike RNNs, Transformers are stateless architectures, allowing for parallel training and significantly accelerating the training process. RNNs and Transformers are particularly well-suited for audio-based tasks, as they can handle the temporal dependencies that are critical for sequential data processing. These architectures, along with their various adaptations, form the backbone of many audio-based deep learning models.

Despite considerable advancements in applying recurrent neural networks to bio-signal tasks, several challenges remain. The increasing volume of data necessitates faster processing capabilities, as well as the ability to effectively address tasks that demand the integration of both long-term and short-term temporal dependencies. Moreover, for these models to be practical in real-world environments, they should be optimized for deployment in resource-limited devices and edge computing scenarios, where lightweight architectures and real-time inference are essential. However, current models often fail to fully meet these requirements of training speed and efficiency needed in real-world healthcare environments. Gated RNNs, while effective in various contexts, are hindered by the vanishing gradient problem, which compromises their ability to capture and maintain long-term dependencies across extended sequences. The nonlinear memory state updates inherent in Gated RNNs also result in longer training times and pose significant challenges for parallelization, limiting their overall efficiency. Transformer-based architectures, although proficient at modeling long-term dependencies through self-attention mechanisms, require the long sequence to be available before computation can begin. This constraint not only impairs their ability to process sequential data efficiently but also leads to computational and memory overheads that scale quadratically with sequence length, making them less suitable for energy-constrained environments [10]. Carefully designed state-space models (e.g., Mamba [11]) can be competitive. Legendre Memory Units (LMUs) [12], [13], a type of linear RNN, have emerged as a promising solution for retaining memory information over theoretically infinite time horizons. They achieve this by integrating a linear time-invariant (LTI) system, thereby enabling efficient parallelization while maintaining the sequential nature of inference. However, despite their strength in handling very long-term dependencies, this kind of state-space model does not fully leverage the dynamic, short-term interactions that are vital for processing audio and biomedical signals [14]. In light of these limitations, this work aims to enhance the functionality of linear RNNs for bio-signal

This work was supported in part by the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" and the Research Foundation - Flanders under grant number G0A0220N, and in part by Programmatic grant no. A1687b0033 from the Singapore government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain). Corresponding author: dick.botteldooren@ugent.be.

Pengfei Sun, Paul Devos, and Dick Botteldooren are with the Department of Information Technology, WAVES Research Group, Ghent University. Present address (Pengfei Sun): Department of Electrical and Electronic Engineering, Imperial College London (e-mail: p.sun@imperial.ac.uk).

Wenyu Jiang is with Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore

processing by increasing their memory capacity for short-term memory interactions.

In this paper, we propose a novel RNN variant referred to as the Parallel Delayed Memory Unit (PDMU). This architecture builds upon the Legendre Memory Unit. The PDMU supports parallel training while processing real-time data sequentially. Additionally, the delayed memory unit facilitates interaction between the hidden states in the temporal dimension and enhances temporal credit assignment, thereby improving the memory capacity of the RNN. To further improve performance and reduce computational complexity, several variants of the PDMU are proposed and analyzed. These modifications achieve state-of-the-art or competitive performance compared to other works and show significant improvement over the baseline. Our primary contributions can be summarized as follows:

- We introduce a novel Parallel Delayed Memory Unit (PDMU), a delay-gated state-space module that explicitly captures short-horizon temporal interactions. Coupled with a linear RNN cell, PDMU retains fully parallel training while preserving causal, real-time inference.
- We devised several variants to further enhance performance or improve energy efficiency, including:
  - Bi-directional PDMU (Bi-PDMU): Captures temporal relationships from both the past and the future. Choose for offline/bidirectional settings.
  - Efficient PDMU (EPDMU): Activates only one delay gate at a time to optimize energy usage. Choose when the compute/energy budget is tight (lower MACs).
  - Spiking DMU: Incorporates spiking neurons to further increase energy efficiency. Choose for neuromorphic/low-power deployment or event-based inputs.
- We demonstrate PDMU's superior performance across a range of audio and bio-signal benchmark tasks, particularly in scenarios with limited data, as commonly encountered in real-world healthcare environments where optimization for edge computing deployment is crucial. PDMU not only significantly enhances signal decoding but also consistently outperforms traditional RNNs and other baseline models.

The structure of this paper is as follows: Section II provides the relevant methods, situating our work within the current research context. Section III details the proposed methodology. Section IV presents the evaluation of our approach on five benchmark tasks, including an ablation study and model analysis. Finally, Section V offers a summary and conclusions of the study.

II. Related Work

A. Legendre Memory Unit and Parallel Training

The Legendre Memory Unit (LMU), introduced by Voelker et al. [12], effectively captures and encodes temporal dependencies in sequential data through the use of Legendre polynomials. This memory cell is constructed by integrating a single-input delay network (DN) with a nonlinear dynamical system. The DN orthogonalizes the input signal over a sliding window of length θ, while the nonlinear system leverages this orthogonalized memory to compute various functions over time. Mathematically, given an input scalar function u(t), the state update can be described as follows:

$$\mathbf{m}'(t) = \mathbf{A}\mathbf{m}(t) + \mathbf{B}u(t) \tag{1}$$

where $\mathbf{m}(t) \in R^d$ denotes the memory state vector with dimension d, and A and B are state-space matrices. Following the use of Padé approximation [15], the state-space matrices can be expressed as follows:

$$\begin{aligned} \mathbf{A} &= [a]_{ij} \in \mathbb{R}^{d\times d}, \quad a_{ij} = (2i+1)\begin{cases} -1 & \text{if } i < j \\ (-1)^{i-j+1} & \text{if } i \geq j \end{cases} \\ \mathbf{B} &= [b]_i \in \mathbb{R}^{d\times 1}, \quad b_i = (2i+1)(-1)^i, \quad i,j \in [0, d-1] \end{aligned} \tag{2}$$

This continuous-time system can be converted to discrete-time k with a time resolution Δt:

$$\mathbf{m}[k] = \bar{\mathbf{A}}\mathbf{m}[k-1] + \bar{\mathbf{B}}u[k] \tag{3}$$

Here, $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are the discretized versions [1] of $\mathbf{A}$ and $\mathbf{B}$, which are usually frozen during training. As suggested by Chilkuri et al. [13], to enhance the efficiency of parallel training, we adopt the strategy of inputting the signal $x[k]$ into the LMU cell as follows:

$$u[k] = f_u(\mathbf{W_u}\mathbf{x}[k] + \mathbf{b_u}) \tag{5}$$

The output of this cell is described as:

$$\mathbf{o}[k] = f_o(\mathbf{W_x}\mathbf{x}[k] + \mathbf{W_m}\mathbf{m}[k] + \mathbf{b_o}) \tag{6}$$

where W_u, W_x, and W_m are trainable weights, b_o and b_u are the trainable bias, and f_u and f_o are activation functions. Since Eqs. (1-3) are linear and time-invariant (LTI), the state update equation can be expressed in a non-sequential fashion, which is crucial for enabling parallel training. Additionally, the fast Fourier transform (FFT) is employed to further reduce training complexity [13].

