



A biologically plausible recurrent spiking architecture for modeling nonlinear dynamic mechanical systems

Vaishnav Bhaskaran^a, Saurabh Balkrishna Tandale^a, Vasileios Polydoros^a,
 Agnes Korcsak-Gorzo^b, Abigail Morrison^b, Markus Diesmann^b,
 Marcus Stoffel^{a,*}

^a Institute of General Mechanics, RWTH Aachen University, Eilfschornsteinstraße 18, 52062 Aachen, Germany

^b Institute for Advanced Simulation (IAS-6), Computational and Systems Neuroscience, Forschungszentrum Jülich, Germany

ARTICLE INFO

Keywords:

Spiking Neural Networks
 e-prop
 Structural Mechanics
 Rheological models
 Brain-inspired learning

ABSTRACT

Spiking Neural Networks (SNNs) have gained significant attention in recent years because their spike-based, event-driven communication, together with the inherent sparsity of spikes, is believed to enable highly energy-efficient computation. This property has been effectively leveraged in a variety of classification and regression tasks. Although inspired by biological neural systems, most existing SNNs bear only limited resemblance to their biological counterparts in terms of dynamics and learning mechanisms. Several studies have aimed to better understand the mathematical functioning of the brain to incorporate greater biological plausibility into existing neuron models. One such approach is to train the networks using a more biologically plausible learning rule called eligibility propagation (e-prop), which serves as an alternative to conventional Back Propagation Through Time (BPTT). In this study, we train a recurrent-SNN composed of Adaptive-threshold based Leaky Integrate-and-Fire (ALIF) neurons using the e-prop learning rule on datasets derived from viscoelastic rheological models. We establish an analogy between these rheological models and the membrane dynamics of LIF neurons, highlighting their mathematical similarity and providing a physical justification for employing SNNs in mechanics beyond energy-efficiency considerations. To model geometrically and physically nonlinear structural deformations, we further propose a multi-layer recurrent spiking architecture that integrates Legendre Memory Units with ALIF neurons for predicting nonlinear transient displacement responses of plates obtained from shock tube experiments. To train this architecture, a hybrid learning approach is introduced by combining e-prop and BPTT. Finally, a comparative analysis between e-prop and BPTT is conducted, emphasizing their respective trade-offs in accuracy, convergence speed, and computational efficiency.

1. Introduction

One of the most complex systems in the universe is the human brain, comprising approximately 86 billion neurons [1] interconnected by 100–500 trillion synapses [2]. The brain performs advanced cognitive and perceptual tasks such as memorization, sensory processing, and abstract reasoning while consuming only 20 W of power [3,4]. One of the earliest attempts to emulate biological neural networks was the perceptron [5]. The introduction of backpropagation [6] enabled Artificial Neural Networks (ANNs) to perform complex pattern recognition tasks, and their success has since grown exponentially with the advent of increased

* Corresponding author.

E-mail address: stoffel@iam.rwth-aachen.de (M. Stoffel).

<https://doi.org/10.1016/j.cma.2026.119081>

Received 11 December 2025; Received in revised form 14 April 2026; Accepted 12 May 2026

Available online 19 May 2026

0045-7825/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

computational power and data accessibility. Architectural variants such as convolutional (CNN) and recurrent (RNN) neural networks further enhanced their representational capabilities [7–10].

With artificial intelligence (AI) models achieving significant breakthroughs in image recognition [11,12], reasoning [13,14] and mathematics [15,16], their adoption has expanded into computational science domains such as structural mechanics [17,18], fluid dynamics [19,20], biomechanics [21–23], and material parameter identification [24–26]. In solid mechanics, neural models have been employed to improve or replace material laws [27–31], map stress–strain relationships directly [32–35] assist Finite Element Methods (FEM) by learning internal forces and stiffness matrices [36–39]. Automated material discovery frameworks have also been developed for interpretable constitutive law identification using only force–displacement data [40–43].

However, the increasing computational and data demands of large AI models [44,45] raise sustainability concerns. For instance, recent measurements show that inference with GPT-4o consumes on average about 1.2 Wh per query and a single day of global usage can exceed 400 MWh of energy [46]. Such power-hungry models are unsuitable for low-power or edge devices without cloud support. This is in stark contrast to the human brain, which achieves high computational capacity with minimal energy. Although modern AI models are inspired by biological principles, they differ significantly from the brain's continuous, feedback-driven, and dynamic processing [47–51].

A step towards more biologically grounded computation is offered by SNNs [52,53]. In these networks, neurons emit discrete spikes when their membrane potential exceeds a threshold, leading to sparse and event-driven computation. This sparsity is the key to their low-power operation, allowing SNNs to achieve comparable performance to ANNs with substantially reduced energy consumption [54–56]. They have also demonstrated strong performance in image-based tasks [57–59] and nonlinear regression [60,61].

Recently, several studies have applied SNNs to problems in mechanics, highlighting their potential for energy-efficient and physically interpretable modeling of dynamic systems. In [62], spiking recurrent networks integrating Legendre Memory Units (LMUs) [63] with dense layers were developed as neuromorphic solvers for boundary value problems, achieving up to 1000× improvements in energy-efficiency compared to conventional ANNs. In [64], an SNN-based nonlinear regression framework was proposed to predict structural responses exhibiting material plasticity, strain-rate sensitivity and dynamic deformation using neuromorphic hardware for real-time processing. [65] presented a hybrid mechanics-informed neural network approach incorporating bio-inspired spiking layers to enforce mechanical consistency constraints in structural analysis. Similarly, [27] introduced a physics-based self-learning SNN coupling spike-driven computation with governing equations for adaptive identification of time-dependent mechanical states. Extending this line of research, [66] introduced a physics-based SNN surrogate model for nonlinear viscoplastic behavior in impulsively loaded beams, replacing constitutive material laws in a finite element solver with SNN and PINN surrogates, demonstrating superior computational efficiency and physical consistency. Broader reviews, such as [67], emphasized the promise of SNNs for real-time mechanical dynamics and vibration analysis, while [68] proposed mechanical neural networks, a neuromorphic-inspired approach in which explicit mechanical neurons compute deformation-based responses.

Recurrent Neural Networks (RNNs) are well suited for modeling temporal dependencies, and experimental evidence confirms that biological networks also exhibit recurrent connectivity [49]. Classical gated architectures such as Long Short-Term Memory (LSTM) networks [69] improve the learning of long-range temporal structure through carefully designed gating mechanisms, but they are computationally expensive and difficult to deploy on low-power hardware. More recently, LMUs were introduced as theoretically grounded state–space models that encode long-timescale information with optimal memory compression. LMUs achieve high accuracy while drastically reducing the number of trainable parameters compared to LSTMs, making them attractive for real-time dynamical modeling.

These advances align naturally with SNNs, where recurrence enables richer temporal processing [70–72]. Incorporating recurrent structure into SNNs has yielded substantial improvements in representational capacity, giving rise to Recurrent Spiking Neural Networks (RSNNs). Several recent studies [62,66] have used RSNNs to address mechanical and structural dynamics problems. However, these networks have primarily been trained using BPTT [73], which unrolls the recurrent network into a feedforward sequence [74] and computes gradients through time. Because spike events are discontinuous, surrogate derivatives are typically used to approximate gradients in RSNNs to solve computationally demanding tasks [75,76]. While effective, BPTT is biologically implausible and computationally expensive, as it requires storing all neuron states across time steps—rendering it unsuitable for real-time or neuromorphic applications.

To overcome these limitations, [77] introduced e-prop, a biologically plausible, local, and online learning rule that approximates BPTT using neuron-level eligibility traces and broadcast error signals. These eligibility traces act as transient memory that links recent activity to delayed feedback. Unlike BPTT, e-prop decomposes the global gradient into a learning signal and eligibility trace, computed locally for each synapse. This formulation enables compatibility with diverse neuron models and allows continuous, online updates without storing historical states. The learning signal, modulated through a random feedback matrix, mimics neuromodulatory processes in biological systems. Although this form of feedback alignment performs comparably to backpropagation in feedforward networks [78,79] and multi-layer SNNs [80,81], it remains less effective for deep architectures for image classification tasks [82]. In [83], e-prop was formalized as part of a family of causal and local approximation algorithms for recurrent networks, while [84–86] demonstrated e-prop-based learning on neuromorphic hardware.

Despite these advances, there remains very limited work applying e-prop to train multi-layer RSNNs for nonlinear regression, particularly in the context of dynamic mechanical systems. To address this gap, the present study proposes a biologically plausible recurrent spiking framework for modeling nonlinear and transient mechanical responses. The main contributions of this work are summarized as follows:

- Establishing an analogy between viscoelastic rheological models and the membrane dynamics of LIF neurons, thereby providing a conceptual bridge that motivates the use of spiking neural networks in mechanics beyond their energy-efficiency advantages.
- Developing and training a two-layer RSNN with ALIF neurons on a viscoelastic dataset using both BPTT and the biologically plausible e-prop learning rules.
- Proposing a multi-layer recurrent spiking model integrating LMUs with ALIF neurons to predict geometrically and physically nonlinear viscoplastic structural deformations.
- Introducing a hybrid e-prop/BPTT training approach for the LMU-RSNN architecture and evaluating its performance against fully BPTT-based and non-spiking models.
- Performing a comparative analysis of the different learning paradigms in terms of accuracy, convergence speed, and computational efficiency, highlighting the potential of biologically inspired learning for modeling dynamic mechanical systems.

Incorporating biologically plausible learning principles into recurrent spiking architectures represents a significant step toward sustainable, brain-inspired computing for modeling dynamic mechanical systems—a research direction that remains largely unexplored.

The remainder of this article is organized as follows. Section 2 introduces spiking neural networks and presents the formulations of the ALIF neuron model, the LMU cell and its spiking variant, as well as the learning rules considered in this study. Section 3 describes the fundamentals of the viscoelastic rheological model and the experimental setup of the shock tube. Section 4 details the recurrent spiking architectures developed in this work. Section 5 provides a comparative analysis of the learning rules in terms of accuracy, convergence speed, and energy efficiency, and compares the performance of the spiking networks with their non-spiking counterparts. Finally, Section 6 summarizes the findings and outlines the study's broader implications.

2. Spiking neural networks

SNNs, often regarded as the third generation of ANNs, are designed to mimic the operation of biological neural systems, where information is transmitted through discrete spike events [52,53]. Unlike traditional ANNs, this spike-based communication naturally endows SNNs with an intrinsic temporal dimension. This temporal characteristic is most evident in the underlying neuron models, which form the fundamental building blocks of such networks. The mathematical formulation of these recurrent neuron models and the e-prop learning rule are briefly explained below.

2.1. ALIF model

The LIF model, first introduced by Lapicque [87], is one of the most established and widely used neuron models. Despite its reduced biological complexity, it effectively captures the essential membrane potential dynamics with sufficient accuracy for many applications [27,59,60], while its mathematical tractability and low computational cost facilitate efficient software and hardware implementations.

In its original form, the LIF model is a current-based point neuron model that describes how the membrane potential evolves as a function of the input current. The dynamics are governed by the following differential equation:

$$\tau_m \frac{dv(t)}{dt} = -v(t) + R I(t), \quad (1)$$

where $\tau_m = RC_m$ is the membrane time constant, $v(t)$ is the membrane potential, R is the membrane resistance, C_m is the membrane capacitance, and $I(t)$ is the input current. The neuron emits a spike when $v(t)$ reaches a fixed threshold v_{th} , after which the potential is reset. The temporal evolution of the LIF neuron under piecewise constant input current is illustrated in Fig. 1.

Though the LIF model provides a useful abstraction of neuronal dynamics, it lacks important biological mechanisms such as adaptation, where the firing threshold depends on the recent spiking activity. The ALIF model extends the LIF formulation by introducing a dynamic threshold that increases transiently after each spike and decays back to a baseline value over time. This extension equips the neuron with intrinsic temporal memory, enhancing its temporal processing capabilities while retaining the efficiency of the LIF model.

The ALIF neuron j is characterized by a two-dimensional hidden state variable $\mathbf{h}_j^t$ and a binary observable state variable $z_j^t \in \{0, 1\}$. The hidden state variable consists of the membrane potential v_j^t and the threshold adaptation a_j^t . The following equations define the dynamics of the ALIF neuron j in a single layer:

$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{rec} z_i^t + \sum_i W_{ji}^{in} x_i^{t+1} - z_j^t v_{th}. \quad (2)$$

Here, α is the decay factor given by $e^{-\delta t/\tau_m}$, where τ_m is the membrane time constant. W_{ji}^{rec} and W_{ji}^{in} denote the synaptic weights from network neuron i and input neuron i to neuron j , respectively. x_i^{t+1} is the input of the current time step and z_i^t is the observable state at the previous time step. The term $z_j^t v_{th}$ resets the membrane potential to the baseline threshold v_{th} after each spike. The observable state is computed with the Heaviside function H as:

$$z_j^t = H(v_j^t - A_j^t), \quad (3)$$

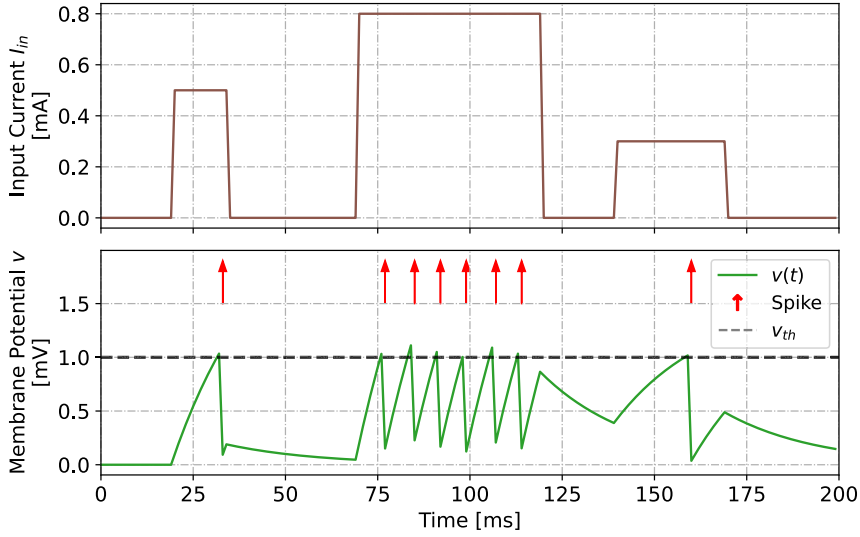


Fig. 1. The dynamics of a leaky integrate-and-fire neuron subjected to piecewise constant input current.

where A_j^t is the threshold potential which increases after each spike and decreases exponentially back to the baseline threshold v_{th} . This evolution of the threshold potential is described by

$$A_j^t = v_{th} + \beta a_j^t, \quad (4)$$

$$a_j^{t+1} = \rho a_j^t + z_j^t, \quad (5)$$

where β is the adaptation strength, a_j^t is the threshold adaptation. ρ is the adaptation decay factor given by $e^{-\delta t/\tau_a}$, with τ_a denoting the adaptation time constant. Eq. (2) can be modified in the following way for multiple layers:

$$v_j^{t+1,l} = \alpha v_j^{t,l} + \sum_{i \neq j} W_{ji}^{rec,l} z_i^{t,l} + \sum_i W_{ji}^{in,l} I - z_j^{t,l} v_{th}, \quad (6)$$

$$I = \begin{cases} x_i^{t+1} & l = 1 \\ z_i^{t+1,l-1} & \text{otherwise.} \end{cases} \quad (7)$$

2.2. Legendre memory unit

Although the ALIF neuron exhibits memory through its adaptive threshold dynamics, this mechanism alone is insufficient to capture complex transient behaviors observed in real-world systems. To address this limitation, LMUs [63] are used in this study. The key aspect of LMUs is the ability to learn complex functions of longer duration with less internal state variables. With the same network size, they outperform Long Short-Term Memory cells (LSTMs) [69], which are very successful in learning complex temporal relationships.

