Universal approximation theorem for nonlinear resistive networks

Benjamin Scellier[1,*] and Siddhartha Mishra[2]

[1] *Rain AI, San Francisco, USA*
[2] *SAM, D-Math and ETH AI Center, ETH Zurich, Switzerland*

(Received 19 November 2024; accepted 13 February 2025; published 3 April 2025)

This paper is a contribution to the Physical Review Applied collection titled Physics-Inspired Computing.

Resistor networks have recently been studied as analog computing platforms for machine learning, particularly due to their compatibility with the Equilibrium Propagation training framework. In this work, we explore the computational capabilities of these networks. We prove that electrical networks consisting of independent voltage sources, linear resistors, diodes, and voltage-controlled voltage sources (VCVSs) can approximate any continuous function to arbitrary precision. Central to our proof is a method for translating a neural network with rectified linear units into an approximately equivalent electrical network comprising these four elements. Our proof relies on two assumptions: (a) that circuit elements are ideal, and (b) that variable resistor conductances and VCVS amplification factors can take any value (arbitrarily small or large). Our findings provide insights that could guide the development of universal self-learning electrical networks.

DOI: 10.1103/PhysRevApplied.23.044009

I. INTRODUCTION

Machine learning (ML) has become an integral part of modern technology, with neural networks playing a central role in many ML applications. A fundamental characteristic of neural networks is their ability to implement or approximate arbitrary functions: they are *universal function approximators*. However, training and deploying neural networks on graphics processing units (GPUs) is energy-intensive, primarily due to the separation of computation and memory in GPUs, and the digital nature of computing in these processors. As the need for energy-efficient ML systems becomes pressing, researchers explore alternative physical substrates based on analog physics as a replacement for GPU-based neural networks [1]. One key area of research has been the development of analog electrical versions of neural networks, which aim to implement their arithmetic operations—including matrix-vector multiplications and nonlinear activation functions—and the backpropagation algorithm in the analog domain. However, these systems are typically mixed-signal (analog/digital), executing the nonlinear activation functions in the digital domain, and therefore necessitating the use of analog-to-digital converters and digital-to-analog converters [2]. This hurdle has motivated researchers to develop and explore alternative training frameworks, potentially better suited for training analog physical systems [3–8] (for recent reviews, see also [1,9]). However, the physical systems compatible with these training frameworks are typically not isomorphic to neural networks. In particular, unlike neural networks, the computational capabilities and expressivity of these systems remain largely unexplored. Universality has been studied in optical systems [10] (see also [11] where universal linear optical transformations are studied), and a recent study has shown how neural ordinary differential equations can be implemented in analog electrical circuits [12].

We investigate the question of universal function approximation within the context of resistor networks. These electrical networks are composed of variable resistors that serve as trainable weights and voltage sources representing input variables. The steady state of such electrical networks is characterized by the principle of least power dissipation. Although resistor networks have been studied as platforms for analog computing since the 1980s [13], their applications were initially limited to inference tasks. More recently, the equilibrium propagation (EP) framework [3,14] has provided a method to train these networks. EP uses local contrastive learning rules to adjust the resistor conductances to perform gradient descent on a cost function [14,15]. Experimental implementations of resistor networks that self-train using local learning rules closely related to EP have also been realized [16,17], demonstrating the feasibility of this approach. Additionally, an experimental demonstration on a memristor crossbar array [18] suggests potential energy savings of up to four orders of magnitude compared to neural networks trained on GPUs. Importantly, EP is also applicable in nonlinear networks, without requiring detailed knowledge of the nonlinear elements' characteristics (see

*Contact author: benjamin@rain.ai

 © 2025 American Physical Society

Appendix D for details). These and other studies have renewed interest in resistor networks [19–27]. However, the computational expressivity of these networks has remained unexplored.