B. Spiking Neural Networks

Spiking Neural Networks (SNNs) represent the third generation of neural networks [16], and have garnered increasing attention due to their sparse activation patterns and the development of specialized hardware [17]. In contrast to traditional artificial neurons, spiking neurons accumulate input spike trains, which maintain the membrane potential, functioning as a form of temporal memory. When the membrane potential surpasses a certain threshold, the neuron emits a spike that is propagated to the subsequent layer, followed by a reset of the potential to a predefined baseline. This dynamic enables spiking neurons to model spatial-temporal relationships effectively [18], while also offering greater energy efficiency relative to artificial neurons [17]. Large spiking networks are typically trained with surrogate

[1] Using zero-order hold (ZOH) method, exact discretization gives

$$\bar{\mathbf{A}} = e^{\mathbf{A}\Delta t} \quad \text{and} \quad \bar{\mathbf{B}} = \mathbf{A}^{-1}(e^{\mathbf{A}\Delta t} - \mathbf{I})\mathbf{B}. \tag{4}$$

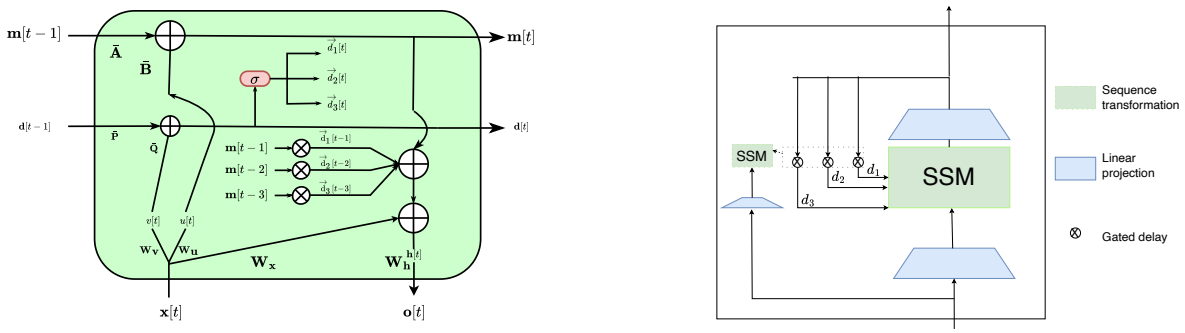


Fig. 1: Time-unrolled PDMU Cell (**left**) and simplified block diagram(**right**): A d-dimensional state vector ($h[t]$) is dynamically coupled with a d-dimensional memory vector ($m[t]$) and a delayed memory vector ($\vec{d}_{t-i}[i]m[t-i]$). At each step, the gate produces n coefficients (here n=3) that route the current memory to specific future offsets—i.e., learnable delays.

gradient descent [19], which has been shown to exploit spike timing information [20].

Historically, SNNs have predominantly been built upon recurrent neural network architectures [21]. Enhancements to these models have been realized through the incorporation of synaptic and axonal delays [22]–[25], heterogeneous time constant [26], as well as adaptive delays and thresholds [21], [27], which contribute to their flexibility and performance. Furthermore, spiking neurons are inherently capable of processing spike-based information directly. However, when addressing data that contains multi-bit information, it is necessary to introduce an encoding layer that converts these inputs into spike trains [28]. In this work, for tasks involving multi-bit inputs, we utilize an encoding layer that transforms these inputs into spike trains.

Recently, there has been a growing interest in using SNNs for sequential learning tasks. As more advanced architectures have emerged, researchers have begun exploring the integration of spiking neurons within structured state-space models. For example, Du et al. [29] have successfully combined these models with SNNs, demonstrating significant potential to enhance both performance and efficiency. Nevertheless, the optimal methods for incorporating spiking neurons into state-space models remain an open area of research.

III. Method

A. Parallel Delayed Memory Unit

The Parallel Delayed Memory Unit (PDMU), illustrated in Figure 1, is designed to facilitate parallel training while supporting sequential data processing during inference. Building upon the Legendre Memory Unit (LMU) framework, our model incorporates a delay line architecture equipped with a gating function that distributes current memory information to optimal future time points, enhancing the network's capacity to capture temporal dependencies.

The core component of PDMU is the delay gate, which controls the flow of memory information along the delay lines. To support parallel training, we employ a lightweight state-space model to achieve this:

$$\mathbf{d}'(t) = \mathbf{P}\mathbf{d}(t) + \mathbf{Q}v(t) \tag{7}$$

where $\mathbf{d}(t) \in R^n$ denotes the delay memory state vector with a length of n, and $\mathbf{P}$ and $\mathbf{Q}$ are state-space matrices similar to $\mathbf{A}$ and $\mathbf{B}$ except for their dimensions. Following the previous discretization, it can be expressed as:

$$\mathbf{d}[k] = \bar{\mathbf{P}}\mathbf{d}[k-1] + \bar{\mathbf{Q}}v[k] \tag{8}$$

where $v[k] = f_u(\mathbf{W_v}\mathbf{x}[k] + \mathbf{b_v})$, with $\mathbf{W_v}$ and $\mathbf{b_v}$ denoting the trainable weight matrix and bias vector, respectively. The output memory state $\mathbf{h}[k]$ is obtained not only from the long-term memory $\mathbf{m}[k]$ but also from the short-term delayed memory. This can be expressed as:

$$\mathbf{h}[k] = \mathbf{m}[k] + \sum_{i=k-n}^{k-1} \vec{d}_{k-i}[i]\,\mathbf{m}[i] \tag{9}$$

Here, we denote '$\rightarrow$' as the information flow direction for the future. Then, the output of the PDMU module can be described as:

$$\mathbf{o}[k] = f_o(\mathbf{W_h}\mathbf{h}[k] + \mathbf{W_x}\mathbf{x}[k] + \mathbf{b_o}) \tag{10}$$

For the parallel training, because this is an LTI system, standard control theory gives us a non-iterative way of evaluating this equation [13] as shown below:

$$\mathbf{m}[k] = \sum_{j=1}^{k} \bar{\mathbf{A}}^{k-j}\bar{\mathbf{B}}u_j \tag{11}$$

and

$$\mathbf{d}[k] = \sum_{j=1}^{k} \bar{\mathbf{P}}^{k-j}\bar{\mathbf{Q}}\bar{v}_j \tag{12}$$

Here, we define the delay gate as $\vec{d}_i[k] = \sigma_i(\mathbf{d}[k])$, where $i \in \{1, \ldots, n\}$, and $\sigma_i(\cdot)$ denotes its i-th output component (e.g., Softmax). For better training and visualization, the output hidden state of the PDMU module can alternatively be written as a matrix multiplication and summation. Assuming $\mathbf{H} = [H[1], H[2], \cdots, H[T]]$, each $H[k]$ is the result of the

element-wise multiplication and summation of the k^{th} columns of matrices $\hat{\mathbf{D}}$ and $\hat{\mathbf{M}}$.