LMUs are derived by constructing a finite-dimensional approximation of the delay operator with Laplace transform $F(s) = e^{-\theta s}$, which corresponds to storing the recent history of a signal over a sliding window of length $\theta \in \mathbb{R}_{>0}$. Instead of retaining the full delayed signal $u(t - \tau)$, $\tau \in [0, \theta]$, the LMU compresses this history into a low-dimensional state vector by projecting it onto the first d Legendre polynomials. This derivation leads to a continuous-time linear dynamical system of the form:

$$\theta \dot{\mathbf{m}}(t) = \mathbf{A} \mathbf{m}(t) + \mathbf{B} u(t), \quad (8)$$

where the matrices $\mathbf{A}$ and $\mathbf{B}$ are ideal state-space matrices, meaning they arise from the closed-form analytical solution of the Legendre delay approximation prior to discretization or learning. Their entries are given by:

$$\mathbf{A} = [a]_{ij} \in \mathbb{R}^{d \times d}, a_{ij} = (2i + 1) \begin{cases} -1 & i < j \\ (-1)^{i-j+1} & i \geq j \end{cases}, \quad (9)$$

$$\mathbf{B} = [b]_i \in \mathbb{R}^{d \times 1}, b_i = (2i + 1)(-1)^i, \quad i, j \in [0, d - 1]. \quad (10)$$

For a recurrent layer, Eq. (8) can be discretized at any time step $t \in \mathbb{N}$ in the following way:

$$\mathbf{m}^t = \bar{\mathbf{A}} \mathbf{m}^{t-1} + \bar{\mathbf{B}} u^t. \quad (11)$$

$\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are discretized using the zero-order hold method, which assumes that the input remains constant within each sampling interval. The input signal $u(t)$ is computed based on the input vector $\mathbf{x}^t$, the hidden state variable of the previous time step $\mathbf{h}^{t-1}$

and the memory of the previous time step $\mathbf{m}^{t-1}$, which are unique to each recurrent layer. The hidden state can be represented by any spiking or non-spiking recurrent cell. In this work, the hidden state is implemented using an ALIF neuron, as described in Section 2.1, for the spiking version of the LMU. Accordingly, the hidden state variable $\mathbf{h}^t$ corresponds to the binary spike output of the ALIF neurons. To couple the LMU with the spiking dynamics, the input signal u^t is constructed from the input vector $\mathbf{x}^t$, the previous spike output $\mathbf{h}^{t-1}$, and the previous memory state $\mathbf{m}^{t-1}$ as:

$$u^t = \mathbf{e}_x \mathbf{x}^t + \mathbf{e}_h \mathbf{h}^{t-1} + \mathbf{e}_m \mathbf{m}^{t-1}, \quad (12)$$

where $\mathbf{e}_x, \mathbf{e}_h, \mathbf{e}_m$ are learnable encoding vectors, which project relevant information into the memory at every time step. Thus, the LMU receives the spike-based representation of the network activity, allowing the memory unit to capture a compressed history of the spike trains over a finite time window. The computed memory state $\mathbf{m}^t$ is then fed back to the spiking neurons as an additional input current:

$$\mathbf{h}^t = f(\mathbf{W}_x \mathbf{x}^t + \mathbf{W}_h \mathbf{h}^{t-1} + \mathbf{W}_m \mathbf{m}^t). \quad (13)$$

where $\mathbf{W}_x, \mathbf{W}_h$ and $\mathbf{W}_m$ are trainable kernels and $f(\cdot)$ denotes the nonlinear dynamics of the ALIF neuron.

2.3. Learning rule

Training RNNs in machine learning is performed by unrolling the network into a virtual feedforward network where each layer represents a time step. By applying backpropagation to every layer, the gradients are propagated backwards in time and thus updating the weights across all time steps. Hence this method is called backpropagation through time. The objective of training a network is to find the optimal synaptic weights W_{ji} by minimizing the loss function E . For a neuron j in a recurrent layer l , the general equation to compute the gradients across via BPTT can be expressed as:

$$\frac{dE}{dW_{ji}} = \sum_{t=1}^T \sum_{l=1}^L \frac{dE}{dh_j^{t,l}} \cdot \frac{\partial h_j^{t,l}}{\partial W_{ji}^l}, \quad (14)$$

$$\frac{dE}{dh_j^{t,l}} = \frac{\partial E}{\partial z_j^{t,l}} \cdot \frac{\partial z_j^{t,l}}{\partial h_j^{t,l}} + \sum_{k \neq j}^N \frac{dE}{dh_k^{t+1,l}} \cdot \frac{\partial h_k^{t+1,l}}{\partial h_j^{t,l}} + \sum_{k \neq j}^N \frac{dE}{dh_k^{t,l+1}} \cdot \frac{\partial h_k^{t,l+1}}{\partial h_j^{t,l}}. \quad (15)$$

Here, Eqs. (14) and (15) give the overview of how the gradients are summed over for all layers L across all time steps T . The term $\frac{dE}{dh_j^{t,l}}$ consists of the direct loss contribution, the recurrent contribution from neurons $k \neq j$ in the same layer at the next time step, and the top-down influence from neurons $k \neq j$ in the subsequent layer, where k indexes the neurons of the respective layer. The limitations of BPTT arise mainly from two properties: non-locality and non-causality.

- *Non-locality*: Due to the explicit references in Eq. (15), the gradient of a synapse W_{ji} depends not only on the postsynaptic neuron j but also on the future states of all other neurons k , where $k \neq j$. This dependency propagates forward in time, such that the weight update for W_{ji} required access to hidden variables from all neurons across all subsequent time steps. On the contrary, a local algorithm would only require information that is available in the immediate neighborhood of the synapse.
- *Non-causality*: Similarly, BPTT suffers from non-causality because the computation of gradients at time t requires information from all future time steps. This causes two primary issues: (i) update locking, since weight updates can be computed only after the entire sequence has been processed, and (ii) backward locking, where gradients must be propagated backwards through all time steps before the current gradient can be determined. Also, the entire trajectory of all hidden state variables across all time steps T should be stored during inference, resulting in a memory complexity of order $\mathcal{O}(nT)$ for n neurons. While checkpointing strategies can reduce this memory requirement by trading off additional forward computation, the memory still scales with the sequence length and remains a limiting factor for long sequences. A causal algorithm, in contrast, would only rely on past and present activity, enabling online computation of gradients during inference.

To address these limitations, [77] introduced e-prop, a biologically inspired learning rule that replaces the non-local and non-causal computations with local and causal eligibility traces and learning signals. The mathematical basis for e-prop starts from the gradient computation of a single recurrent layer by BPTT, which can be obtained from Eq. (14) as:

$$\frac{dE}{dW_{ji}} = \sum_{t'} \frac{dE}{dh_j^{t'}} \cdot \frac{\partial h_j^{t'}}{\partial W_{ji}}, \quad (16)$$

where the first term $\frac{dE}{dh_j^{t'}}$ accounts for the evolution of the internal state $\mathbf{h}_j$ of neuron j and its influence on the neuron's firing at later time steps. For ALIF neurons and other biologically realistic neurons where the internal states change very slowly, this is relevant. The first term is replaced by an approximation $\frac{\partial E}{\partial z_j^{t'}}$ in the e-prop equation, with the derivative computed based on the observable state $z_j^{t'}$. This approximation ignores these indirect influences and captures only the direct influence of the observable state $z_j^{t'}$ on

the loss function E . The evolution of the internal state of neuron j is then accounted for by another term, that integrates the entire history of the synapse up to time t . This is achieved by first expressing $\frac{dE}{dh_j^{t'}}$ as a recursive function as:

$$\frac{dE}{dh_j^{t'}} = \frac{\partial E}{\partial z_j^{t'}} \frac{\partial z_j^{t'}}{\partial h_j^{t'}} + \frac{dE}{dh_j^{t'+1}} \frac{\partial h_j^{t'+1}}{\partial h_j^{t'}}, \quad (17)$$

where $\frac{\partial E}{\partial z_j^{t'}}$ is defined as the learning signal $L_j^{t'}$. By substituting Eq. (17) into Eq. (16), we obtain:

$$\frac{dE}{dW_{ji}} = \sum_{t'} \left(L_j^{t'} \frac{\partial z_j^{t'}}{\partial h_j^{t'}} + \left(L_j^{t'+1} \frac{\partial z_j^{t'+1}}{\partial h_j^{t'+1}} + (\dots) \frac{\partial h_j^{t'+2}}{\partial h_j^{t'+1}} \right) \frac{\partial h_j^{t'+1}}{\partial h_j^{t'}} \right) \cdot \frac{\partial h_j^{t'}}{\partial W_{ji}}. \quad (18)$$

The terms within the parentheses in Eq. (18) can be reformulated by adding a second summation indexed by t and pulling the learning signal $L_j^{t'}$ out by swapping the summation indices, as shown below:

$$\frac{dE}{dW_{ji}} = \sum_{t'} \sum_{t \geq t'} L_j^{t'} \frac{\partial z_j^{t'}}{\partial h_j^{t'}} \frac{\partial h_j^{t'}}{\partial h_j^{t-1}} \dots \frac{\partial h_j^{t'+1}}{\partial h_j^{t'}} \cdot \frac{\partial h_j^{t'}}{\partial W_{ji}}, \quad (19)$$

$$= \sum_{t'} L_j^{t'} \frac{\partial z_j^{t'}}{\partial h_j^{t'}} \underbrace{\sum_{t' \leq t} \frac{\partial h_j^{t'}}{\partial h_j^{t-1}} \dots \frac{\partial h_j^{t'+1}}{\partial h_j^{t'}}}_{\text{eligibility trace } e_{ji}^{t'}} \cdot \frac{\partial h_j^{t'}}{\partial W_{ji}}. \quad (20)$$

The second term $e_{ji}^{t'}$ is called the eligibility trace, gathers the evolution of the internal state of neuron j and is a product of $\frac{\partial z_j^{t'}}{\partial h_j^{t'}}$ and the eligibility vector $e_{ji}^{t'}$. The eligibility vector and the hidden state have the same dimension i.e., $\{e_{ji}^{t'}, h_j^{t'}\} \in \mathbb{R}^d$, with $d = 1$ for the LIF neuron and $d = 2$ for the ALIF neuron. The eligibility vector is a recursive function given by:

$$e_{ji}^{t'} = \frac{\partial h_j^{t'}}{\partial h_j^{t-1}} \cdot e_{ji}^{t-1} + \frac{\partial h_j^{t'}}{\partial W_{ji}}. \quad (21)$$

This makes e-prop an online learning rule since the loss gradients can be computed locally at the current time step t , based on the learning signal and eligibility traces. Thus, the fundamental equation of e-prop for a single layer can be written as:

$$\frac{dE}{dW_{ji}} = \sum_t \frac{\partial E}{\partial z_j^t} \cdot \left[\frac{dz_j^t}{dW_{ji}} \right]_{\text{local}}. \quad (22)$$

The fundamental e-prop framework can be extended to multiple layers [84] by incorporating distinct learning signals and eligibility traces for each layer as shown below:

$$\frac{dE}{dW_{ji}} = \sum_{t=1}^T \sum_{l=1}^L L_j^{t,l} e_{ji}^{t,l} \quad (23)$$

Following prior work, we briefly discuss the derivation of the eligibility trace for ALIF neurons to establish notation. From Eq. (3), it is clear that the eligibility vector is two-dimensional i.e., $e_{ji}^{t'} \stackrel{\text{def}}{=} [e_{ji,v}^{t'}, e_{ji,a}^{t'}]$. The dynamics of the membrane potential and the adaptive threshold are captured by the derivatives in a 2×2 matrix given by:

$$\frac{\partial h_j^{t+1}}{\partial h_j^t} = \begin{bmatrix} \frac{\partial v_j^{t+1}}{\partial v_j^t} & \frac{\partial v_j^{t+1}}{\partial a_j^t} \\ \frac{\partial a_j^{t+1}}{\partial v_j^t} & \frac{\partial a_j^{t+1}}{\partial a_j^t} \end{bmatrix} \quad (24)$$

$$= \begin{bmatrix} \alpha & 0 \\ \psi_j^t & \rho - \psi_j^t \beta \end{bmatrix}, \quad (25)$$

where the derivatives are computed based on Eq. (2). Since the spikes z_j^t are non-differentiable, a pseudo derivative ψ_j^t is used to compute the derivative in the backward pass. While [77] employ a piecewise linear surrogate derivative, we instead use the derivative of a fast sigmoid function as the surrogate gradient, given by:

$$\frac{\partial z_j^t}{\partial v_j^t} = \psi_j^t \approx \frac{1}{(1 + k|v_j^t|)^2}, \quad (26)$$

where k is the slope of the fast sigmoid surrogate gradient function. The partial derivative $\frac{\partial h_j^{t+1}}{\partial h_j^t}$ accounts only the direct influence of h_j^t on h_j^{t+1} and ignores the indirect influences of h_j^t on h_j^{t+1} through the observable states z^t . With $\frac{\partial h_j^t}{\partial W_{ji}} = \left[\frac{\partial v_j^t}{\partial W_{ji}}, \frac{\partial a_j^t}{\partial W_{ji}} \right] = [z_j^{t-1}, 0]$,