In this paper, we derive sufficient conditions for resistor networks to be universal function approximators. A recent line of work has shown that nonlinear computation can be achieved even using purely linear physics [7,28,29]; in an analog electrical circuit, this can be achieved by viewing tunable resistances as inputs (see [7] where the method is briefly outlined). Here, however, we consider networks in which inputs are supplied via voltage sources, tunable resistors serve as adjustable parameters, and the voltage response is read at a set of output branches. Clearly, a network of this kind composed solely of ohmic resistors and voltage sources can only implement linear functions. In such networks, achieving nonlinear computation—which is a prerequisite for any sort of universal computation—requires nonlinear elements. Furthermore, in a network with only voltage sources and passive (linear or nonlinear) resistors, the voltage across any branch is bounded by the sum of the voltages from the sources. Following Kendall *et al*. [14], we address these two issues using diodes for nonlinearities, and voltage-controlled voltage sources (VCVSs) for amplification, allowing the implementation of functions with output voltages greater than input voltages. We prove that, under suitable assumptions, these four elements—ohmic resistors, voltage sources, diodes and VCVSs—are sufficient for universal function approximation. The assumptions are that (a) the elements are ideal, and (b) the conductances of variable resistors and the amplification factors of the VCVSs can take arbitrary values (arbitrarily small or arbitrarily large). To prove this result, we consider a specific nonlinear resistive network architecture similar to the one proposed by Kendall *et al*. [14] which we refer to as a "deep resistive network" (DRN), and we show that DRNs can approximate neural networks with rectified linear units (ReLU) to arbitrary accuracy. Since ReLU neural networks are known to be universal function approximators [30,31], it follows that DRNs are universal approximators, too. Our work thus presents the first universal approximation result for a class of analog electrical networks compatible with EP. Additionally, our proof offers a systematic method for translating a ReLU neural network into an equivalent-sized approximate DRN.

We emphasize that, in our construction, DRNs do not *implement* ReLU neural networks but rather *approximate* them, in contrast to the line of research on analog electrical neural networks that use backpropagation for training. Furthermore, although in this work we assume ideal elements to simplify the mathematical analysis of DRNs, EP does not require these assumptions. In fact, EP is agnostic to the characteristics of untrainable components such as diodes (see Appendix D for details).

II. NONLINEAR RESISTIVE NETWORKS

We consider an electrical circuit composed of voltage sources, linear resistors, diodes and VCVSs (Fig. 1). We call such a circuit a "nonlinear resistive network." It implements an input-output function as follows: (a) a set of voltage sources serve as "input variables" where voltages play the role of input values; and (b) a set of branches serve as "output variables" where voltage drops play the role of the model's prediction. Computing with this electrical network proceeds as follows: first, set the input voltage sources to input values; next, let the electrical network reach steady state (the dc operating point of the circuit), then read the output voltages across output branches.

The steady state of the network is the configuration of branch voltages and branch currents that satisfies all the branch equations, as well as Kirchhoff's current law (KCL) at every node, and Kirchhoff's voltage law (KVL) in every loop. We note that some conditions on the network topology and branch characteristics must be met to ensure that there exists a steady state: for instance, if the network contains a loop formed of voltage sources, then the voltage values must add up to zero in order to obey KVL. Subsequently, we will consider circuits that do not contain loops of voltage sources.

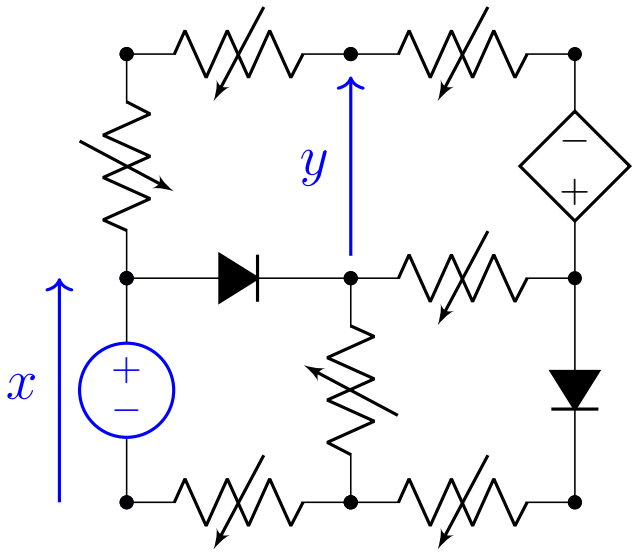


FIG. 1. A nonlinear resistive network composed of variable resistors, voltage sources, diodes and voltage-controlled voltage sources (VCVSs). Voltage sources are used as inputs (x) and voltages across pairs of "output nodes" are used as outputs (y). Variable resistors represent the trainable weights, diodes introduce nonlinearities, and the VCVSs are used for amplification. Such electrical networks are universal function approximators (Theorem 1). Note: for simplicity, the input terminals of the VCVS are not represented on the figure; only its output terminals are displayed.