$$\mathbf{H}[k] = \sum_{i=k-n}^{k} \hat{\mathbf{D}}_{i,k} \circ \hat{\mathbf{M}}_{i,k} \tag{13}$$

where

$$\hat{\mathbf{D}} = \begin{bmatrix} 1 & \overrightarrow{d}_1[1] & \overrightarrow{d}_2[1] & \cdots & \overrightarrow{d}_n[1] & \cdots & 0 & 0 \\ 0 & 1 & \overrightarrow{d}_1[2] & \cdots & \overrightarrow{d}_{n-1}[2] & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & \overrightarrow{d}_{n-2}[3] & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & \overrightarrow{d}_1[T-2] & \overrightarrow{d}_2[T-2] \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & \overrightarrow{d}_1[T-1] \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{14}$$

and

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{m}[1] & \mathbf{m}[1] & \mathbf{m}[1] & \cdots & \mathbf{m}[1] & \cdots & 0 & 0 \\ 0 & \mathbf{m}[2] & \mathbf{m}[2] & \cdots & \mathbf{m}[2] & \cdots & 0 & 0 \\ 0 & 0 & \mathbf{m}[3] & \cdots & \mathbf{m}[3] & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & \mathbf{m}[T-2] & \mathbf{m}[T-2] \\ 0 & 0 & 0 & \cdots & 0 & \cdots & \mathbf{m}[T-1] & \mathbf{m}[T-1] \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & \mathbf{m}[T] \end{bmatrix} \tag{15}$$

B. Bidirectional Delayed Memory Unit

Some non-causal sequence tasks require accessing both past and future input features at a given time. To achieve this, we can utilize a bidirectional PDMU network. This approach efficiently leverages past features (via forward states) and future features (via backward states) for a specific time frame. The key equation can be expressed as:

$$\mathbf{h}[k] = \mathbf{m}[k] + \sum_{i=k-n}^{k-1} \overrightarrow{d}_{k-i}[i]\, \mathbf{m}[i] + \sum_{i=k+1}^{k+n} \overleftarrow{d}_{i-k}[i]\, \mathbf{m}[i] \tag{16}$$

In this equation, the delay gates $\overrightarrow{d}_{k-i}[i]$ and $\overleftarrow{d}_{i-k}[i]$ capture dependencies from past and future memory states, respectively. A partial display of the corresponding $\hat{\mathbf{D}}$ matrix is illustrated in Figure 2.

C. Efficient Delayed Memory Unit

The Efficient PDMU constrains the PDMU to allow only one delay gate to be active at a time across all channels. This constraint is implemented through a mask operation, succinctly expressed as:

$$\mathbf{Mask}_i[k] = \begin{cases} 1, & \text{if } d_i[k] = \arg\max(\mathbf{d}[k]) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Then, the output hidden state can be described as:

$$\mathbf{h}[k] = \mathbf{m}[k] + \sum_{i=k-n}^{k-1} \overrightarrow{d}_{k-i}[i] \mathbf{Mask}_{k-i}[i] \mathbf{m}[i] \tag{18}$$

Due to the non-differentiable nature of the delay gradient, we employ the 'Straight-Through Estimator' (STE) technique [30], [31].This method addresses the challenge by facilitating optimization via gradient descent. The STE approach allows the forward pass to activate only a single gate while facilitating optimization via gradient descent. Thus, we apply the STE to the output of each layer as follows:

$$\frac{\partial \mathbf{Mask}_i[k]}{\partial d_i[k]} \approx 1$$

The training process for the PDMU and its variants, Bi-PDMU and EPDMU, is summarized in Algorithm 1.

D. Spiking Delayed Memory Unit

Spiking neural networks (SNNs) are energy-efficient owing to their discrete spike activations: a spike realises an accumulation operation, whereas silence incurs no communication. We employ the leaky integrate-and-fire (LIF) neuron [32]. The discrete form of the spiking DMU is

$$\mathbf{x}[k] = \Theta(\mathbf{h}[k] - \theta_u) \tag{19}$$

Here, $\mathbf{x}[k]$ is the spike vector and $\Theta(\cdot)$ denotes the Heaviside step function. Whenever the membrane potential exceeds the threshold θ_u, a spike is emitted and $\mathbf{h}[k]$ is reset to zero. For each branch (memory state and delay-gate state), the input spikes are mapped by the weight matrices W_u and W_v, respectively, and then passed through Θ to obtain scalar outputs. Thus, $u[k]$ and $v[k]$ are binary, taking values in $\{0, 1\}$.

$$\hat{\mathbf{D}} = \begin{bmatrix} 1 & \overrightarrow{d}_1[m] & \overrightarrow{d}_2[m] & \cdots & \overrightarrow{d}_{n-1}[m] & \overrightarrow{d}_n[m] \\ \overleftarrow{d}_1[m+1] & 1 & \overrightarrow{d}_1[m+1] & \cdots & \overrightarrow{d}_{n-2}[m+1] & \overrightarrow{d}_{n-1}[m+1] \\ \overleftarrow{d}_2[m+2] & \overleftarrow{d}_1[m+2] & 1 & \cdots & \overrightarrow{d}_{n-3}[m+2] & \overrightarrow{d}_{n-2}[m+2] \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overleftarrow{d}_{n-1}[m+n-1] & \overleftarrow{d}_{n-2}[m+n-1] & \overleftarrow{d}_{n-3}[m+n-1] & \cdots & 1 & \overrightarrow{d}_1[m+n-1] \\ \overleftarrow{d}_n[m+n] & \overleftarrow{d}_{n-1}[m+n] & \overleftarrow{d}_{n-2}[m+n] & \cdots & \overleftarrow{d}_1[m+n] & 1 \end{bmatrix}$$

Fig. 2: Partial display of the bidirectional delay gate matrix $\hat{D}$.

Algorithm 1 Training Procedure for PDMU and Its Variants

Input: Audio/EEG signals at time step k, $\mathbf{x}[k]$; class label for each task; model hyperparameters.
Output: Optimized Model parameters.

Initialize the Legendre memory unit state-space matrices **A** and **B**, and the delayed memory unit state-space matrices **P** and **Q** using the Padé approximation according to Equation 2.
Discretize the state-space matrices using the zero-order hold (ZOH) method.
Initialize weights and biases using Kaiming uniform initialization.
while the convergence criterion is not met **do**
    Input the signal into the RNN cell as described by Equation 5.
    Compute the memory cell $\mathbf{m}[t]$ as described in Equation 11.
    Compute the delay cell and delay gate using Equations 8 and 12.
    Compute the output hidden state:
    if using PDMU **then**
        Use Equation 9.
    else if using Bi-PDMU **then**
        Use Equation 16.
    else if using EPDMU **then**
        Use Equation 18.
    end if
    Compute the output signal $\mathbf{o}[k]$ using Equation 10.
end while

IV. Experimental Results

In our experiment, we endeavor to investigate three primary research questions (RQs):

RQ1 (Results): How does the PDMU model compare in terms of accuracy and effectiveness with other competitive models when processing biosignals that involve both long-term and short-term temporal dependencies?

RQ2 (Ablation study): To what extent does each component of the PDMU contribute to the overall improvements in performance and efficiency?

RQ3 (Training time and parameter increments): Does our model enhance training efficiency and achieve competitive results while requiring fewer parameter increments compared to other models?