Eq. (21) can be written as:

$$\begin{bmatrix} \epsilon_{ji,v}^{t+1} \\ \epsilon_{ji,a}^{t+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial v_j^{t+1}}{\partial v_j^t} & \frac{\partial v_j^{t+1}}{\partial a_j^t} \\ \frac{\partial a_j^{t+1}}{\partial v_j^t} & \frac{\partial a_j^{t+1}}{\partial a_j^t} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{ji,v}^t \\ \epsilon_{ji,a}^t \end{bmatrix} + \begin{bmatrix} \frac{\partial v_j^{t+1}}{\partial W_{ji}} \\ \frac{\partial a_j^{t+1}}{\partial W_{ji}} \end{bmatrix} \quad (27)$$

$$= \begin{bmatrix} \alpha & 0 \\ \psi_j^t & \rho - \psi_j^t \beta \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{ji,v}^t \\ \epsilon_{ji,a}^t \end{bmatrix} + \begin{bmatrix} z_i^t \\ 0 \end{bmatrix} \quad (28)$$

$$= \begin{bmatrix} \alpha \epsilon_{ji,v}^t + z_i^t \\ \psi_j^t \epsilon_{ji,v}^t + (\rho - \psi_j^t \beta) \epsilon_{ji,a}^t \end{bmatrix}. \quad (29)$$

The first component of the eligibility vector $\epsilon_{ji,v}^t$ can be replaced by a low-pass filter of the presynaptic spike-train $F_\alpha(z_i^t) = \bar{z}_i^t$, where the temporal filter F_α for any spike-train x^t , is given by:

$$F_\alpha(x^t) = \alpha F_\alpha(x^{t-1}) + x^t. \quad (30)$$

To compute the eligibility trace, the eligibility vector is multiplied by $\frac{\partial z_j^t}{\partial h_j^t}$ as shown in Eq. (20). This gives the final form of the eligibility trace as:

$$e_{ji}^t = \begin{bmatrix} \frac{\partial z_j^t}{\partial v_j^t} & \frac{\partial z_j^t}{\partial a_j^t} \end{bmatrix} \begin{bmatrix} \epsilon_{ji,v}^t \\ \epsilon_{ji,a}^t \end{bmatrix} = \psi_j^t (\bar{z}_i^{t-1} - \beta \epsilon_{ji,a}^t) \quad (31)$$

3. Data acquisition

In this study, we investigate the performance of the different learning rules based on datasets representing various material behaviors. A 3-element rheological model is used to obtain data for the viscoelastic material behavior. To capture the geometrically and physically nonlinear viscoplastic behavior, shock tube experiments were used since they resemble a complex real-world problem.

3.1. Rheological model

Rheological models are widely employed to describe time-dependent mechanical behavior of materials, particularly when both elastic and viscous effects are significant. Classical viscoelastic models consist of springs and dampers, arranged in series or parallel to capture different deformation mechanisms. In this study, we adopt a rheological representation of a Maxwell element and a damper with a viscosity of η_0 , connected in parallel as shown in Fig. 2(a). The Maxwell element is composed of a spring of stiffness E_1 in series with a damper of viscosity η_1 . The constitutive equation of the model is defined as:

$$\dot{\sigma} = \eta_0 \ddot{\epsilon} + \left(E_1 + \frac{E_1 \eta_0}{\eta_1} \right) \dot{\epsilon} - \frac{E_1}{\eta_1} \sigma \quad (32)$$

The typical stress response of the model σ based on the strain rate $\dot{\epsilon}$ is shown in Fig. 2(b). The strain rate here is expressed as a step function. It can be seen that the stress-strain rate relationship is very similar to that of a LIF neuron as shown in Fig. 1. This similarity becomes evident when comparing the constitutive relation of the rheological model in Eq. (32) with the membrane potential dynamics of the LIF neuron in Eq. (2). Both are first-order linear ODEs characterized by a decay term proportional to the state variable and a driving term determined by the external input. In the rheological model, stress σ relaxes with a rate $\frac{E_1}{\eta_1}$ and is driven by the strain rate $\dot{\epsilon}$ and its derivative, whereas in the LIF neuron the membrane potential $v(t)$ decays with time constant τ_m and is driven by the input current $I(t)$. Thus, both systems implement a form of leaky integration: the state variable (stress or membrane potential) integrates the input over time while simultaneously decaying toward equilibrium. This structural analogy provides a physical justification for employing spiking neurons in mechanics, extending their relevance beyond energy efficiency to a framework grounded in similar governing equations. The proof of this analogy will be discussed in Section 5.

Since the rheological model does not feature an explicit reset mechanism as in the LIF neuron, the strain rate and its derivative are expressed as a series of piecewise constant step functions as shown in Fig. 3. The inputs were chosen in this form to enable a direct comparison between the two dynamical systems under a discrete-time excitation. This representation induces abrupt changes in the input, effectively emulating switching events that support the analogy to the reset behavior in the LIF neurons. The use of an adaptive-threshold LIF neuron further strengthens the correspondence by introducing a memory effect similar to the transient responses observed in viscoelastic materials. Importantly, the qualitative correspondence between the rheological model and the neuron dynamics is not restricted to this specific input choice and holds for general input signals.

Two sets of material parameters were employed in the rheological model. For Set 1, the elastic modulus was $E_1 = 600$ Pa, with viscosities $\eta_1 = 10$ Pa s and $\eta_2 = 5$ Pa s. Set 2 was characterized by an elastic modulus of $E_1 = 800$ Pa, and viscosities $\eta_1 = 20$ Pa s and $\eta_2 = 10$ Pa s. The model was simulated for 1 s with a total of 1000 time steps.

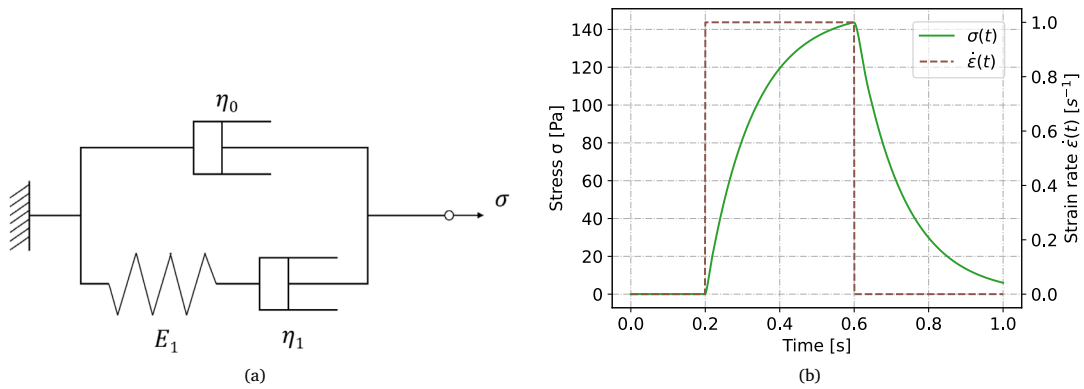


Fig. 2. (a) The 3-element rheological model. (b) Stress response of the model to a constant strain rate input.

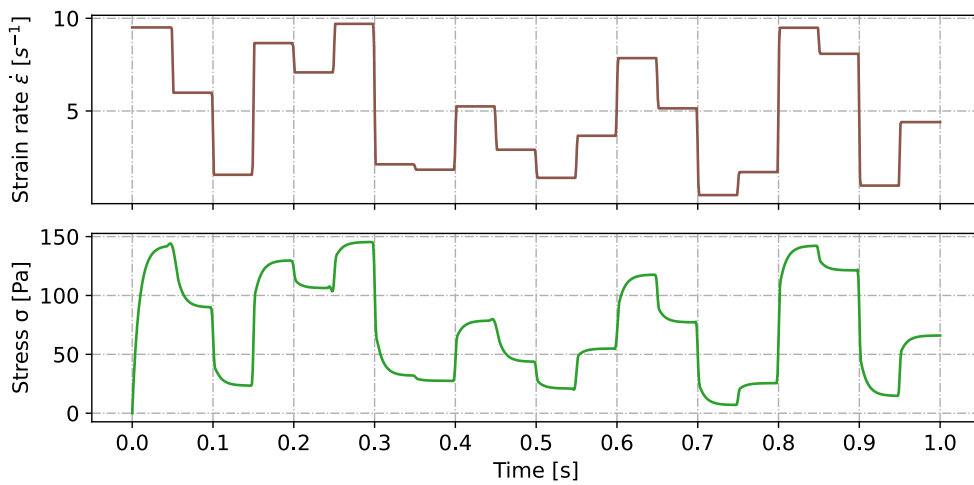


Fig. 3. Stress response of the rheological model based on the strain rate. The strain rate is described by a series of piecewise constant step functions.

3.2. Shock tube experiments

The data used here is from the shock tube experiments conducted in [88–90] to investigate the viscoplastic vibrations in a plate subjected to impulsive loadings. The dataset comprises geometrically and physically nonlinear structural deformations over a broad spectrum of strain rates. The schematic representation of the setup is shown in Fig. 4. The shock tube consists of two chambers separated by a hostaphane diaphragm. High pressure is maintained in one chamber and the other chamber has a low pressure. The plate specimen to be investigated is clamped at the end of the tube. The plate is made of steel with a diameter of 0.138 m and a thickness of 0.002 m. The diameter of the area subjected to the impulse is only 0.108 m as a result of clamping. The high pressure chamber is filled with gas until the diaphragm is ruptured. As a result, a shock wave of high density and high pressure is transmitted through the low pressure chamber, hitting the plate specimen. Piezoelectric sensors located on a separate ring flange on the front, capture the pressure history accurately. The midpoint displacement of the plate was recorded using a capacitive displacement sensor, with the circular front electrode of the measurement device and the specimen itself acting as the two plates of the capacitor. The voltage signal is converted into displacement by a calibration curve relating the measured voltage and the distance between the capacitor plates. Since the dielectric properties of the capacitor are temperature dependent, the calibration has to be performed newly after each experiment. The pressure and displacement signals were recorded over an impact duration of 0.015 s with a sampling frequency of 1 MHz. This results in sequences containing approximately 15 000 time steps.

Feeding longer sequences directly into a recurrent neural network would be computationally inefficient and may not be effective for training. To address this, the datasets are divided into shorter segments of 500 time steps, which can later be sequentially provided to the network while maintaining temporal continuity between segments through stateful training. The acquired datasets, consisting of approximately 1000 samples, are normalized between $[-1, 1]$ to avoid the problem of vanishing and exploding gradients. A 80–10–10 split was performed randomly to obtain the training, validation and testing datasets, respectively.

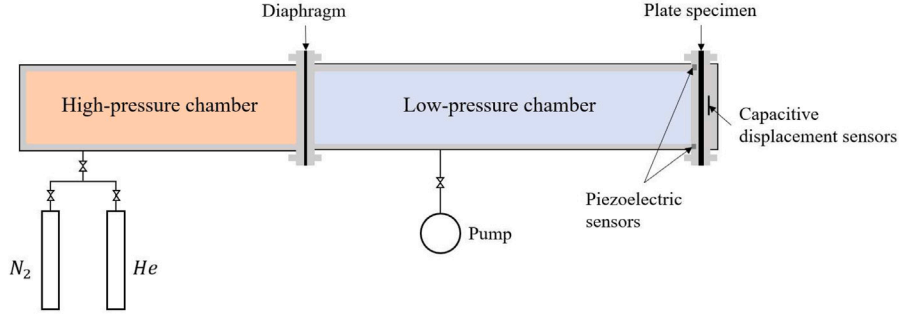


Fig. 4. Schematic representation of the shock tube experiment.

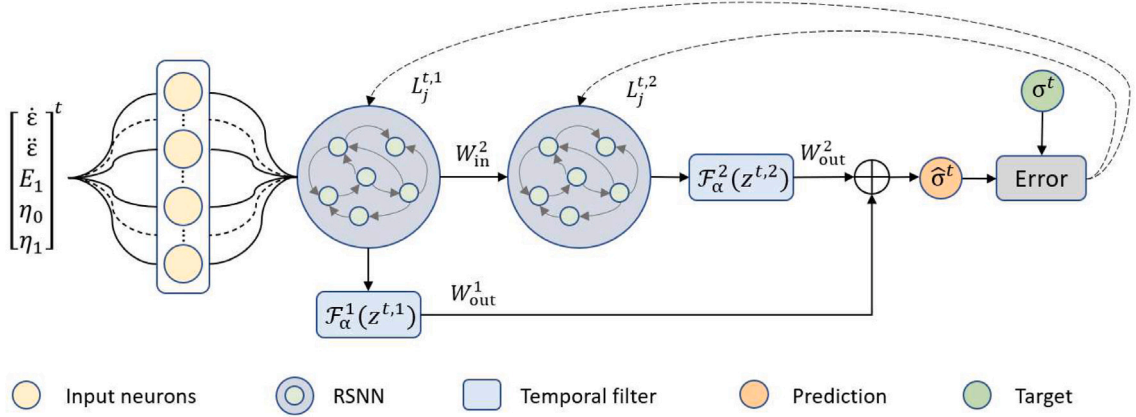


Fig. 5. Model architecture used to predict the stress based on the strain derivatives and the rheological material parameters in the viscoelastic dataset.