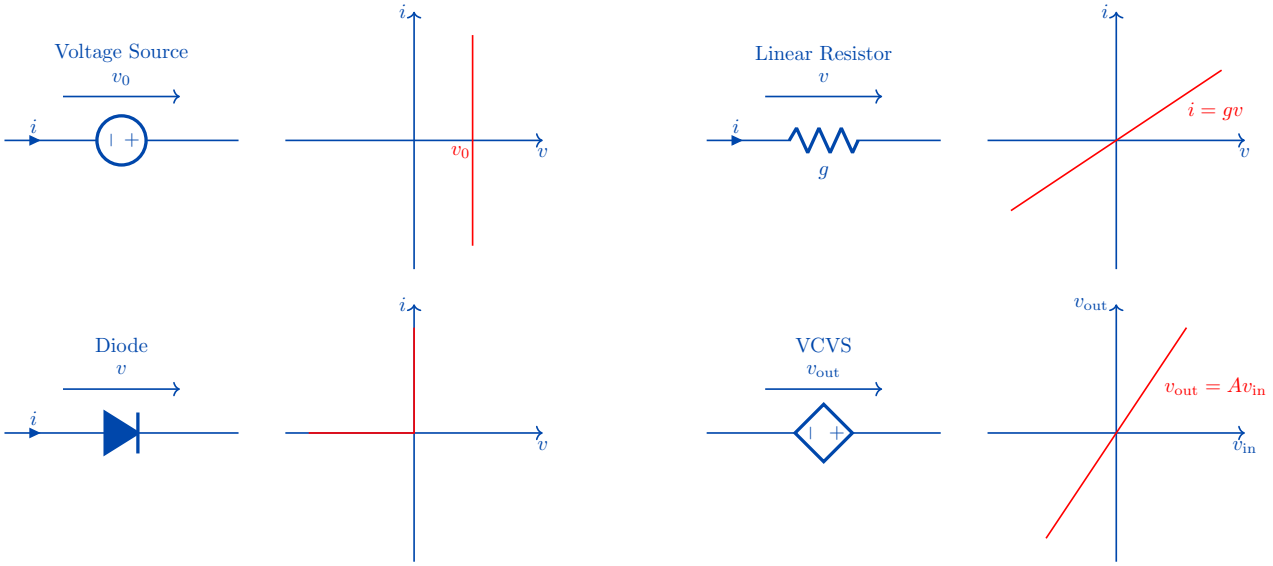


FIG. 2. Current-voltage (i-v) characteristics of ideal elements. A linear resistor is characterized by Ohm's law, $i = gv$, where g is the conductance of the resistor ($g = 1/r$ where r is the resistance). An ideal diode is characterized by $i = 0$ for $v \leq 0$ ("off state," behaving like an open switch), and $v = 0$ for $i > 0$ ("on state," behaving like a closed switch). An ideal voltage source is characterized by $v = v_0$ for some constant voltage v_0. Finally, a voltage-controlled voltage source (VCVS) has four terminals—two terminals that form an input voltage v_{in} and two terminals that form an output voltage v_{out}—and is characterized by the relationship $v_{\text{out}} = A\, v_{\text{in}}$, where $A > 0$ is the amplification factor, or gain. Note: for simplicity, the input terminals of the VCVS are not represented on the figure; only its output terminals are represented.

We assume that the above four circuit elements are *ideal*, meaning that their behavior is determined by the following characteristics (see Fig. 2):

(a) The current-voltage (i-v) characteristic of a voltage source satisfies $v = v_0$ for some constant voltage value v_0, regardless of the current i.

(b) A resistor follows Ohm's law, that is, its i-v characteristic is $i = gv$, where g is the conductance ($g = 1/r$ where r is the resistance).

(c) The i-v characteristic of a diode satisfies $i = 0$ for $v < 0$, and $v = 0$ for $i > 0$.