To answer the above questions, we evaluate the performance of the proposed model on real-world temporal classification tasks, focusing primarily on cough audio classification, bio-signals (EEG and spiking audio datasets), and two benchmark tasks: speech commands and permuted sequential MNIST. We demonstrate the superiority of the proposed model by comparing it against other state-of-the-art baseline models with similar or fewer neurons.

A. Experimental Setups and Training Configuration

We selected benchmark tasks in bio-signal processing, spiking audio processing, and speech audio processing to evaluate our model. We implemented all models using the PyTorch library. Across all tasks, we set the nonlinear activation functions f_u and f_o to be $\mathrm{ReLU}(\cdot)$. We used the $\mathrm{softmax}(\cdot)$ function for σ to ensure a proper probability distribution. For all experiments, the delay line resolution and discretization resolution, denoted by Δt, was set to 1 time step.

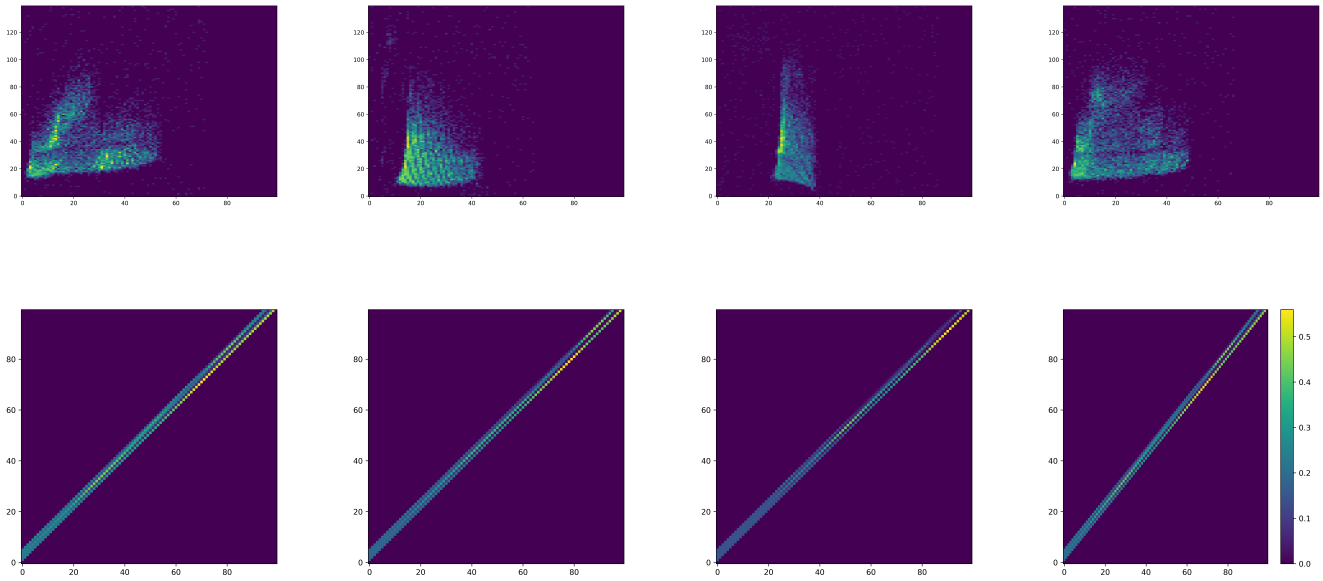
All models were trained using the Adam optimizer, with a batch size of 128 and a constant learning rate of 0.001. The Kaiming weight initialization approach was adopted for both network weights and biases. The state space matrices $\mathbf{A}, \mathbf{B}, \mathbf{P}, \mathbf{Q}$ were discretized using the Padé approximation and the ZOH method. The length of the delay line ranged from 2 to 10, depending on the task. All model training was conducted on Nvidia GeForce GTX 2080Ti GPUs, each equipped with 12 GB of memory.

B. Main results

To address RQ1, we compared PDMU with several state-of-the-art algorithms across a range of biosignals in healthcare, as well as on benchmarks designed to measure temporal dependencies, such as PS-MNIST.

1) Event-based Spoken Word Recognition: The Spiking Heidelberg Dataset (SHD) [33] was introduced for event-based speech processing research. This dataset is generated by converting raw audio waveforms into an event-based representation using a biologically plausible artificial cochlear model. In this study, we utilize the SHD dataset to evaluate the applicability and generalization of the PDMU across various data modalities. Each SHD sample consists of 700 channels, each representing distinct frequency-sensitive neurons in the peripheral auditory system. To ensure adequate temporal resolution and minimize post-processing effort, spikes are aggregated within 10 ms time bins, resulting in 100 time steps.

We employ a two-layer neural network, using the states (or membrane potential) at the last time step for decoding. We employ this decoding method to assess the network's long-term memory capabilities which follows the work by previous RNNs. In this task, delay-based models have demonstrated state-of-the-art performance. For example, models incorporating temporal attention and learnable axonal delays have achieved cutting-edge results on the SHD dataset [34]–[41]. Furthermore, models using parameter-free attention mechanisms [42] have reached accuracies up to 96.26%. However, these approaches typically rely on Spikerate decoding methods [43] and employ larger network architectures and introduce decision latency, which are not compatible with our current framework. Furthermore, as illustrated in Figure 3, we present the attention maps generated by the gating mechanism. When the input contains substantial information, as seen in panels (a) to (c), where the spike trains are dense, the gating function exhibits complex and dynamic interactions. In contrast, when the input carries minimal information, the gating mechanism functions analogously to a skip connection, assigning the highest attention weights to the preceding states [44]. This allows information from earlier time steps to be transmitted forward with minimal attenuation, effectively preserving historical signals.



(a) Voice '1' (b) Voice '2' (c) Voice '6' (d) Voice '9'

Fig. 3: Visualization of SHD Samples and Hidden Layer Attention Maps. Panels (a) voice '1', (b) voice '2', (c) voice '6', and (d) voice '9' are shown. The top row illustrates the input samples (x-axis: time steps; y-axis: channels), while the bottom row shows the corresponding gating maps at the hidden layer (both axes denote time steps).

TABLE I: Comparison of model performance on the SHD dataset. The architecture utilized two layers, each with 128 neurons. The Skip DMU* denotes a single constant-delay gate (skip connection in the temporal domain).

Methods	SNN	Accuracy
FSNN [33]	Yes	48.10%
RSNN [33]	Yes	83.20%
Adaptive RSNN [45]	Yes	84.40%
TC-LIF (Rec) [46]	Yes	88.91%
Spiking DMU	Yes	**90.98%**
LSTM [21]	No	79.90%
Skip DMU*	Yes	82.41%
PDMU	No	84.57%
EPDMU	No	**86.48%**
Bi-PDMU	No	84.94%

The results of our model, along with other state-of-the-art recurrent spiking neural networks (RSNNs) and LSTM models, are provided in Table I. Notably, for the spiking models, our proposed Spiking DMU model surpasses RSNN models by 7.78%. This result also competes favorably with the state-of-the-art RSNN model that employs temporal attention and advanced data augmentation techniques [47], as well as the two-component neuron models [46]. In addition, we evaluated an architecture equipped with a skip connection, which achieved an accuracy of 82.41%. Although this performance is an improvement over the baseline, it remains lower than that of the delay-line architecture, underscoring the importance of the internal complex interactions.