4. Model architectures

4.1. Architecture for the rheological dataset

Fig. 5 illustrates the network architecture used to train the rheological model dataset. The network receives an input vector composed of the strain rate $\dot{\epsilon}$ and its derivative $\ddot{\epsilon}$, together with the material parameters E_1 , η_0 , and η_1 , which characterize the viscoelastic behavior according to the rheological model shown in Fig. 2(a). The input vector is first processed by the input neurons, which serve as a transformation layer mapping the input features to the dimensionality of the recurrent spiking neuron population. This RSNN consists of ALIF neurons with all-to-all recurrent connectivity, meaning that each neuron is connected to every other neuron within the recurrent pool, excluding self-connections, consistent with the original formulation [77]. The RSNN generates a spike-train $\mathbf{z}^1$ which is then propagated to a second RSNN layer. The spike-trains from both layers, $\mathbf{z}^1$ and $\mathbf{z}^2$, are subsequently passed through a temporal filter $\mathcal{F}(\alpha)$, described in Eq. (30), with the trainable filter parameter α unique to each layer. The filtered spike-trains are transformed to the desired output dimensions via the respective readout weights W_{out}^1 and W_{out}^2 . By employing a current-based neuron model rather than a conductance-based one, the network avoids the need for explicit spike encoding of the input signals, which can be directly provided to the first layer. Unlike conventional approaches, where the membrane potential of the final layer is used to generate continuous output signals, the present architecture directly utilizes the spike-trains from each recurrent layer to compute the output. This design allows each RSNN layer to contribute to the final prediction through its own readout pathway, thereby providing distinct layer-specific learning signals. Such a structure is particularly well suited for the e-prop learning rule, which relies on local eligibility traces and layer-wise learning signals to approximate backpropagation through time. The error between the predicted stress $\hat{\sigma}^t$ and the target σ^t is propagated backward as a layer-wise learning signal L_j^t , modulating the eligibility traces within each RSNN layer. This interaction enables biologically plausible, layer-local learning without the need for full backpropagation through time.

4.2. Architecture for the shock-tube dataset

For the shock-tube experiment dataset, a much more complex architecture is used, as shown in Fig. 6. This can be seen as an extension to the previous architecture with a few modifications, which are required to approximate the highly nonlinear dynamic

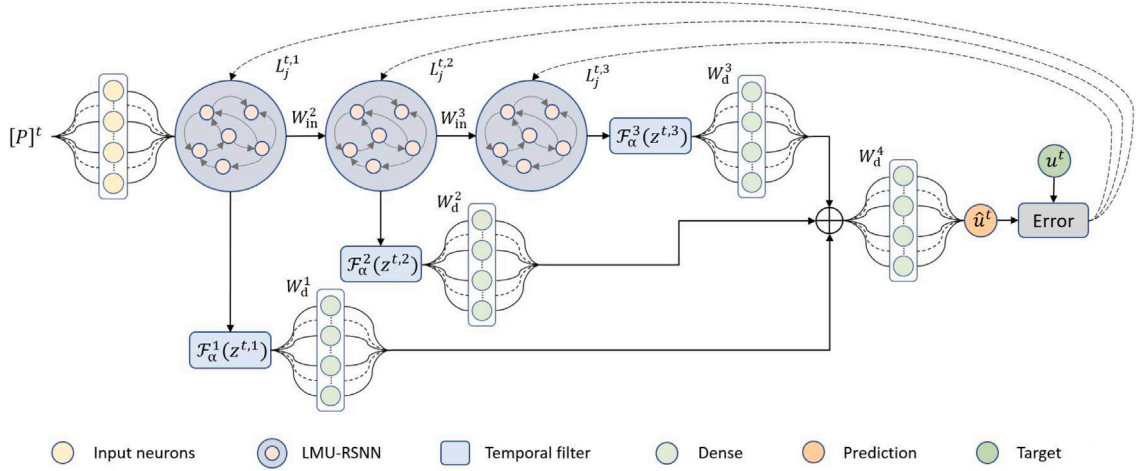


Fig. 6. Multi-layer network architecture used to predict the midpoint displacement of a plate subjected to shock wave loads.

Table 1

Network parameters for the proposed architectures.

Architecture	Layer	Parameters	Values
Architecture 1 (Fig. 5)	RSNN 1	W_{in}^1, W_{rec}^1	5, 128
	RSNN 2	W_{in}^2, W_{rec}^2	128, 128
	Readout	W_{out}^1, W_{out}^2	128, 128
Architecture 2 (Fig. 6)	LMU-RSNN 1	$d^1, \theta^1, W_{in}^1, W_{rec}^1$	50, 500, 50, 50
	LMU-RSNN 2	$d^2, \theta^2, W_{in}^2, W_{rec}^2$	50, 500, 50, 50
	LMU-RSNN 3	$d^3, \theta^3, W_{in}^3, W_{rec}^3$	50, 500, 50, 50
	Dense	$W_d^1, W_d^2, W_d^3, W_d^4$	64, 64, 64, 192

responses. The network receives the pressure history P as input, which is first transformed by the input neurons and then propagated through a series of RSNN layers. The spike-trains z^l generated by each layer l are transformed by the corresponding input weight matrix W_{in}^{l+1} , acting as the input to the subsequent layer. Since ALIF neurons have only a limited intrinsic memory, they are not well suited to represent highly nonlinear temporal dependencies over longer time steps. To address this limitation, we integrate LMUs into the recurrent layers, thereby providing enhanced memory capacity. Similar to the previous architecture, the generated spike-train from each RSNN layer is passed through distinct temporal filters. In preliminary experiments, we observed that directly multiplying the filtered spike-trains with the readout weights produced noisy outputs, a result of information loss due to the discontinuous nature of spikes. To mitigate this, we replace the readout weights with dense layers W_d^k , $k \in [1, 4]$ employing a tanh activation function. These dense layers serve as additional filters, smoothing the outputs and reducing their noise. The outputs from all dense layers are concatenated and passed through a final dense layer, which further refines the signal and maps it to the desired output dimensions.

The non-spiking counterparts of the above architectures are employed by replacing the temporal filters with dropout layers, with a dropout probability of 0.15. This facilitates regularization in the networks and also partially emulates the irregular activation patterns observed in the spiking networks. In contrast, explicit dropout is not required for the spiking networks, as the intrinsic spiking activity introduces sparsity and discontinuities in the network dynamics, which act as an implicit form of regularization.

4.3. Training strategy

To compare the performance of the two learning paradigms, the proposed architectures are trained independently using either e-prop or BPTT. Training the networks with BPTT is straightforward, with gradients accumulated across all layers and time steps, as described in Eq. (14). The e-prop training follows the formulation presented in Section 2.3, where layer-specific learning signals – computed based on the prediction error and spike activity – are combined with local eligibility traces to estimate the gradients.

For the architecture shown in Fig. 6, a hybrid training approach is adopted that combines e-prop and BPTT. Specifically, the hidden cell states of the LMU-RSNN layers are trained using e-prop, while all other trainable parameters, including those of the dense layers and the LMU's memory component, are updated using BPTT. This hybrid approach leverages the biological plausibility and computational efficiency of e-prop for the recurrent spiking components, while retaining the gradient accuracy and stability of BPTT for the continuous, differentiable parts of the network, such as the LMU memory and dense layers.

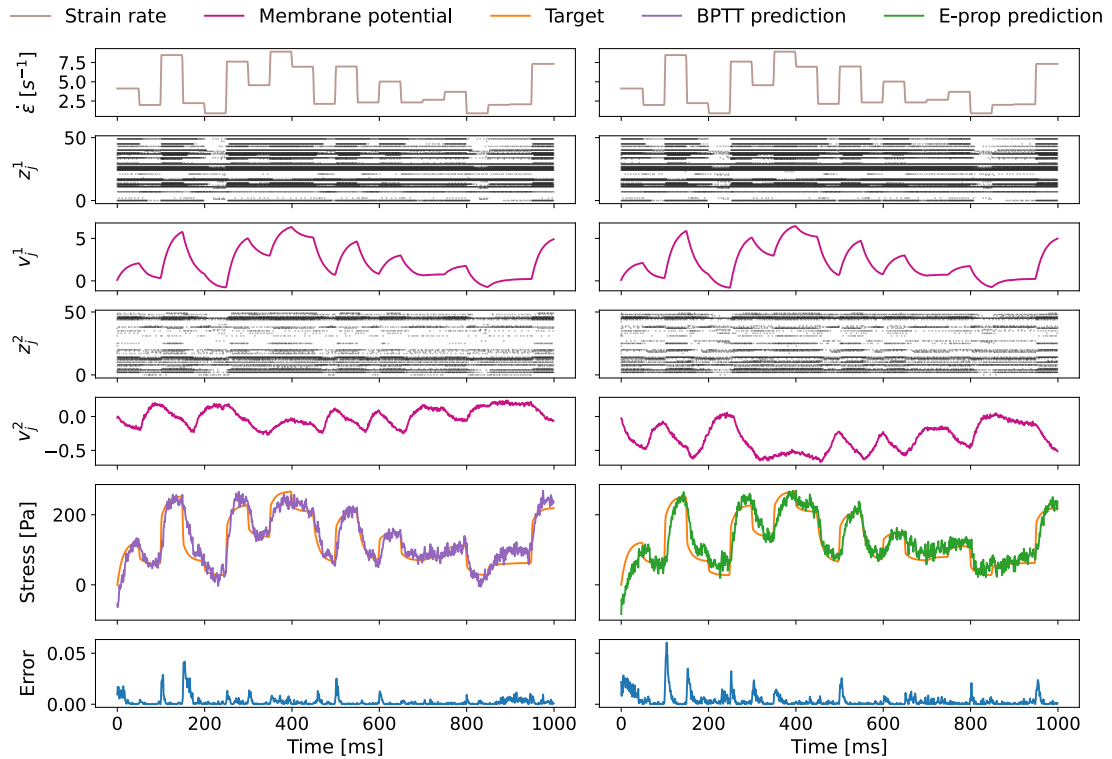


Fig. 7. Network dynamics and prediction performance on the viscoelastic testing dataset. The plots display the input strain rate, membrane potentials, spike activity, the predicted and target stress signals, and the corresponding prediction error. The left panel shows a sample trained using BPTT, whereas the right panel shows the results for the same sample obtained using e-prop.

4.4. Implementation details

The details of the network parameters of the two architectures can be found in Table 1. For the non-spiking versions of these networks, the ALIF neurons are replaced by RNN cells in the first network and by LSTM cells in the second. The networks are built and trained in TensorFlow [91] with a batch size of 16 using the Adam optimizer [92] with a learning rate of 1×10^{-3} . The loss is computed as the mean squared error (MSE), and training is performed on an NVIDIA RTX 5000 Ada Generation GPU.

5. Results and discussion

5.1. Qualitative analysis of network dynamics

Once the networks have been trained, their performance is evaluated on both the training and testing datasets. Figs. 7 and 8 illustrate the results obtained for the viscoelastic and shock tube datasets, respectively, using both BPTT and e-prop learning rules. Each panel presents the internal neuron dynamics, including spike activity, membrane potentials, and adaptive thresholds, for different input samples. The figures also compare the predicted stress and midpoint displacement responses with the corresponding target signals and show the estimated error at every time step. A closer examination of these internal dynamics reveal some distinctive characteristics of the spiking models. Additional sample-specific results for both training and testing datasets are provided in Appendix A, Figs. A.12–A.17.

A clear correspondence between the firing of neurons and the input signal can be observed in the first layer, where distinct groups of neurons become active during the increasing and decreasing phases of the input samples. This relationship transitions from a spike–input to a spike–output coupling in the last layer, as the neurons directly contribute to output generation. This behavior is also reflected in the evolution of the membrane potentials across layers, shown in Fig. 7, which exhibit a partial resemblance to the target stress responses. The fact that no single neuron replicates the target dynamics exactly, but rather that the population collectively reconstructs the signal, indicates a distributed coding scheme within the network.

The adaptive threshold dynamics shown in Fig. 8 further influence the spike count. Neurons that fire more frequently gradually increase their thresholds, thereby regulating the excessive firing and stabilizing the network activity. The plots also reveal the distinct

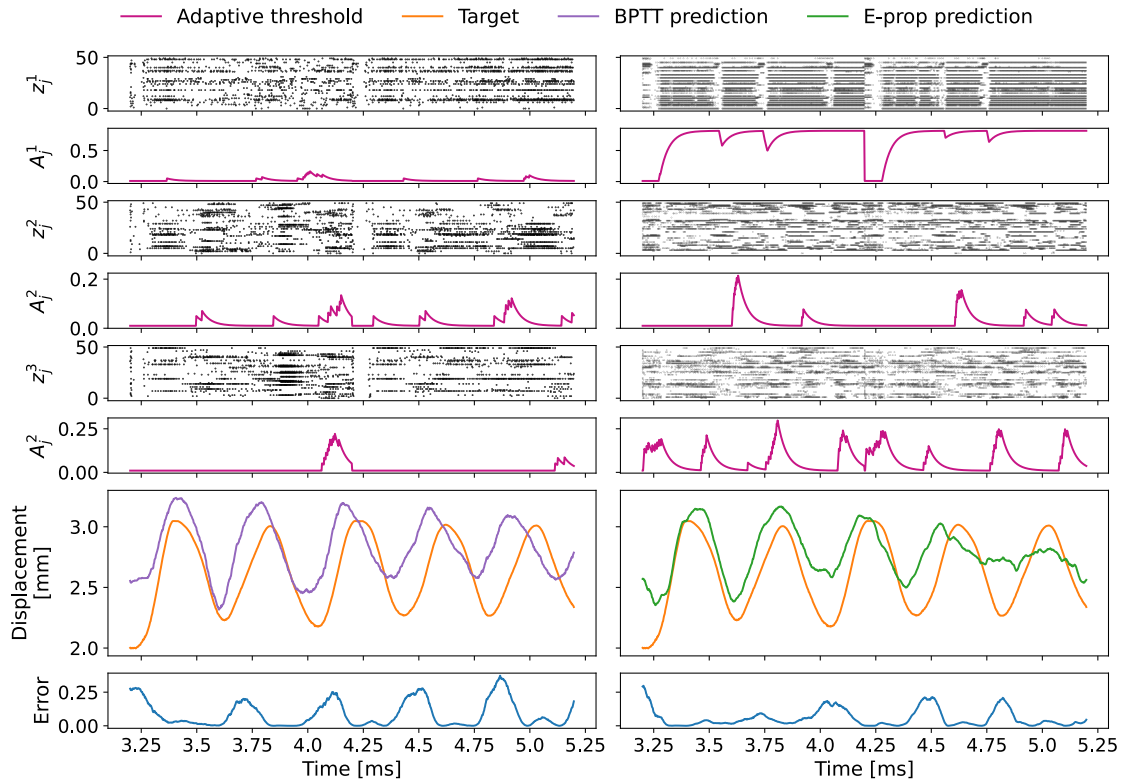


Fig. 8. Network dynamics and prediction performance on the shock tube testing dataset. The plots display the input strain rate, adaptive thresholds, spike activity, the predicted and target displacement signals, and the corresponding prediction error. The left panel shows a sample trained using BPTT, whereas the right panel shows the results for the same sample obtained using e-prop (hybrid).

firing rates and membrane time constants of sample neurons. The predicted stress responses for the viscoelastic dataset appear noisier compared to those for the shock tube dataset. This arises from directly using the filtered spike-trains from the RSNNs for output computation, as shown in Fig. 5. The filtered spike-trains lose some information primarily due to the binary and discontinuous nature of spikes, as well as the resulting sparse and irregular activity patterns. This effect can be mitigated by introducing dense layers in place of the readout weights, as shown in Fig. 6. The continuous nonlinear activations in these dense layers help smoothen the outputs and improve the overall prediction quality.