(d) A VCVS has four terminals: two input terminals whose voltage we denote by v_{in}, and two output terminals whose voltage we denote by v_{out}. It satisfies the relationship $v_{\text{out}} = Av_{\text{in}}$, where $A > 0$ is the "gain" (or "amplification factor").

Under these assumptions, nonlinear resistive networks are universal function approximators in the following sense.

Theorem 1 (A universal approximation theorem for nonlinear resistive networks). Given any continuous function $f : \mathbb{R}^q \to \mathbb{R}^r$, and given any compact subset $C \subset \mathbb{R}^q$ and any $\epsilon > 0$, there exists a nonlinear resistive network with q input voltage sources and r output branches such that, under the above assumptions of ideality, the function $F : \mathbb{R}^q \to \mathbb{R}^r$ that the nonlinear resistive network implements (i.e., the mapping between the inputs and outputs to the network) satisfies

$$\|F(x) - f(x)\| \leq \epsilon, \quad \forall x \in C. \tag{1}$$

The main aim of the next section is to prove Theorem 1.

III. UNIVERSAL APPROXIMATION THROUGH NEURAL NETWORK APPROXIMATION

To prove Theorem 1, we show that nonlinear resistive networks can approximate ReLU neural networks to any desired accuracy. Since ReLU neural networks are universal approximators, it follows that nonlinear resistive networks are universal approximators, too.

We proceed in three steps. First, we recall the definition of a ReLU neural network (Sec. III A) and their universal approximation property (Theorem 2). Then we present a layered nonlinear resistive network model similar to the one proposed by Kendall *et al.* [14] that we call a "deep resistive network" (Sec. III B). Under the assumptions of Sec. II, we derive the equations characterizing the steady state of a DRN (Lemma 3). Finally, using Lemma 3, we prove that any ReLU neural network can be approximated by a DRN (Theorem 5 in Sec. III C). Combining everything, our main Theorem 1 is a consequence of Theorems 2 and 5. Proofs of the lemmas and theorems are provided in Appendix A.

A. ReLU neural network

A neural network processes an input vector x through several transformation stages to produce an output vector. We consider specifically the multilayer perceptron architecture with the rectified linear unit (ReLU) activation function. We refer to this model as the "ReLU neural network."

A ReLU neural network consists of L stages of transformation, called "layers." The input vector is processed sequentially from the input layer (of index $\ell = 0$) to the output layer (of index $\ell = L$), through the hidden layers ($1 \leq \ell \leq L-1$). Each layer ℓ has a dimension M_ℓ, also known as the number of "units" of the layer. Denoting by $s_k^{(\ell)}$ the state of the kth unit of layer ℓ, the equations defining the network's state are as follows:

(1) for the input layer,

$$s_k^{(0)} := x_k, \quad 1 \leq k \leq M_0, \tag{2}$$

where $x = (x_1, \ldots, x_{M_0})$ is the input vector;

(2) for the hidden layers,

$$s_k^{(\ell)} := \max\left(0, \sum_{j=1}^{M_{\ell-1}} w_{jk}^{(\ell)} s_j^{(\ell-1)} + b_k^{(\ell)}\right), \tag{3}$$

for every (ℓ, k) such that $1 \leq \ell \leq L-1$ and $1 \leq k \leq M_\ell$;

(3) for the output layer,

$$s_k^{(L)} := \sum_{j=1}^{M_{L-1}} w_{jk}^{(L)} s_j^{(L-1)} + b_k^{(L)}, \quad 1 \leq k \leq M_L. \tag{4}$$

In these equations, the units' transformations are parameterized by the "weights" $w_{jk}^{(\ell)}$ and "biases" $b_k^{(\ell)}$. The $\max(0, \cdot)$ function is called the "ReLU" nonlinear activation function [32]. The network thus implements a function $G : \mathbb{R}^{M_0} \to \mathbb{R}^{M_L}$ mapping input x to output $s^{(L)}$.

Theorem 2 (ReLU neural networks are universal function approximators). ReLU neural networks can approximate any continuous function on a compact subset. Specifically, for any continuous function $f : \mathbb{R}^q \to \mathbb{R}^r$, compact subset $C \subset \mathbb{R}^q$, and $\epsilon > 0$, there exists a ReLU neural network with $L = 2$ layers, $M_0 = q$ input units, and $M_2 = r$ output units such that

$$\|G(x) - f(x)\| \leq \epsilon, \quad \forall x \in C, \tag{5}$$

where G is the function implemented by the neural network.