TABLE II: Comparison of model performance on COUGHVID dataset.

Methods	SNN	Accuracy	AUC	F1
LSTM [48]	No	76.41%	76.26%	74.48%
PDMU	No	82.45%	81.58%	82.21%
EPDMU	No	**82.45%**	82.08%	82.32%
Bi-PDMU	No	79.63%	78.02%	78.23%
Spiking-DMU	Yes	82.37%	81.88%	82.01%
CNN	No	81.37%	82.27%	80.97%
CNN-PDMU	No	**85.48%**	84.13%	85.10%

2) Cough Audio Signal Classification: The COUGHVID dataset, developed by Orlandic et al. [49], is a large-scale, publicly available corpus consisting of 27,550 cough recordings. Collected between April 1 and December 1, 2020, via a web application, participants provided metadata on age, gender, geolocation, pre-existing respiratory conditions, and COVID-19 status (Healthy, COVID-19, Symptomatic). The recordings, encoded in Opus at a 48kHz sampling rate, were annotated by four physician experts for cough quality, type, and other audible symptoms. The raw audio data was transformed into Mel spectrograms, each with 128 channels, and analyzed over simulation time steps of 420 (approximately 9.75 seconds).

A comparative analysis of models on the COUGHVID dataset (see Table II) demonstrates that CNN-PDMU achieved the accuracy of 85.48%. The pure CNN model (replacing the PDMU module with fully connected layers) only attained an accuracy of 81.37%, indicating the importance of capturing temporal dependencies in this dataset. Both PDMU and EPDMU performed well, each with an accuracy of 82.45%. The LSTM model achieved an accuracy of 76.41%, while Spiking-DMU

TABLE III: Comparison of model performance on the WithMe dataset. The Accuracy column shows performance results, with the former indicating within-subject performance and the latter showing unseen-subject performance. The architecture utilized two layers, each with 64 neurons.

Methods	SNN	Accuracy
EEGNet [53]	No	81.67%/76.42%
LSTM	No	80.09%/74.00%
DMU [51]	No	81.94%/75.92%
PDMU	No	84.17%/74.50%
EPDMU	No	84.43%/74.58%
Bi-PDMU	No	**84.82%/77.13%**
Spiking DMU	Yes	80.21%/75.08%

reached 82.37%, highlighting the potential of neuromorphic computing in cough classification. In addition, following the similar research [50], we also report the F1 and AUC metrics. The improvements persist under F1/AUC, indicating that the gains stem from better discriminative capacity rather than class imbalance. This study highlights the effectiveness of our model in analyzing cough sounds for diagnostic purposes.

3) EEG Attention Detection: The WithMe dataset, derived from the WithMe experiment [6], [51], [52], includes EEG recordings from 42 participants. For this study, we selected 38 participants for training and internal testing, dividing their data into training and testing sets, referred to as within-subjects. Data from the remaining 4 participants were used to evaluate our model's generalizability, termed unseen-subjects. Specifically, we partitioned the WithMe data into a training set and two testing sets: one for within-subject evaluations, consisting of 18,176 training instances and 4,580 validation instances, and another for unseen-subject assessments, with 2,400 validation instances. Preprocessing the EEG data involved re-referencing each channel to the average activity of the mastoid electrodes, band-pass filtering between 1 and 30 Hz, and downsampling to 64 Hz. The data were then segmented into 1.2-second epochs based on trigger events. The final preprocessing step normalized the EEG channel data to ensure zero mean and unit variance for each sample. The dataset is available for access at the following link [2].

As shown in Table III, our proposed model outperforms other baseline models. Compared to LSTM, it surpasses within-subject performance by 4.08% and unseen-subject performance by 0.5%. Furthermore, with the introduction of the bidirectional delay gate, it achieves the best accuracy of 77.13% on unseen subjects. For the spiking DMU, it achieves slightly better performance compared to the LSTM while requiring much fewer parameters and offering energy efficiency due to its sparse activations. These results underscore the effectiveness of the proposed method and its superior temporal modeling capability.

4) Speech Processing: The Speech Commands V2 dataset [54] consists of 105,829 audio files featuring 35 different words. Each recording lasts up to 1 second and is sampled at 16 kHz. The dataset is divided into 84,843 samples for training, 9,981 for validation, and 11,005 for testing. For our study, we selected the most challenging classification task involving all 35 classes to comprehensively assess our models' performance.

[2]https://github.com/sunpengfei1122/Withme-EEG-dataset

TABLE IV: Comparison of model performance on GSC dataset.

Methods	SNN	Accuracy
Rate-based SNN [57]	Yes	75.20%
SRNN+ALIF [21]	Yes	92.10%
SNN [58]	Yes	89.04%
SNN with SFA [58]	Yes	91.21%
LIF [59]	Yes	83.03%
RadLIF [59]	Yes	94.51%
TC-LIF [46]	Yes	94.84%
Spiking DMU	Yes	**96.39%**
RNN [59]	No	92.09%
liBRU [60]	No	95.06%
LMUFormer [61]	No	96.53%
PDMU	No	96.93%
EPDMU	No	**97.01%**
Bi-PDMU	No	96.96%

TABLE V: Comparison of model performance on PSMNIST dataset.

Methods	Architecture	SNN	Accuracy
GRU [62]	-	No	92.39%
LSTM [62]	200	No	89.86%
Lipschitz RNN [63]	200	No	96.40%
coRNN [64]	200	No	96.06%
DMU [2]	200	No	96.39%
LMU [12]	200+200	No	97.15%
PDMU	200+200	No	98.22%
EPDMU	200+200	No	98.24%
Bi-PDMU	200+200	No	**98.25%**
SRNN+ALIF [21]	64+256+256	Yes	94.30%
TC-LIF [46]	64+256+256	Yes	95.36%
Spiking DMU	200+200	Yes	**96.07%**

As reported in Table IV, our spiking DMU achieves the best results among spiking neural networks, with a noteworthy performance of 96.39%. For the non-spiking version, we compare our method with other models that can perform sequential inference, demonstrating competitive results. There is a $\sim$1% gap between our model and the transformer-based SOTA [55]. However, our model uses far fewer parameters (1.6M vs. 86.9M; $\approx 53\times$ fewer) and operates causally with strictly sequential inference. Relative to UniRepLKNet [56], the accuracy gap is $\sim$1.5% while the parameter count remains much smaller (1.6M vs. 55.5M; $\approx 34\times$ fewer).

5) Permuted Sequential Image Classification: The permuted sequential MNIST (PSMNIST) dataset [65] was developed to assess the capability of RNN models in capturing long-range temporal dependencies. This dataset is derived from the MNIST dataset by first flattening each image into a 784-dimensional vector and then shuffling its spatial information using a consistent permutation vector. In our experiments, the elements of this vector are presented sequentially to the network, with decisions made after processing all pixels. This task requires the model to reconstruct the original temporal

TABLE VI: Ablation study on different tasks including EEG classification (WithMe), audio processing (GSC and SHD), and PSMNIST. The GSC* represents the LMU cell augmented with LMUFormer [61].