5.2. Training performance and learning efficiency

The training performance of the proposed architectures on the rheological model and shock tube datasets was evaluated using BPTT, e-prop, and their non-spiking counterparts. Across both datasets, networks trained with BPTT consistently achieve lower mean-squared-error (MSE) in both training and testing, confirming that BPTT provides more accurate gradient estimates. This behavior is expected, as BPTT computes exact gradients by unrolling the network through time, thereby capturing the full temporal dependencies across all time steps. In contrast, networks trained with e-prop require more epochs to reach convergence and yield slightly higher MSE values. This trend is clearly illustrated in Fig. 9, which shows the evolution of the average training and validation losses for the networks trained on the rheological dataset over 10 runs and 200 epochs. The slower convergence of e-prop is primarily due to its approximate gradient estimation, which relies on local eligibility traces and broadcast signals rather than backpropagating exact error gradients through time. While this enables online and memory-efficient learning without storing past states, the resulting gradient estimates are noisier, leading to slower convergence and marginally reduced accuracy relative to BPTT. For the shock tube dataset, the overall performance difference between e-prop and BPTT follows the same trend, though it is somewhat more pronounced—potentially due to the hybrid training strategy combining e-prop and BPTT, as the full potential of e-prop for RSNNs integrated with LMUs has not yet been explored. A complete summary of average MSE values for both datasets and all learning rules is provided in Appendix B (Table B.2).

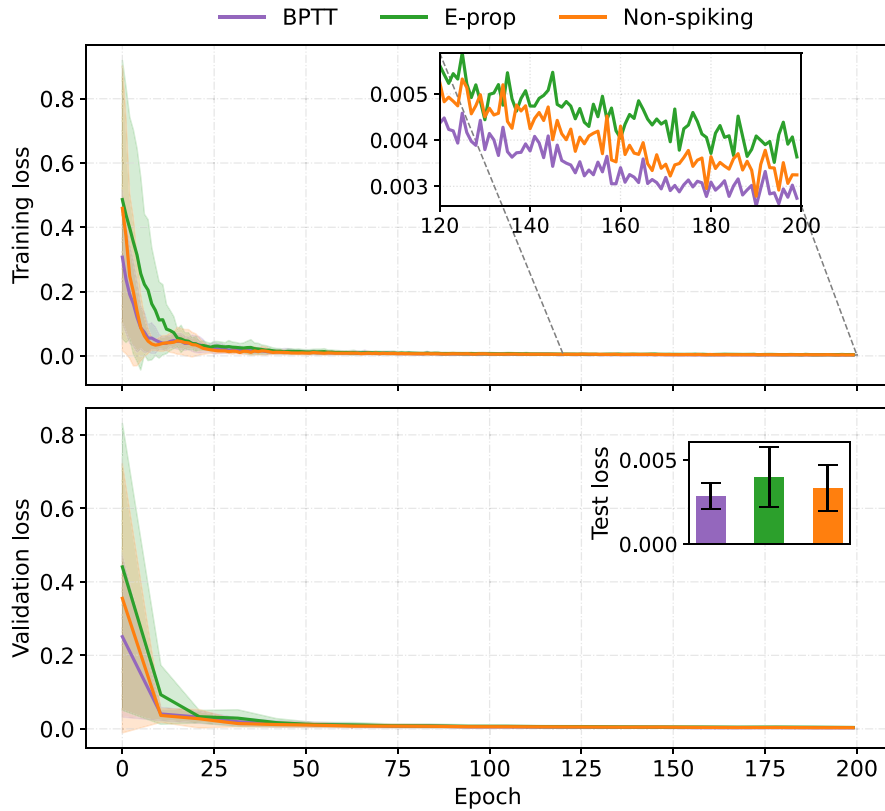


Fig. 9. Training, validation and test performance of BPTT, e-prop, and a non-spiking baseline. The upper panel shows the mean training loss with standard deviations over multiple runs for 200 epochs, with an inset highlighting the convergence behavior in the final epochs. The lower panel shows the corresponding validation losses and a bar plot of the test losses after 200 training epochs.

5.2.1. Energy and memory efficiency

In terms of energy efficiency, e-prop remains theoretically more sustainable than BPTT, as it eliminates the need to store and backpropagate through all time steps. As a result, e-prop's memory usage does not scale with time steps unlike BPTT. These characteristics make e-prop particularly attractive for large-scale recurrent spiking networks and for deployment on neuromorphic hardware. In [85], such online learning on the SpiNNaker 2 neuromorphic platform using e-prop was demonstrated, empirically validating its energy- and memory-efficiency advantages. These developments open up promising directions for applying e-prop-trained RSNNs to dynamic mechanical systems on neuromorphic hardware.

5.3. Comparison of spiking and non-spiking architectures

A notable observation is the difference between spiking and non-spiking networks across the two datasets. In the shock tube experiment, the non-spiking network outperforms the spiking version both in speed and accuracy (Table B.2). However, in the rheological model dataset, the spiking network trained with BPTT achieves faster convergence and slightly lower MSE than the non-spiking version (Table B.2).

Figs. 10 and 11 show the predictions of the networks trained with the two learning rules. The plots also show the predictions of the non-spiking networks along with the target responses. This contrasting behavior aligns with the analogy discussed in Section 3.1: The stress-strain dynamics of the rheological model closely resemble the membrane potential dynamics of the LIF neuron—both being governed by first-order linear ODEs with exponential decay and input-driven terms. Owing to this structural similarity, the spiking network can represent the system's temporal dynamics more naturally, which likely facilitates faster learning in this specific case. Also, the non-spiking network has to learn the temporal dependencies implicitly through their parameters, which increases the training burden. Conversely, the shock tube dataset involves more complex and nonlinear transients that are not well captured by the neuron dynamics. This results in the slightly better performance of the non-spiking model in this setting.

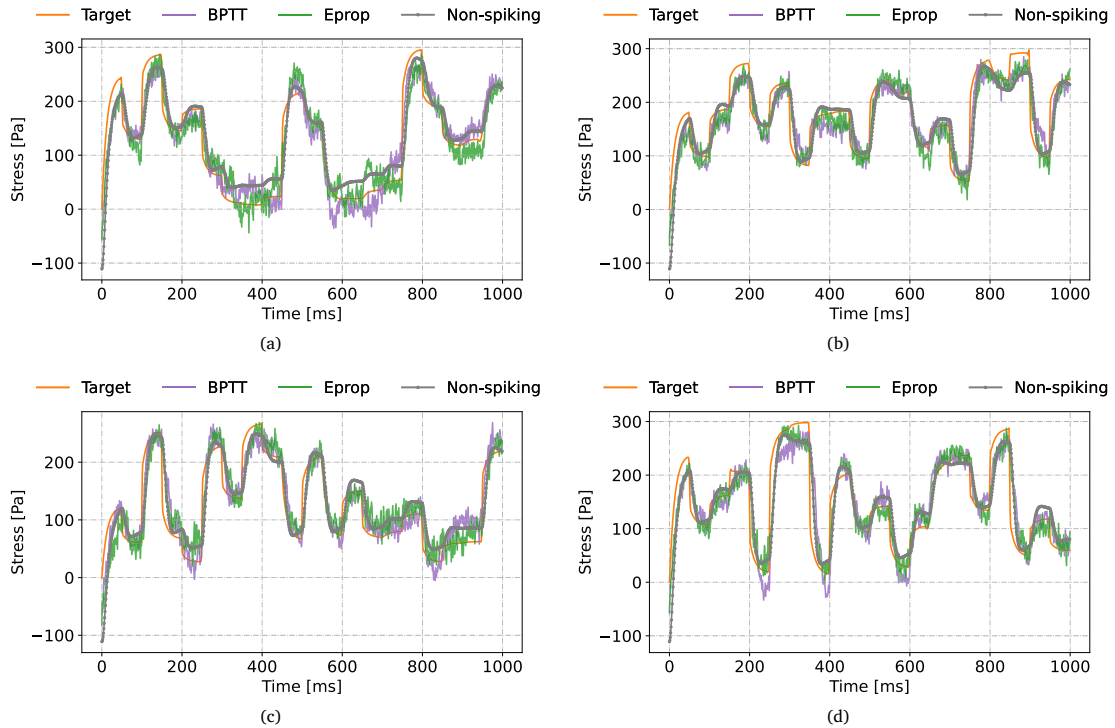


Fig. 10. Comparison of predicted and target responses for the viscoelastic dataset. The plots show the performance of the network trained using BPTT, e-prop and a non-spiking version. The top row (a, b) presents the predictions for the training dataset, while the bottom row (c, d) shows the results for the testing dataset.

5.4. Implications and future directions

While this analogy provides a strong physical and mathematical motivation for employing spiking neural networks in modeling mechanical systems, it does not imply that SNNs will universally outperform non-spiking models in all cases. Their advantage can be realized primarily when the underlying mechanical system shares structural similarities with the neuron dynamics. In other cases, SNNs may be limited in terms of accuracy and convergence speed, but can still offer significant benefits in terms of energy efficiency. Future improvements could be achieved through more comprehensive hyperparameter optimization and by extending the e-prop learning framework to the complete LMU-based recurrent architecture. Such developments would enable a fully biologically plausible yet computationally efficient modeling paradigm for complex dynamic mechanical systems.

6. Conclusion

This study presented a biologically plausible framework for modeling nonlinear and transient mechanical responses using RSNNs. Two datasets were considered—a viscoelastic rheological model and an impulsively loaded plate from shock tube experiments. By drawing an analogy between viscoelastic rheological models and the membrane dynamics of LIF neurons, a physical basis was established for employing SNNs in mechanics beyond their energy-efficiency benefits. A two-layer ALIF-based RSNN was trained on the viscoelastic dataset using both BPTT and the biologically plausible e-prop. For the shock tube dataset, a multi-layer recurrent spiking architecture combining LMUs with ALIF neurons was trained using a hybrid e-prop/BPTT approach to predict the transient midpoint displacements of plates. Comparative analyses revealed that BPTT achieved faster convergence and lower mean-squared errors, whereas e-prop enabled online, memory- and energy-efficient learning with greater biological plausibility. The spiking model exhibited superior performance for the rheological dataset, consistent with the structural similarity between the rheological model and the neuron dynamics, while the non-spiking model performed better for the complex shock tube responses. Overall, the proposed architectures demonstrate that recurrent spiking networks – particularly when aligned with the physics of the target system – can offer interpretable, energy-efficient alternatives to conventional neural models in computational mechanics. The present findings highlight the substantial potential for further research into biologically inspired, energy-efficient learning paradigms and their neuromorphic implementation for modeling complex mechanical systems.

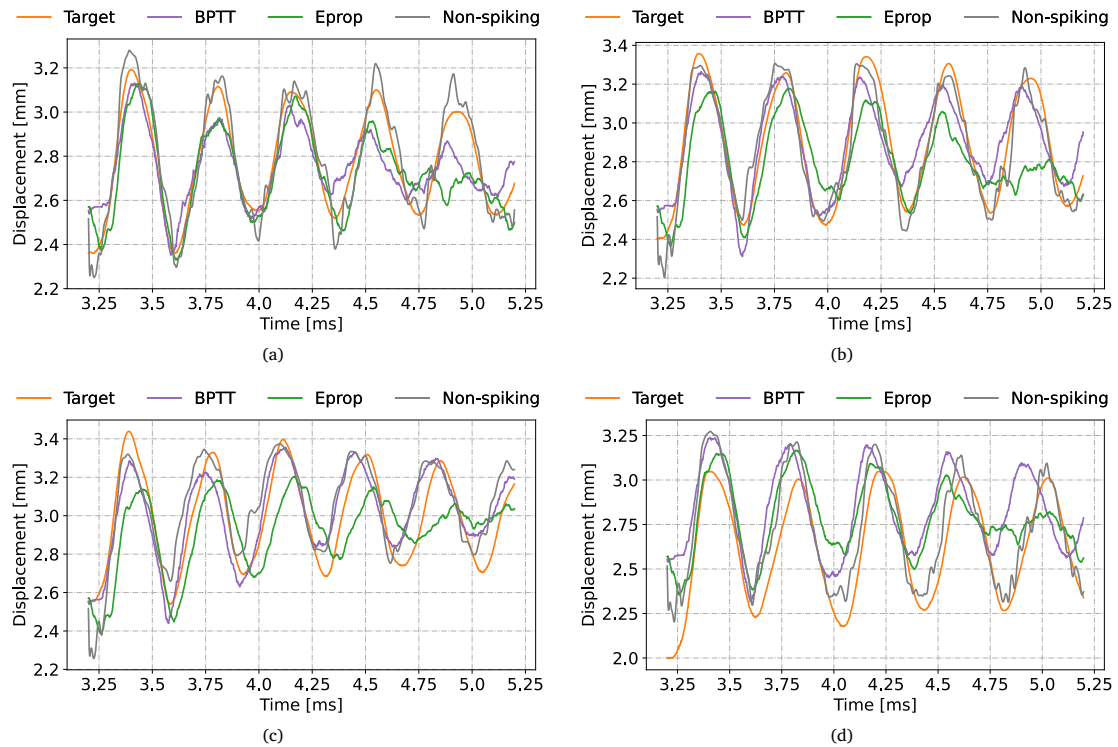


Fig. 11. Comparison of predicted and target responses for the shock tube dataset. The plots show the performance of the network trained using BPTT, e-prop (hybrid) and a non-spiking version. The top row (a, b) presents the predictions for the training dataset, while the bottom row (c, d) shows the results for the testing dataset.