This well-known result was first proved by [30] (see also [31]). We will use this result to prove our Theorem 1.

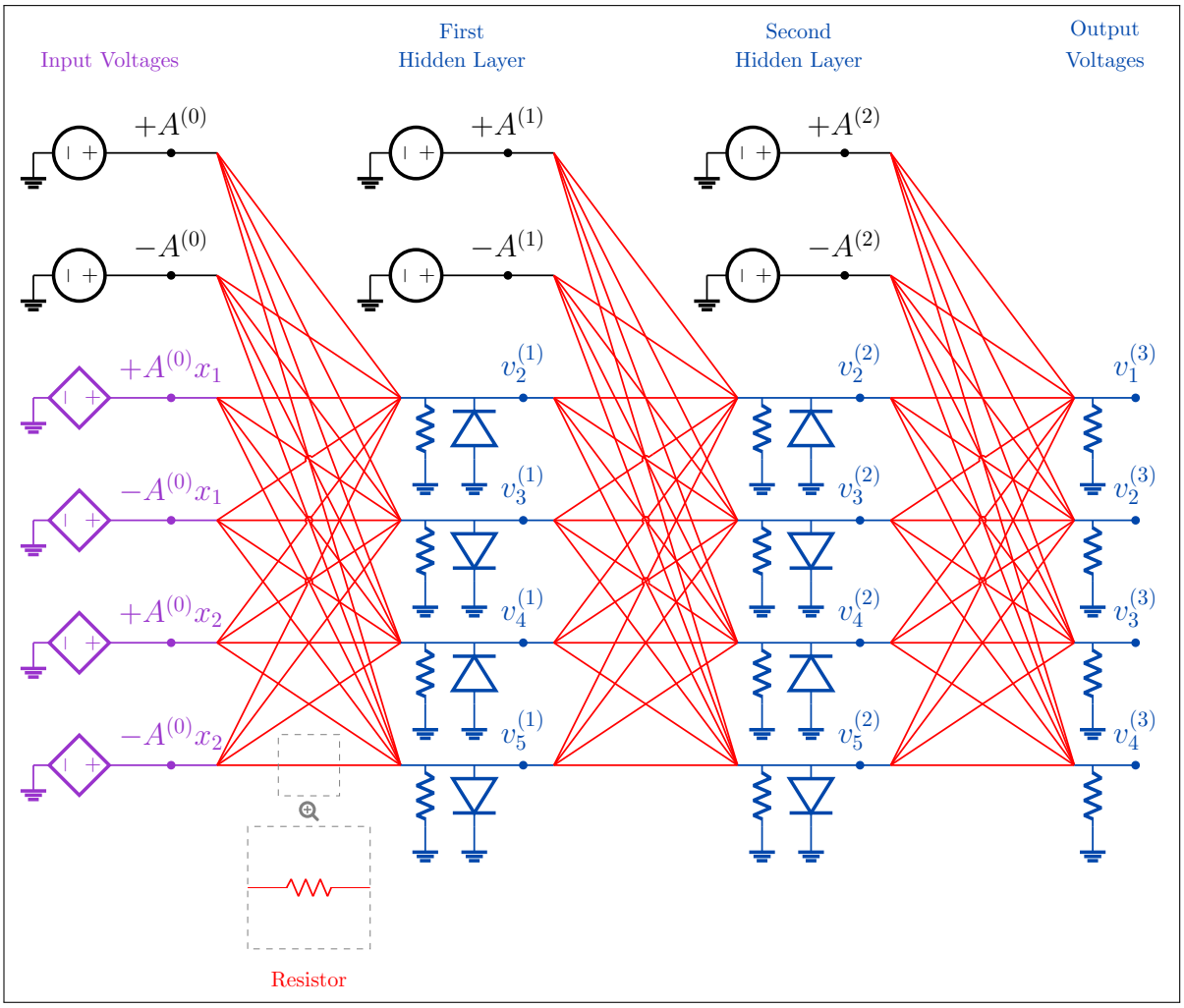
B. Deep resistive network

Next, we present a nonlinear resistive network model similar to that of Kendall *et al.* [14] which takes inspiration from the layered architecture of neural networks. We call this a "deep resistive network." Our goal will be to prove that DRNs can approximate ReLU neural networks to arbitrary accuracy.