Datasets	DMU	SNN	Parallel Training	Acc. (%)	Δ (%)
COUGHVID	No	No	Yes	80.23	0.00
	Yes	No	Yes	82.45	↑**2.22**
	No	Yes	No	76.52	0.00
	Yes	Yes	No	82.37	↑**5.85**
WithMe	No	No	Yes	82.68/69.83	0.00/0.00
	Yes	No	Yes	84.17/74.50	↑**1.49**/↑**4.67**
	No	Yes	No	79.24/66.83	0.00/0.00
	Yes	Yes	No	80.21/75.08	↑**0.97**/↑**8.25**
SHD	No	No	Yes	78.16	0.00
	Yes	No	Yes	84.57	↑**6.41**
	No	Yes	No	85.09	0.00
	Yes	Yes	No	90.98	↑**5.89**
PSMNIST	No	No	Yes	97.85	0.00
	Yes	No	Yes	98.22	↑**0.37**
	No	Yes	No	93.75	0.00
	Yes	Yes	No	96.07	↑**2.32**
GSC	No	No	Yes	74.44	0.00
	Yes	No	Yes	78.52	↑**4.08**
	Yes (Bi)	No	Yes	80.07	↑**5.63**
GSC*	No	No	Yes	96.53	0.00
	Yes	No	Yes	96.93	↑**0.40**
	No	Yes	No	95.71	0.00
	Yes	Yes	No	96.39	↑**0.68**

order and identify dependencies between different pixels.

Results in Table V show that our proposed PDMU competes with other RNN models engineered for learning long-term dependencies, such as Lipschitz RNN and coRNN [63], [64], [66]. Our model attains 98.22% with around 0.2M parameters, compared with 98.84% reported by 2M parameter CNN designed for long-range dependencies [67]. The 0.62% accuracy gap is modest relative to the 10 × reduction in model size. . Furthermore, the spiking DMU achieves a test accuracy of 96.07% using only two spiking layers with 200 neurons, establishing a new benchmark for this task.

C. Ablation study

To answer RQ2, we compare our approach with the baseline LMU module and calculate the performance improvement of each variant. The results, presented in Table VI, demonstrate that the PDMU model, when enhanced with our delayed memory unit, achieves significant improvements across all tasks. Specifically, in the non-spiking version, the causal delayed memory unit exhibits substantial gains in tasks requiring short-term memory, such as EEG classification and spiking spoken word detection. This suggests that while the LMU can capture very long-term memory, it may overlook detailed features present in bio-signals. Gains are largest on tasks with prominent short-term structure: on SHD, +6.41% (non-spiking) and +5.89% (spiking); on WithMe, +1.49% (within-subject) and +4.67% (unseen-subject) in the non-spiking setting, and +0.97% / +8.25% for the spiking setting. On GSC, PDMU yields +4.08%, while Bi-PDMU reaches +5.63%, highlighting the benefit of improved temporal credit assignment. When the baseline is already strong (PSMNIST and GSC*), improvements are smaller but consistent (typically +0.37% to +0.68%). Overall, these trends support our design goal: the delay-gated path complements the LMU's long-range memory by selectively aggregating recent states, yielding robust gains with minimal overhead.

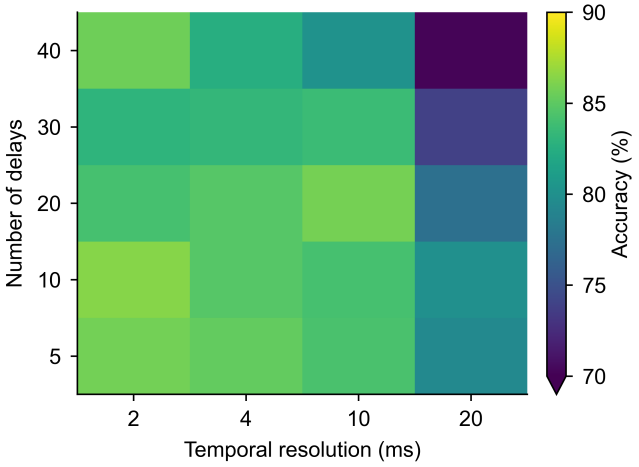


Fig. 4: Landscape showing classification accuracy as a function of temporal resolution (x-axis) and number of delays (y-axis) across SHD dataset.

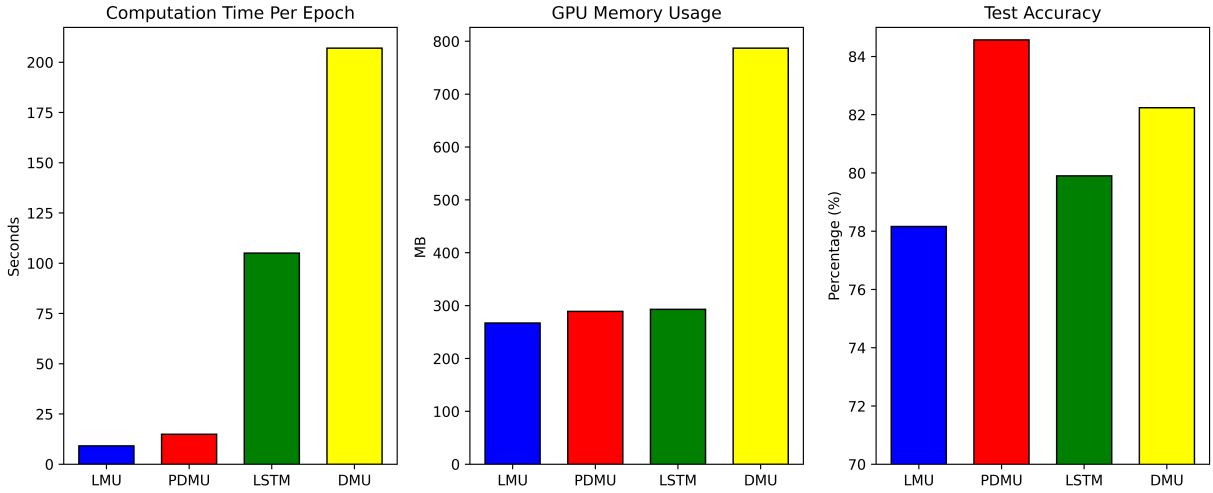


Fig. 5: Comparison of the computation time, GPU memory usage, and test accuracy of the proposed PDMU over other RNNs on the SHD dataset.

D. Training time and parameter increments

To answer RQ3, we first evaluated both the temporal resolution (time-step size) and the number of delays on SHD. Across settings (Figure 4), performance is stable, and a moderate delay budget (about 10 delays) consistently works well. Very long delay lines degrade accuracy by injecting stale context (e.g., at resolution $= 20$ with 50 simulation steps, using 40 delays reduces accuracy by 9%). In practice, the base SSM captures smoother, long-range context, while the delay path emphasizes short transients; excessive delays blur this balance. For this dataset, a total delay of less 100ms is a sensible target. Since increasing the number of delays also increases model size, we keep the delay count below 10 across all datasets (typically 5).