CRediT authorship contribution statement

Vaishnav Bhaskaran: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Saurabh Balkrishna Tandale:** Writing – review & editing, Software, Funding acquisition, Conceptualization. **Vasileios Polydoros:** Supervision, Conceptualization. **Agnes Korcsak-Gorzo:** Writing – review & editing, Methodology, Conceptualization. **Abigail Morrison:** Writing – review & editing, Funding acquisition, Formal analysis. **Markus Diesmann:** Writing – review & editing, Funding acquisition, Formal analysis. **Marcus Stoffel:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge that this study was made possible by the support of the German Research Foundation (DFG) through two projects (Grant Nos. 504279932, 550613313) that build on one another. The authors acknowledge support from HiRSE_PS, the Helmholtz Platform for Research Software Engineering — Preparatory Study, an innovation pool project of the Helmholtz Association; funding from the European Union's Horizon Europe Programme under the Specific Grant Agreement No. 101147319 (EBRAINS 2.0 Project); and support from NeuroSys as part of the initiative “Clusters4Future” funded by the Federal Ministry of Education and Research BMBF (03ZU2106CB).

Appendix A. Network dynamics and prediction performance on the viscoelastic and shock tube datasets

This section presents additional results illustrating the network dynamics and prediction performance on the viscoelastic and shock tube datasets. For each figure, the plots display the input strain rate, membrane potentials/adaptive thresholds, spike activity, predicted and target stress/displacement signals, and the corresponding prediction error. The left panel shows a sample trained using BPTT, whereas the right panel shows the same sample evaluated with e-prop. Multiple samples from the training and testing datasets are presented to demonstrate the consistency of the network behavior and learning performance (see Figs. A.12–A.17).

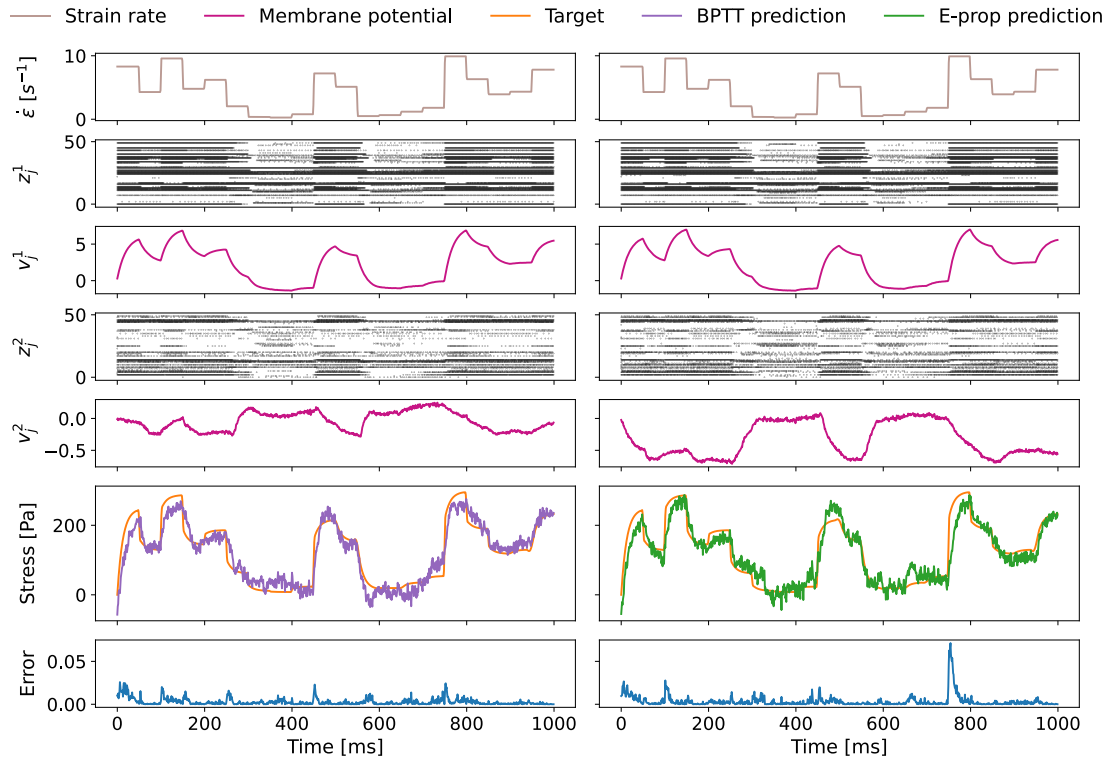


Fig. A.12. Network dynamics and prediction performance for train sample 1. Left: BPTT; Right: e-prop.

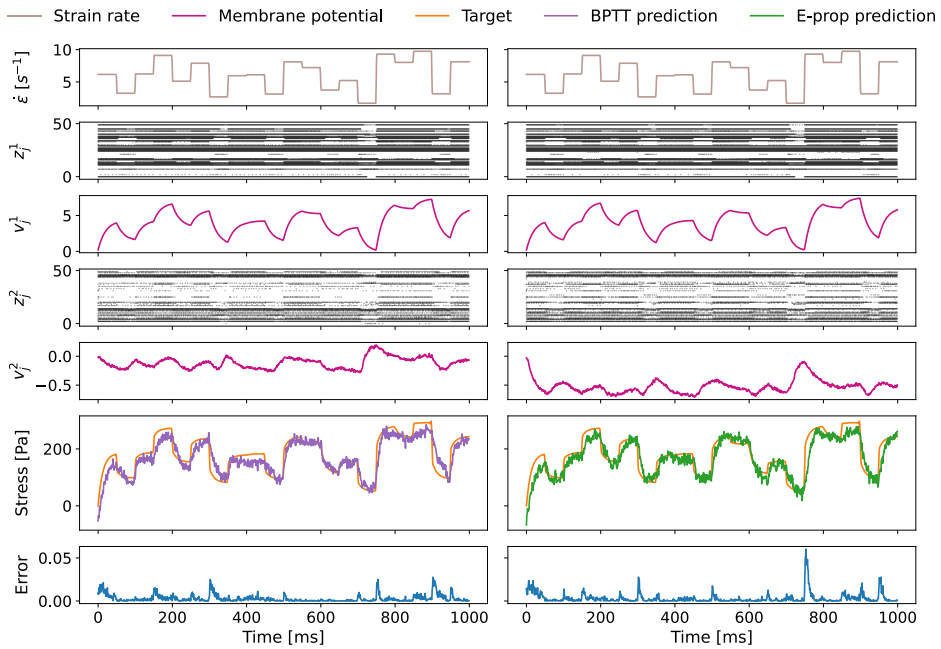


Fig. A.13. Network dynamics and prediction performance for train sample 2. Left: BPTT; Right: e-prop.

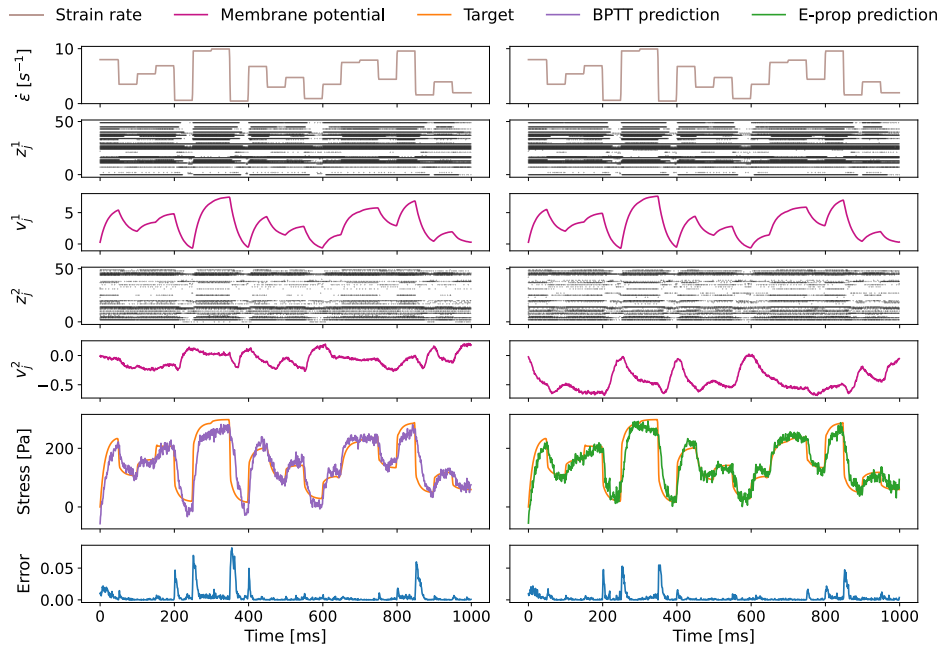


Fig. A.14. Network dynamics and prediction performance for test sample 1. Left: BPTT; Right: e-prop.

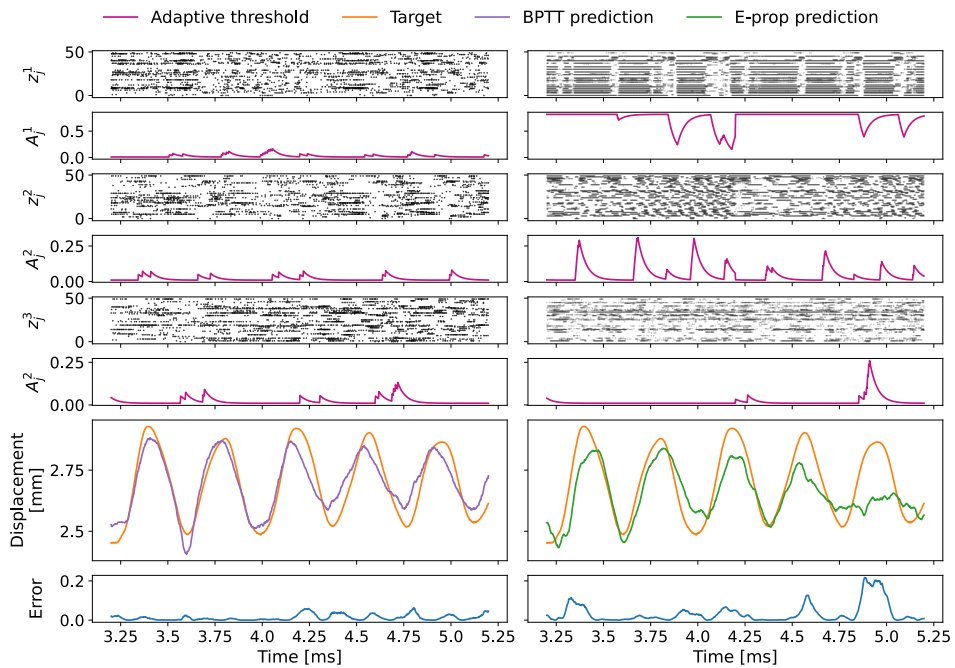


Fig. A.15. Network dynamics and prediction performance for train sample 1. Left: BPTT; Right: e-prop.

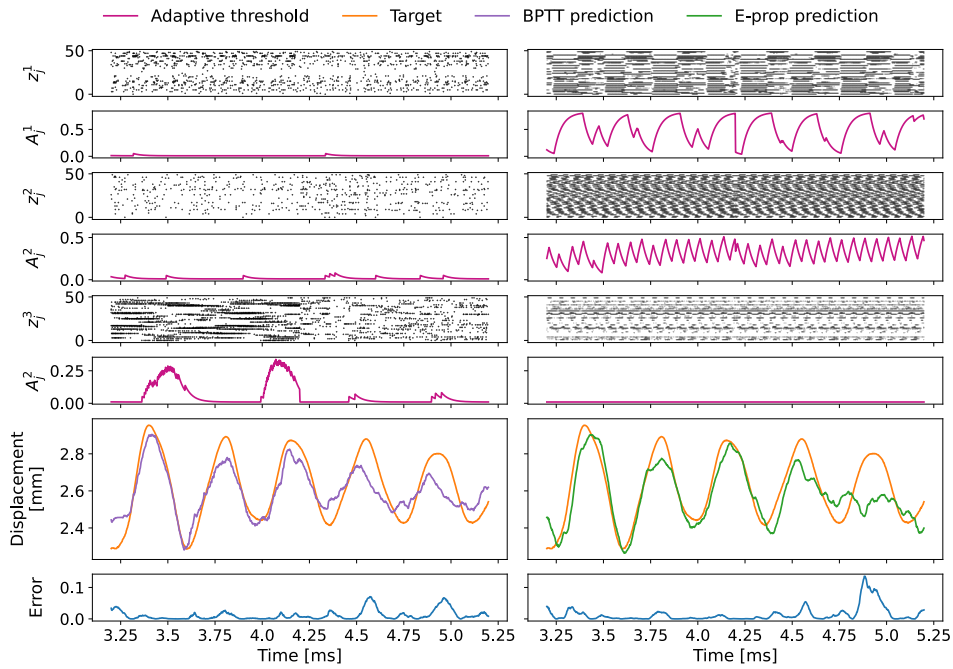


Fig. A.16. Network dynamics and prediction performance for train sample 2. Left: BPTT; Right: e-prop.