In a DRN, the circuit elements—voltage sources, resistors and diodes—are assembled to form a layered network. A DRN is depicted in Fig. 3 and defined as follows. First of all, we choose a reference node called "ground." We denote by L the number of layers in the DRN, and for each ℓ such that $0 \leq \ell \leq L$, we denote by N_ℓ the number of nodes in layer ℓ. We choose $N_0 = 2q + 2$ input nodes and $N_L = r$ output nodes. We also call each node a "unit" by analogy with the units of a neural network. We denote by $v_k^{(\ell)}$ the electrical potential of the kth node of layer ℓ, which we may think of as the unit's activation. Pairs of nodes from two consecutive layers are interconnected by (variable) resistors (i.e., the "trainable weights"). We denote by $g_{jk}^{(\ell)}$ the conductance of the resistor between the j th node of layer $\ell - 1$ and the kth node of layer ℓ. The layer of index $\ell = 0$ is the "input layer," whose units are connected to ground by VCVSs with the same gain $A^{(0)} > 0$. Moreover, there are q voltage sources that play the role of input variables and serve as input voltages to the $2q$ VCVSs. Given an input vector $x = (x_1, \ldots, x_q)$, we set the q input voltage sources to $x_1, \ldots, x_q$, so that the input nodes (VCVS output voltages) have electrical potentials

$$v_{2k}^{(0)} := +A^{(0)} x_k, \; v_{2k+1}^{(0)} := -A^{(0)} x_k, \quad 1 \leq k \leq q. \tag{6}$$

The units of intermediate ("hidden") layers ($1 \leq \ell \leq L-1$) of the network are nonlinear: for each hidden unit, a diode is placed between the unit's node and ground, which can be oriented in either of the two directions. For the units of even index k, the diode points from the unit's node to ground, so the unit's electrical potential is non-negative: when it reaches 0, current flows through the diode from ground to the unit's node so as to maintain the unit's electrical potential at zero; we call such a unit an *excitatory unit*. Conversely, for the units of odd index k, the diode points from ground to the unit's node, so the unit's electrical potential is non-positive: when it reaches 0, the extra current sinks to ground through the diode; we call this an *inhibitory unit*. Thus, half of the hidden units are excitatory and the other half are inhibitory. Output units are linear, meaning that no diode is used for the output units. Finally, each hidden and output unit $v_k^{(\ell)}$ is also linked to ground by a resistor whose conductance is denoted by $g_k^{(\ell)}$. In addition, each of the input and hidden layers contains a pair of "bias units" that are connected to ground by voltage