We evaluate the learning efficiency and GPU resource utilization of our proposed method compared to other recurrent neural network architectures, including the Legendre Memory Unit (LMU), Long Short-Term Memory (LSTM), and Delayed Memory Unit (DMU). Experiments were performed on a single Nvidia GeForce GTX 2080Ti GPU, utilizing a network configuration comprising two layers, each with 128 neurons, and a batch size of 64.

As illustrated in Figure 5, when processing the Spiking Heidelberg Digits (SHD) dataset, both the LSTM and DMU exhibit extended computation times due to their need to store and process intermediate neuronal states. In contrast, our model benefits from parallel training, achieving a 9 $\times$ and 29 $\times$ training speed up compared to the LSTM and DMU, respectively, while also delivering superior test performance. Regarding GPU memory consumption, the DMU shows the highest usage, primarily due to its requirement for a sliding memory mechanism to maintain delayed hidden states.

Regarding parameter efficiency, the PDMU is designed as a streamlined, state–space auxiliary module with learnable delay gates (typically fewer than 10). In the illustrative configuration where the delay line length is set to 5, the gate state–space uses two matrices $\mathbf{P} \in \mathbb{R}^{5\times5}$ and $\mathbf{Q} \in \mathbb{R}^{5\times1}$, contributing $25+5 = 30$ parameters. In addition, two small linear mappings handle the input signal and the delay gate; their contributions are modest (on the order of the input and delay dimensions).

At inference time, all three models are causal RNN variants and thus have the same single- step latency (real time). However, their per-sample compute differs because the intermediate activation and gate counts are different. We thus report (i) the total parameter count per layer and (ii) the amount of *runtime state* per layer that hardware must retain between steps as a proxy for compute comparison. Let M denote the input dimension, N the number of output neurons, and d the number of delays (typically $d < 10$). Then, for *inference parameter counts*:

$$\begin{aligned}
\text{LSTM:}\quad & \#\text{params} = 4\big(MN + N^2 + N\big),\\
\text{DMU:}\quad & \#\text{params} = MN + N^2 + N{+}Md{+}d^2{+}d,\\
\text{PDMU:}\quad & \#\text{params} = N^2 + M + N{+}2MN{+}d^2{+}d.
\end{aligned}$$

For *inference resources*, we have:

$$\begin{aligned}
\text{LSTM:}\quad & \#\text{runtime states} = 2N\\
\text{DMU:}\quad & \#\text{runtime states} = N \times d\\
\text{PDMU:}\quad & \#\text{runtime states} = N + d
\end{aligned}$$

Thus, PDMU adds only a small overhead relative to a standard State-Space Model (SSM), is smaller than LSTM (which carries the four-gate factor), and, unlike DMU, does not require an $N \times d$ buffer to store delayed hidden states, because PDMU compresses past information into a vector via the state–space module.

V. Conclusion

In this study, we introduced the Parallel Delayed Memory Unit (PDMU), an RNN variant designed to enhance the efficiency and effectiveness of audio and biomedical signal processing. The PDMU incorporates a delay line structure with delay gates to improve short-term temporal interaction and

credit assignment, supporting parallel training and sequential inference.

Our experiments demonstrate that the PDMU achieves competitive performance on benchmarks such as cough audio signal classification, EEG detection, spiking spoken word detection, speech processing, and permuted sequential image classification. Variants like bi-directional and efficient PDMUs further optimize performance and energy efficiency. The PDMU's ability to process complex biomedical and audio data efficiently has significant implications for medical diagnostics and explorations, facilitating advancements in health and medicine through improved signal analysis. In addition, the module is plug-and-play: the delay mechanism is, in principle, compatible with any SSM framework, and we plan to explore such integrations in future work.

REFERENCES

[1] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. Improving speech recognition by revising gated recurrent units. *arXiv preprint arXiv:1710.00641*, 2017.

[2] Pengfei Sun, Jibin Wu, Malu Zhang, Paul Devos, and Dick Botteldooren. Delayed memory unit: Modeling temporal dependency through delay gate. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[3] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection-a survey. *ACM Comput. Surv.*, 54(7):141–1, 2022.

[4] Kun Qian, Zhihao Bao, Zhonghao Zhao, Tomoya Koike, Fengquan Dong, Maximilian Schmitt, Qunxi Dong, Jian Shen, Weipeng Jiang, Yajuan Jiang, et al. Learning representations from heart sound: A comparative study on shallow and deep models. *Cyborg and Bionic Systems*, 5:0075, 2024.

[5] Evaldas Vaiciukynas, Antanas Verikas, Adas Gelzinis, and Marija Bacauskiene. Detecting parkinson's disease from sustained phonation and speech signals. *PloS one*, 12(10):e0185613, 2017.

[6] Jorg De Winne, Paul Devos, Marc Leman, and Dick Botteldooren. With no attention specifically directed to it, rhythmic sound does not automatically facilitate visual task performance. *Frontiers in Psychology*, 13:894366, 2022.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] 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.

[9] Guochen Yu, Andong Li, Hui Wang, Yutian Wang, Yuxuan Ke, and Chengshi Zheng. Dbt-net: Dual-branch federative magnitude and phase estimation with attention-in-attention transformer for monaural speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2629–2644, 2022.

[10] Yibin Zheng, Xinhui Li, Fenglong Xie, and Li Lu. Improving end-to-end speech synthesis with local recurrent neural network enhanced transformer. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6734–6738. IEEE, 2020.

[11] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[12] Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.

[13] Narsimha Reddy Chilkuri and Chris Eliasmith. Parallelizing legendre memory unit training. In *International Conference on Machine Learning*, pages 1898–1907. PMLR, 2021.

[14] Linhui Sun, Shuo Yuan, Aifei Gong, Lei Ye, and Eng Siong Chng. Dual-branch modeling based on state-space model for speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1457–1467, 2024.

[15] Henri Padé. Sur la représentation approchée d'une fonction par des fractions rationnelles. In *Annales scientifiques de l'Ecole normale supérieure*, volume 9, pages 3–93, 1892.

[16] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[17] Sumit Bam Shrestha, Jonathan Timcheck, Paxon Frady, Leobardo Campos-Macias, and Mike Davies. Efficient video and audio processing with loihi 2. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13481–13485. IEEE, 2024.

[18] Yuchen Wang, Kexin Shi, Chengzhuo Lu, Yuguo Liu, Malu Zhang, and Hong Qu. Spatial-temporal self-attention for asynchronous spiking neural networks. In *IJCAI*, pages 3085–3093, 2023.

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

[20] Ziqiao Yu, Pengfei Sun, and Dan FM Goodman. Beyond rate coding: Surrogate gradients enable spike timing learning in spiking neural networks. *arXiv preprint arXiv:2507.16043*, 2025.

[21] 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.

[22] Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

[23] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, and Haizhou Li. Supervised learning in spiking neural networks with synaptic delay-weight plasticity. *Neurocomputing*, 409:103–118, 2020.

[24] Pengfei Sun, Longwei Zhu, and Dick Botteldooren. Axonal delay as a short-term memory for feed forward deep spiking neural networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8932–8936. IEEE, 2022.