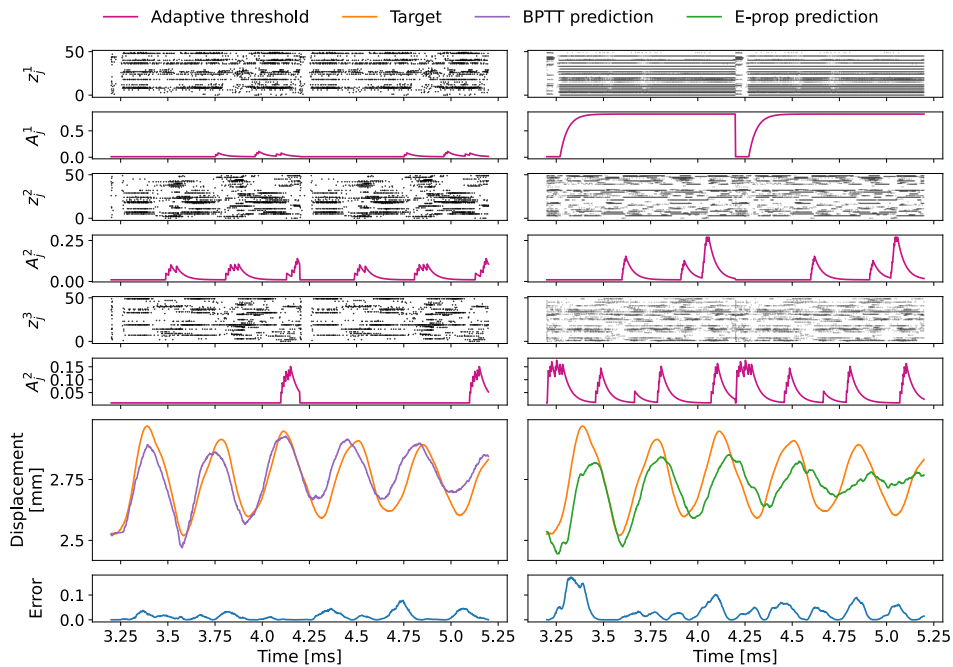


Fig. A.17. Network dynamics and prediction performance for test sample 1. Left: BPTT; Right: e-prop.

Appendix B. Average training performance

This section provides a summary of the average training performance of the proposed architectures on the viscoelastic and shock tube datasets. The table reports the average number of epochs required for convergence and the mean-squared error (MSE) for both training and testing datasets, comparing BPTT, e-prop, and non-spiking networks.

Table B.2

Training performance comparison of BPTT, e-prop, and non-spiking networks for the rheological model and shock tube experiment datasets.

Dataset	Method	Epochs	MSE (Train)	MSE (Test)
Rheological model	BPTT	180	0.00141	0.00165
	e-prop	440	0.00234	0.00273
	Non-spiking	230	0.00269	0.00335
Shock tube experiment	BPTT	940	0.0356	0.0155
	e-prop (hybrid)	1220	0.0578	0.0376
	Non-spiking	720	0.0142	0.0222

Data availability

Data will be made available on request.

References

- [1] F.A. Azevedo, L.R. Carvalho, L.T. Grinberg, J.M. Farfel, R.E. Ferretti, R.E. Leite, W.J. Filho, R. Lent, S. Herculano-Houzel, Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain, *J. Comp. Neurol.* 513 (5) (2009) 532–541, <http://dx.doi.org/10.1002/cne.21974>, URL <https://onlinelibrary.wiley.com/doi/10.1002/cne.21974>.
- [2] D.A. Drachman, Do we have brain to spare? *Neurology* 64 (12) (2005) 2004–2005, <http://dx.doi.org/10.1212/01.WNL.0000166914.38327.BB>, URL <https://www.neurology.org/doi/10.1212/01.WNL.0000166914.38327.BB>.
- [3] D. Drubach, *The Brain Explained*, Prentice Hall Health, Upper Saddle River, NJ, 2000.
- [4] L. Sokoloff, The metabolism of the central nervous system in vivo, in: *Handbook of Physiology—Neurophysiology*, vol. 3, American Physiological Society, pp. 1843–1864.
- [5] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychol Rev* 65 (6) (1958) 386–408, <http://dx.doi.org/10.1037/h0042519>, URL <https://doi.apa.org/doi/10.1037/h0042519>.
- [6] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature* 323 (6088) (1986) 533–536, <http://dx.doi.org/10.1038/323533a0>, URL <https://www.nature.com/articles/323533a0>.
- [7] K. Fukushima, S. Miyake, Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, in: S. Levin, S.-i. Amari, M.A. Arbib (Eds.), *Competition and Cooperation in Neural Nets*, in: *Lecture Notes in Biomathematics*, vol. 45, Springer Berlin Heidelberg, Berlin, Heidelberg, 1982, pp. 267–285, http://dx.doi.org/10.1007/978-3-642-46466-9_18, URL http://link.springer.com/10.1007/978-3-642-46466-9_18.
- [8] D.H. Hubel, T.N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *J. Physiol.* 195 (1) (1968) 215–243, <http://dx.doi.org/10.1113/jphysiol.1968.sp008455>, URL <https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1968.sp008455>.
- [9] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, in: *The Handbook of Brain Theory and Neural Networks*, MIT Press, 1998, pp. 255–258.
- [10] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Comput. Sci. Rev.* 3 (3) (2009) 127–149, <http://dx.doi.org/10.1016/j.cosrev.2009.03.005>, URL <https://linkinghub.elsevier.com/retrieve/pii/S1574013709000173>.
- [11] M. Tan, Q.V. Le, EfficientNetV2: Smaller models and faster training, 2021, <http://dx.doi.org/10.48550/arXiv.2104.00298>, arXiv:2104.00298 [cs]. URL <http://arxiv.org/abs/2104.00298>.
- [12] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, 2022, <http://dx.doi.org/10.48550/arXiv.2201.03545>, arXiv:2201.03545 [cs]. URL <http://arxiv.org/abs/2201.03545>.
- [13] DeepSeek-AI, D. Guo, D. Yang, DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025, <http://dx.doi.org/10.48550/arXiv.2501.12948>, arXiv:2501.12948 [cs]. URL <http://arxiv.org/abs/2501.12948>.
- [14] G. Comanici, E. Bieber, M. Schaeckermann, Gemini 2.5: Pushing the Frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025, <http://dx.doi.org/10.48550/arXiv.2507.06261>, arXiv:2507.06261 [cs]. URL <http://arxiv.org/abs/2507.06261>.
- [15] DeepMind, Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad. URL <https://deepmind.google/discover/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/>.
- [16] IntuitionLabs, AI Reasoning: Gold-Medal Performance at the 2025 IMO. URL <https://intuitionlabs.ai/articles/ai-reasoning-math-olympiad-imo>.
- [17] M. Stoffel, F. Bamer, B. Markert, Deep convolutional neural networks in structural dynamics under consideration of viscoplastic material behaviour, *Mech. Res. Commun.* 108 (2020) 103565, <http://dx.doi.org/10.1016/j.mechrescom.2020.103565>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0093641320300938>.
- [18] M.B. Gorji, M. Mozaffar, J.N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, *J. Mech. Phys. Solids* 143 (2020) 103972, <http://dx.doi.org/10.1016/j.jmps.2020.103972>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0022509620302076>.
- [19] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (1) (2020) 477–508, <http://dx.doi.org/10.1146/annurev-fluid-010719-060214>, URL <https://www.annualreviews.org/doi/10.1146/annurev-fluid-010719-060214>.
- [20] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, *Science* 367 (6481) (2020) 1026–1030, <http://dx.doi.org/10.1126/science.aaw4741>, URL <https://www.science.org/doi/10.1126/science.aaw4741>.
- [21] R. Gulakala, B. Markert, M. Stoffel, Rapid diagnosis of Covid-19 infections by a progressively growing GAN and CNN optimisation, *Comput. Methods Programs Biomed.* 229 (2023) 107262, <http://dx.doi.org/10.1016/j.cmpb.2022.107262>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0169260722006435>.
- [22] N. Blomeyer, S.B. Tandale, L.F. Nicolini, P. Kobbe, T. Pufe, B. Markert, M. Stoffel, Prediction of temperature and loading history dependent lumbar spine biomechanics under cyclic loading using recurrent neural networks, *Ann. Biomed. Eng.* 51 (6) (2023) 1244–1255, <http://dx.doi.org/10.1007/s10439-022-03128-3>, URL <https://link.springer.com/10.1007/s10439-022-03128-3>.
- [23] R. Gulakala, B. Markert, M. Stoffel, Generative adversarial network based data augmentation for CNN based detection of Covid-19, *Sci. Rep.* 12 (1) (2022) 19186, <http://dx.doi.org/10.1038/s41598-022-23692-x>, URL <https://www.nature.com/articles/s41598-022-23692-x>.
- [24] P.S. Theocaris, P.D. Panagiotopoulos, Plasticity including the Bauschinger effect, studied by a neural network approach, *Acta Mech.* 113 (1–4) (1995) 63–75, <http://dx.doi.org/10.1007/BF01212634>, URL <http://link.springer.com/10.1007/BF01212634>.

- [25] P.S. Theocaris, C. Bisbos, P.D. Panagiotopoulos, On the parameter identification problem for failure criteria in anisotropic bodies, *Acta Mech.* 123 (1–4) (1997) 37–56, <http://dx.doi.org/10.1007/BF01178399>, URL <http://link.springer.com/10.1007/BF01178399>.
- [26] P. Meißner, H. Watschke, J. Winter, T. Vietor, Artificial neural networks-based material parameter identification for numerical simulations of additively manufactured parts by material extrusion, *Polymers* 12 (12) (2020) 2949, <http://dx.doi.org/10.3390/polym12122949>, URL <https://www.mdpi.com/2073-4360/12/12/2949>.
- [27] S.B. Tandale, M. Stoffel, Physics-based self-learning spiking neural network enhanced time-integration scheme for computing viscoplastic structural finite element response, *Comput. Methods Appl. Mech. Engrg.* 422 (2024) 116847, <http://dx.doi.org/10.1016/j.cma.2024.116847>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782524001038>.
- [28] L. Wu, V.D. Nguyen, N.G. Kilinger, L. Noels, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, *Comput. Methods Appl. Mech. Engrg.* 369 (2020) 113234, <http://dx.doi.org/10.1016/j.cma.2020.113234>, URL <https://linkinghub.elsevier.com/retrieve/pii/S00457825163004199>.
- [29] H.J. Logarzo, G. Capuano, J.J. Rimoli, Smart constitutive laws: Inelastic homogenization through machine learning, *Comput. Methods Appl. Mech. Engrg.* 373 (2021) 113482, <http://dx.doi.org/10.1016/j.cma.2020.113482>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782520306678>.
- [30] D.K. Klein, M. Fernández, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, *J. Mech. Phys. Solids* 159 (2022) 104703, <http://dx.doi.org/10.1016/j.jmps.2021.104703>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0022509621003215>.
- [31] J.-H. Bastek, S. Kumar, B. Telgen, R.N. Glaesener, D.M. Kochmann, Inverting the structure–property map of truss metamaterials by deep learning, *Proc. Natl. Acad. Sci.* 119 (1) (2022) e2111505119, <http://dx.doi.org/10.1073/pnas.2111505119>, URL <https://pnas.org/doi/full/10.1073/pnas.2111505119>.
- [32] T. Kirchdoerfer, M. Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 304 (2016) 81–101, <http://dx.doi.org/10.1016/j.cma.2016.02.001>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782516300238>.
- [33] D. González, F. Chinesta, E. Cueto, Thermodynamically consistent data-driven computational mechanics, *Contin. Mech. Thermodyn.* 31 (1) (2019) 239–253, <http://dx.doi.org/10.1007/s00161-018-0677-z>, URL <http://link.springer.com/10.1007/s00161-018-0677-z>.
- [34] R. Gulakala, V. Bhaskaran, M. Stoffel, A generative learning and graph-based framework for computing field variables in Finite Element simulations, *Comput. Methods Appl. Mech. Engrg.* 428 (2024) 117111, <http://dx.doi.org/10.1016/j.cma.2024.117111>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782524003670>.
- [35] X. He, Q. He, J.-S. Chen, Deep autoencoders for physics-constrained data-driven nonlinear materials modeling, *Comput. Methods Appl. Mech. Engrg.* 385 (2021) 114034, <http://dx.doi.org/10.1016/j.cma.2021.114034>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782521003650>.
- [36] M. Stoffel, F. Bamer, B. Markert, Artificial neural networks and intelligent finite elements in non-linear structural mechanics, *Thin-Walled Struct.* 131 (2018) 102–106, <http://dx.doi.org/10.1016/j.tws.2018.06.035>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0263823118303537>.
- [37] S.B. Tandale, B. Markert, M. Stoffel, Smart stiffness computation of one-dimensional Finite Elements, *Mech. Res. Commun.* 119 (2022) 103817, <http://dx.doi.org/10.1016/j.mechrescom.2021.103817>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0093641321001440>.
- [38] G. Capuano, J.J. Rimoli, Smart finite elements: A novel machine learning application, *Comput. Methods Appl. Mech. Engrg.* 345 (2019) 363–381, <http://dx.doi.org/10.1016/j.cma.2018.10.046>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782518305541>.
- [39] S.B. Tandale, B. Markert, M. Stoffel, Physically constrained deep recurrent neural network for stiffness computation of plate structures, *PAMM* 22 (1) (2023) e202200068, <http://dx.doi.org/10.1002/pamm.202200068>, URL <https://onlinelibrary.wiley.com/doi/10.1002/pamm.202200068>.
- [40] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, *Comput. Methods Appl. Mech. Engrg.* 381 (2021) 113852, <http://dx.doi.org/10.1016/j.cma.2021.113852>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782521001894>.
- [41] Z. Wang, J. Estrada, E. Arruda, K. Garikipati, Inference of deformation mechanisms and constitutive response of soft material surrogates of biological tissue by full-field characterization and data-driven variational system identification, *J. Mech. Phys. Solids* 153 (2021) 104474, <http://dx.doi.org/10.1016/j.jmps.2021.104474>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0022509621001459>.
- [42] M. Flaschel, S. Kumar, L. De Lorenzis, Discovering plasticity models without stress data, *Npj Comput. Mater.* 8 (1) (2022) 91, <http://dx.doi.org/10.1038/s41524-022-00752-4>, URL <https://www.nature.com/articles/s41524-022-00752-4>.
- [43] E. Marino, M. Flaschel, S. Kumar, L.D. Lorenzis, Automated identification of linear viscoelastic constitutive laws with EUCLID, *Mech. Mater.* 181 (2023) 104643, <http://dx.doi.org/10.1016/j.mechmat.2023.104643>, arXiv:2212.10969 [cond-mat]. URL <http://arxiv.org/abs/2212.10969>.
- [44] K.F. Pilz, J. Sanders, R. Rahman, L. Heim, Trends in AI Supercomputers, 2025, <http://dx.doi.org/10.48550/arXiv.2504.16026>, arXiv:2504.16026 [cs]. URL <http://arxiv.org/abs/2504.16026>.
- [45] S.M.H. Amiri, P. Goswami, M.M. Islam, M.S. Hossen, S.M.H. Amiri, N. Akter, The carbon cost of conversation, sustainability in the age of language models, 2025, <http://dx.doi.org/10.48550/arXiv.2507.20018>, arXiv:2507.20018 [cs]. URL <http://arxiv.org/abs/2507.20018>.
- [46] N. Jegham, M. Abdelatti, L. Elmoubarki, A. Hendawi, How Hungry is AI? Benchmarking energy, water, and carbon footprint of LLM inference, 2025, <http://dx.doi.org/10.48550/arXiv.2505.09598>, arXiv:2505.09598 [cs]. URL <http://arxiv.org/abs/2505.09598>.
- [47] J. Sacramento, R.P. Costa, Y. Bengio, W. Senn, Dendritic cortical microcircuits approximate the backpropagation algorithm, 2018, <http://dx.doi.org/10.48550/ARXIV.1810.11393>, Version Number: 1. URL <https://arxiv.org/abs/1810.11393>.
- [48] S. Woźniak, A. Pantazi, T. Bohnstingl, E. Eleftheriou, Deep learning incorporating biologically inspired neural dynamics and in-memory computing, *Nat. Mach. Intell.* 2 (6) (2020) 325–336, <http://dx.doi.org/10.1038/s42256-020-0187-0>, URL <https://www.nature.com/articles/s42256-020-0187-0>.
- [49] E.R. Kandel (Ed.), *Principles of Neural Science*, fourth ed., McGraw-Hill, New York, NY, 2000.
- [50] U.S. Bhalla, Molecular computation in neurons: a modeling perspective, *Curr. Opin. Neurobiol.* 25 (2014) 31–37, <http://dx.doi.org/10.1016/j.conb.2013.11.006>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0959438813002171>.
- [51] P. Sterling, S.B. Laughlin, *Principles of Neural Design*, MIT Press, Cambridge, 2017.
- [52] W. Gerstner, W.M. Kistler (Eds.), *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, U.K New York, 2002.
- [53] W. Maass, Networks of spiking neurons: The third generation of neural network models, *Neural Netw.* 10 (9) (1997) 1659–1671, [http://dx.doi.org/10.1016/S0893-6080\(97\)00011-7](http://dx.doi.org/10.1016/S0893-6080(97)00011-7), URL <https://linkinghub.elsevier.com/retrieve/pii/S0893608097000117>.
- [54] M. Davies, N. Srinivasa, T.-H. Lin, A. Wild, Y. Yang, H. Wang, Loihi: A neuromorphic manycore processor with on-chip learning, *IEEE Micro* 38 (1) (2018) 82–99, <http://dx.doi.org/10.1109/MM.2018.112130359>, URL <https://ieeexplore.ieee.org/document/8259423/>.
- [55] M.R. Azghadi, C. Lammie, J.K. Eshraghian, M. Payvand, E. Donati, B. Linares-Barranco, G. Indiveri, Hardware implementation of deep network accelerators towards healthcare and biomedical applications, *IEEE Trans. Biomed. Circuits Syst.* 14 (6) (2020) 1138–1159, <http://dx.doi.org/10.1109/TBCAS.2020.3036081>, URL <https://ieeexplore.ieee.org/document/9250613/>.
- [56] E. Ceolini, C. Frenkel, S.B. Shrestha, G. Taverni, L. Khacaf, M. Payvand, E. Donati, Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing, *Front. Neurosci.* 14 (2020) 637, <http://dx.doi.org/10.3389/fnins.2020.00637>, URL <https://www.frontiersin.org/article/10.3389/fnins.2020.00637/full>.
- [57] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, Y. Tian, Deep residual learning in spiking neural networks, 2021, <http://dx.doi.org/10.48550/ARXIV.2102.04159>, Version Number: 6. URL <https://arxiv.org/abs/2102.04159>.
- [58] K. Patel, E. Hunsberger, S. Batir, C. Eliasmith, A spiking neural network for image segmentation, 2021, <http://dx.doi.org/10.48550/arXiv.2106.08921>, arXiv:2106.08921 [cs]. URL <http://arxiv.org/abs/2106.08921>.