[25] Pengfei Sun, Jascha Achterberg, Zhe Su, Dan F. M. Goodman, and Danyal Akarca. Exploiting heterogeneous delays for efficient computation in low-bit neural networks, 2025.

[26] Nicolas Perez-Nieves, Vincent CH Leung, Pier Luigi Dragotti, and Dan FM Goodman. Neural heterogeneity promotes robust learning. *Nature communications*, 12(1):5791, 2021.

[27] Pengfei Sun, Ehsan Eqlimi, Yansong Chua, Paul Devos, and Dick Botteldooren. Adaptive axonal delays in feedforward spiking neural networks for accurate spoken word recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[28] Jibin Wu, Yansong Chua, Malu Zhang, Guoqi Li, Haizhou Li, and Kay Chen Tan. A tandem learning rule for effective training and rapid inference of deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):446–460, 2021.

[29] Yu Du, Xu Liu, and Yansong Chua. Spiking structured state space model for monaural speech enhancement. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 766–770. IEEE, 2024.

[30] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[31] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[32] Larry F Abbott and Thomas B Kepler. Model neurons: from hodgkin-huxley to hopfield. In *Statistical Mechanics of Neural Networks: Proceedings of the XIth Sitges Conference Sitges, Barcelona, Spain, 3–7 June 1990*, pages 5–18. Springer, 2005.

[33] 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*, 2020.

[34] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10221–10230, 2021.

[35] Pengfei Sun, Yansong Chua, Paul Devos, and Dick Botteldooren. Learnable axonal delay in spiking neural networks improves spoken word recognition. *Frontiers in Neuroscience*, 17:1275944, 2023.

[36] Ilyass Hammouamri, Ismail Khalfaoui-Hassani, and Timothée Masquelier. Learning delays in spiking neural networks using dilated convolutions with learnable spacings. In *The Twelfth International Conference on Learning Representations*.

[37] Simone D'agostino, Filippo Moro, Tristan Torchet, Yiğit Demirağ, Laurent Grenouillet, Niccolò Castellani, Giacomo Indiveri, Elisa Vianello,

and Melika Payvand. Denram: neuromorphic dendritic architecture with rram for efficient temporal processing with delays. *Nature communications*, 15(1):3446, 2024.

[38] Alberto Patino-Saucedo, Roy Meijer, Amirreza Yousefzadeh, Manil-Dev Gomony, Federico Corradi, Paul Detteter, Laura Garrido-Regife, Bernabe Linares-Barranco, and Manolis Sifalakis. Hardware-aware training of models with synaptic delays for digital event-driven neuromorphic processors. *arXiv preprint arXiv:2404.10597*, 2024.

[39] Prajna G Malettira, Shubham Negi, Wachirawit Ponghiran, and Kaushik Roy. Tskips: Efficiency through explicit temporal delay connections in spiking neural networks. *arXiv preprint arXiv:2411.16711*, 2024.

[40] Balázs Mészáros, James C Knight, and Thomas Nowotny. Efficient event-based delay learning in spiking neural networks. *arXiv preprint arXiv:2501.07331*, 2025.

[41] Alexandre Queant, Ulysse Rançon, Benoit R Cottereau, and Timothée Masquelier. Delrec: learning delays in recurrent spiking neural networks. *arXiv preprint arXiv:2509.24852*, 2025.

[42] Pengfei Sun, Jibin Wu, Paul Devos, and Dick Botteldooren. Towards parameter-free attentional spiking neural networks. *Neural Networks*, 185:107154, 2025.

[43] Sumit Bam Shrestha, Longwei Zhu, and Pengfei Sun. Spikemax: spike-based loss methods for classification. In *2022 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2022.

[44] Jack Cook, Danyal Akarca, Rui Ponte Costa, and Jascha Achterberg. Brain-like processing pathways form in models with heterogeneous experts. *arXiv preprint arXiv:2506.02813*, 2025.

[45] Bojian Yin, Federico Corradi, and Sander M Bohté. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, pages 1–8, 2020.

[46] 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, pages 16838–16847, 2024.

[47] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021.

[48] Skander Hamdi, Mourad Oussalah, Abdelouahab Moussaoui, and Mohamed Saidi. Attention-based hybrid cnn-lstm and spectral data augmentation for covid-19 diagnosis from cough sound. *Journal of Intelligent Information Systems*, 59(2):367–389, 2022.

[49] Lara Orlandic, Tomas Teijeiro, and David Atienza. The coughvid crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms. *Scientific Data*, 8(1):156, 2021.

[50] Mesut Melek. Diagnosis of covid-19 and non-covid-19 patients by classifying only a single cough sound. *Neural Computing and Applications*, 33(24):17621–17632, 2021.

[51] Pengfei Sun, Jorg De Winne, Paul Devos, and Dick Botteldooren. Eeg decoding with conditional identification information. *Advances in Signal Processing and Artificial Intelligence*, page 130, 2024.

[52] Pengfei Sun, Jorg De Winne, Malu Zhang, Paul Devos, and Dick Botteldooren. Electroencephalography decoding with conditional identification generator. *International Journal of Neural Systems*, 35(07), 2025.

[53] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[54] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[55] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[56] Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. Unireplknet: A universal perception large-kernel convnet for audio video point cloud time-series and image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5513–5524, 2024.

[57] Emre Yılmaz, Ozgür Bora Gevrek, Jibin Wu, Yuxiang Chen, Xuanbo Meng, and Haizhou Li. Deep convolutional spiking neural networks for keyword spotting. In *Proceedings of INTERSPEECH*, pages 2557–2561, 2020.

[58] Darjan Salaj, Anand Subramoney, Ceca Kraisnikovic, Guillaume Bellec, Robert Legenstein, and Wolfgang Maass. Spike frequency adaptation supports network computations on temporally dispersed information. *Elife*, 10:e65459, 2021.

[59] Alexandre Bittar and Philip N Garner. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16:865897, 2022.

[60] Douglas Coimbra De Andrade, Sabato Leo, Martin Loesener Da Silva Viana, and Christoph Bernkopf. A neural attention model for speech command recognition. *arXiv preprint arXiv:1808.08929*, 2018.

[61] Zeyu Liu, Gourav Datta, Anni Li, and Peter Anthony Beerel. Lmuformer: Low complexity yet powerful spiking model with legendre memory units. In *The Twelfth International Conference on Learning Representations*.

[62] Sarath Chandar, Chinnadhurai Sankar, Eugene Vorontsov, Samira Ebrahimi Kahou, and Yoshua Bengio. Towards non-saturating recurrent units for modelling long-term dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3280–3287, 2019.

[63] N Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2020.

[64] T Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2020.

[65] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[66] T Konstantin Rusch, Siddhartha Mishra, N Benjamin Erichson, and Michael W Mahoney. Long expressive memory for sequence modeling. *arXiv preprint arXiv:2110.04744*, 2021.

[67] David W Romero, David M Knigge, Albert Gu, Erik J Bekkers, Efstratios Gavves, Jakub M Tomczak, and Mark Hoogendoorn. Towards a general purpose cnn for long range dependencies in n d. *arXiv preprint arXiv:2206.03398*, 2022.