- [59] R. Gulakala, M. Stoffel, A sustainable neuromorphic framework for disease diagnosis using digital medical imaging, *Comput. Methods Programs Biomed. Updat.* 7 (2025) 100171, <http://dx.doi.org/10.1016/j.cmpbup.2024.100171>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2666990024000387>.
- [60] A. Henkes, J.K. Eshraghian, H. Wessels, Spiking neural networks for nonlinear regression, 2022, <http://dx.doi.org/10.48550/arXiv.2210.03515>, arXiv:2210.03515 [cs]. URL <http://arxiv.org/abs/2210.03515>.
- [61] Q. Zhang, A. Kahana, G.E. Karniadakis, P. Stinis, SMS: Spiking marching scheme for efficient long time integration of differential equations, 2022, <http://dx.doi.org/10.48550/arXiv.2211.09928>, arXiv:2211.09928 [math]. URL <http://arxiv.org/abs/2211.09928>.
- [62] S.B. Tandale, M. Stoffel, Spiking recurrent neural networks for neuromorphic computing in nonlinear structural mechanics, *Comput. Methods Appl. Mech. Engrg.* 412 (2023) 116095, <http://dx.doi.org/10.1016/j.cma.2023.116095>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782523002190>.
- [63] A. Voelker, I. Kajić, C. Eliasmith, Legendre memory units: Continuous-time representation in recurrent neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F.d. Alché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 2019.
- [64] M. Stoffel, S.B. Tandale, Spiking neural networks for nonlinear regression of complex transient signals on sustainable neuromorphic processors, *Npj Unconv. Comput.* 1 (1) (2024) 2, <http://dx.doi.org/10.1038/s44335-024-00002-4>, URL <https://www.nature.com/articles/s44335-024-00002-4>.
- [65] Y. Bao, D. Liu, H. Li, A mechanics-informed neural network method for structural modal identification, *Mech. Syst. Signal Process.* 216 (2024) 111458, <http://dx.doi.org/10.1016/j.ymssp.2024.111458>, URL <https://linkinghub.elsevier.com/retrieve/pii/S088832702400356X>.
- [66] V. Polydoros, S. Tandale, M. Stoffel, Physics-based spiking neural network as a surrogate model for viscoplastic material law in impulsively loaded beams, *PAMM* 24 (4) (2024) e202400066, <http://dx.doi.org/10.1002/pamm.202400066>, URL <https://onlinelibrary.wiley.com/doi/10.1002/pamm.202400066>.
- [67] P. M. Ferreira, S. Wang, Y. Gao, A. Benlarbi-Delai, A comparative review of deep and spiking neural networks for edge AI neuromorphic circuits, *Front. Neurosci.* 19 (2025) 1676570, <http://dx.doi.org/10.3389/fnins.2025.1676570>, URL <https://www.frontiersin.org/articles/10.3389/fnins.2025.1676570/full>.
- [68] T. Mei, Y. Zhou, C.Q. Chen, Mechanical neural networks with explicit and robust neurons, *Adv. Sci.* 11 (33) (2024) 2310241, <http://dx.doi.org/10.1002/adv.202310241>, URL <https://advanced.onlinelibrary.wiley.com/doi/10.1002/adv.202310241>.
- [69] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780, <http://dx.doi.org/10.1162/neco.1997.9.8.1735>, URL <https://direct.mit.edu/neco/article/9/8/1735-1780/6109>.
- [70] W. Zhang, H. Geng, P. Li, Composing recurrent spiking neural networks using locally-recurrent motifs and risk-mitigating architectural optimization, *Front. Neurosci.* 18 (2024) 1412559, <http://dx.doi.org/10.3389/fnins.2024.1412559>, URL <https://www.frontiersin.org/articles/10.3389/fnins.2024.1412559/full>.
- [71] S. Xu, J. Zhang, Z. Wang, R. Jiang, R. Yan, H. Tang, ASRC-SNN: Adaptive skip recurrent connection spiking neural network, 2025, <http://dx.doi.org/10.48550/arXiv.2505.11455>, arXiv:2505.11455 [cs]. URL <http://arxiv.org/abs/2505.11455>.
- [72] Y. Yan, Q. Yang, Y. Wu, H. Liu, M. Zhang, H. Li, K.C. Tan, J. Wu, Efficient and robust temporal processing with neural oscillations modulated spiking neural networks, *Nat. Commun.* 16 (1) (2025) 8651, <http://dx.doi.org/10.1038/s41467-025-63771-x>, URL <https://www.nature.com/articles/s41467-025-63771-x>.
- [73] P. Werbos, Backpropagation through time: what it does and how to do it, *Proc. IEEE* 78 (10) (1990) 1550–1560, <http://dx.doi.org/10.1109/5.58337>, URL <http://ieeexplore.ieee.org/document/58337/>.
- [74] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444, <http://dx.doi.org/10.1038/nature14539>, URL <https://www.nature.com/articles/nature14539>.
- [75] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, W. Maass, Long short-term memory and learning-to-learn in networks of spiking neurons, 2018, <http://dx.doi.org/10.48550/arXiv.1803.09574>, arXiv:1803.09574 [cs]. URL <http://arxiv.org/abs/1803.09574>.
- [76] D. Huh, T.J. Sejnowski, Gradient descent for spiking neural networks, 2017, <http://dx.doi.org/10.48550/arXiv.1706.04698>, arXiv:1706.04698 [q-bio]. URL <http://arxiv.org/abs/1706.04698>.
- [77] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, W. Maass, A solution to the learning dilemma for recurrent networks of spiking neurons, *Nat. Commun.* 11 (1) (2020) 3625, <http://dx.doi.org/10.1038/s41467-020-17236-y>, URL <https://www.nature.com/articles/s41467-020-17236-y>.
- [78] T.P. Lillicrap, D. Cownden, D.B. Tweed, C.J. Akerman, Random synaptic feedback weights support error backpropagation for deep learning, *Nat. Commun.* 7 (1) (2016) 13276, <http://dx.doi.org/10.1038/ncomms13276>, URL <https://www.nature.com/articles/ncomms13276>.
- [79] A. Nøkland, Direct feedback alignment provides learning in deep neural networks, 2016, <http://dx.doi.org/10.48550/arXiv.1609.01596>, arXiv:1609.01596 [stat]. URL <http://arxiv.org/abs/1609.01596>.
- [80] C. Clopath, L. Büsing, E. Vasilaki, W. Gerstner, Connectivity reflects coding: a model of voltage-based STDP with homeostasis, *Nature Neurosci.* 13 (3) (2010) 344–352, <http://dx.doi.org/10.1038/nn.2479>, URL <https://www.nature.com/articles/nn.2479>.
- [81] A. Samadi, T.P. Lillicrap, D.B. Tweed, Deep learning with dynamic spiking neurons and fixed feedback weights, *Neural Comput.* 29 (3) (2017) 578–602, http://dx.doi.org/10.1162/NECO_a_00929, URL <https://direct.mit.edu/neco/article/29/3/578-602/8251>.
- [82] S. Bartunov, A. Santoro, B.A. Richards, L. Marris, G.E. Hinton, T. Lillicrap, Assessing the scalability of biologically-motivated deep learning algorithms and architectures, 2018, <http://dx.doi.org/10.48550/arXiv.1807.04587>, arXiv:1807.04587 [cs]. URL <http://arxiv.org/abs/1807.04587>.
- [83] G. Martín-Sánchez, S. Bohté, S. Otte, A taxonomy of recurrent learning rules, 13529 (2022) 478–490, http://dx.doi.org/10.1007/978-3-031-15919-0_40, arXiv:2207.11439 [cs]. URL <http://arxiv.org/abs/2207.11439>.
- [84] W. van der Veen, Including STDP to eligibility propagation in multi-layer recurrent spiking neural networks, 2022, <http://dx.doi.org/10.48550/ARXIV.2201.07602>, Version Number: 1. URL <https://arxiv.org/abs/2201.07602>.
- [85] A. Rostami, B. Vogginger, Y. Yan, C.G. Mayr, E-prop on SpiNNaker 2: Exploring online learning in spiking RNNs on neuromorphic hardware, *Front. Neurosci.* 16 (2022) 1018006, <http://dx.doi.org/10.3389/fnins.2022.1018006>, URL <https://www.frontiersin.org/articles/10.3389/fnins.2022.1018006/full>.
- [86] E. Arnold, Biological Inspired Learning in RSNN on Neuromorphic Hardware (Masterarbeit), Universität Heidelberg, 2021.
- [87] L. Abbott, Lapique's introduction of the integrate-and-fire model neuron (1907), *Brain Res. Bull.* 50 (5–6) (1999) 303–304, [http://dx.doi.org/10.1016/S0361-9230\(99\)00161-6](http://dx.doi.org/10.1016/S0361-9230(99)00161-6), URL <https://linkinghub.elsevier.com/retrieve/pii/S0361923099001616>.
- [88] M. Stoffel, R. Schmidt, D. Weichert, Shock wave-loaded plates, *Int. J. Solids Struct.* 38 (42–43) (2001) 7659–7680, [http://dx.doi.org/10.1016/S0020-7683\(01\)00038-5](http://dx.doi.org/10.1016/S0020-7683(01)00038-5), URL <https://linkinghub.elsevier.com/retrieve/pii/S0020768301000385>.
- [89] M. Stoffel, An experimental method to validate viscoplastic constitutive equations in the dynamic response of plates, *Mech. Mater.* 37 (12) (2005) 1210–1222, <http://dx.doi.org/10.1016/j.mechmat.2005.06.001>, URL <https://linkinghub.elsevier.com/retrieve/pii/S0167666305000797>.
- [90] M. Stoffel, A measurement technique for shock wave-loaded structures and its applications, *Exp. Mech.* 46 (1) (2006) 47–55, <http://dx.doi.org/10.1007/s11340-006-5870-5>, URL <http://link.springer.com/10.1007/s11340-006-5870-5>.
- [91] T. Developers, TensorFlow, 2025, <http://dx.doi.org/10.5281/ZENODO.4724125>, URL <https://zenodo.org/doi/10.5281/zenodo.4724125>.
- [92] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, <http://dx.doi.org/10.48550/arXiv.1412.6980>, arXiv:1412.6980 [cs]. URL <http://arxiv.org/abs/1412.6980>.