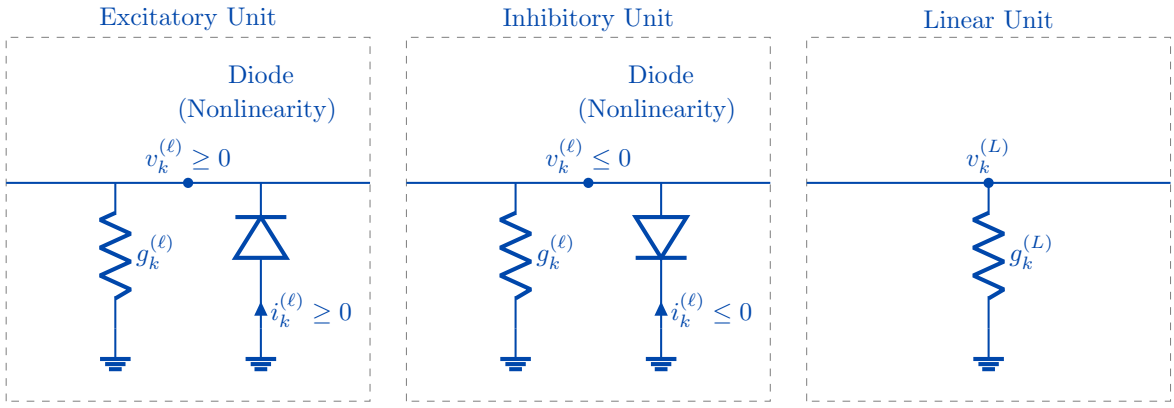


FIG. 3. A deep resistive network (DRN). (Top) A DRN consists of "units" (excitatory, inhibitory, or linear, shown in blue) interconnected by variable resistors (in red). Input voltages (x_1 and x_2) are applied to the network via voltage sources (not shown in the figure for readability) and are amplified using VCVSs to generate the electrical potentials $+A^{(0)}x_1$, $-A^{(0)}x_1$, $+A^{(0)}x_2$, and $-A^{(0)}x_2$ at designated "input nodes" (in purple). The model's prediction is read from a set of output nodes ($v_1^{(3)}$, $v_2^{(3)}$, $v_3^{(3)}$, and $v_4^{(3)}$). Additionally, each layer includes two voltage sources serving as "biases" (shown in black). (Bottom) Each unit is connected to ground through a resistor. Nonlinear units are constructed by placing a diode between the unit's node and ground. Depending on the orientation of the diode, nonlinear units can be either "excitatory" or "inhibitory." Units without a diode are called "linear" units.

sources of fixed voltages:

$$v_0^{(\ell)} := +A^{(\ell)},\ v_1^{(\ell)} := -A^{(\ell)}, \quad 0 \le \ell \le L-1, \tag{7}$$

where $A^{(\ell)} > 0$ a layerwise scalar—the "gain" of layer ℓ.

Under the assumptions of ideality of the elements (Sec. II), the steady state of the DRN—the configuration of branch voltages and branch currents that satisfies all the branch equations, as well as the KCL and KVL—is determined by the following set of equations.

Lemma 1 (Equations of a DRN). The electrical potentials of hidden units satisfy

$$v_k^{(\ell)} = \begin{cases} \max\left(0, p_k^{(\ell)}\right) & \text{if } k \text{ is even (excitatory unit)}, \\ \min\left(0, p_k^{(\ell)}\right) & \text{if } k \text{ is odd (inhibitory unit)}, \end{cases} \tag{8}$$

where

$$p_k^{(\ell)} = \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{9}$$

As for the output units' electrical potentials,

$$v_k^{(L)} = \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} v_j^{(L-1)}}{g_k^{(L)} + \sum_{j=0}^{N_{\ell-1}} g_{jk}^{(L)}}. \tag{10}$$

We prove Lemma 3 in Appendix A 1. A few remarks are in order. First, the equations characterizing the steady state of a DRN share similarities with those of a ReLU neural network [Eqs. (2)–(4)]: these equations involve a weighted sum of the neighboring units' activities (electrical potentials), and the function $\max(0, \cdot)$ is the "ReLU" nonlinear activation function.

However, the equations of a DRN and a ReLU neural network are not identical: a DRN is not a ReLU neural network. One difference is that, in a DRN, the units of a given layer ℓ receive signals not just from the previous layer $\ell - 1$ but also from the next layer $\ell + 1$; in other words, signals flow in both the forward and backward directions.

Another difference is that these signals are normalized by the sum of the weights (total conductance) $g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}$. This results in the following *layerwise maximum principle*, summarized in the following lemma.

Lemma 2 (A maximum principle for DRNs). For every ℓ such that $0 \le \ell \le L$, denote the maximum amplitude of the node electrical potentials in layer ℓ by

$$v_{\max}^{(\ell)} := \max_{2 \le k \le N_\ell - 1} \left| v_k^{(\ell)} \right|. \tag{11}$$

Then we have

$$v_{\max}^{(\ell)} \le \max\left(v_{\max}^{(\ell-1)}, A^{(\ell-1)}, \ldots, A^{(L)}\right), \quad 1 \le \ell \le L. \tag{12}$$

In particular, the electrical potentials of hidden and output units are bounded by those of input and bias units, that is, for every k and ℓ,

$$\left| v_k^{(\ell)} \right| \le \max\left(\max_{2 \le j \le N_0 - 1} \left| v_j^{(0)} \right|, \max_{0 \le \ell \le L-1} \left| A^{(\ell)} \right| \right). \tag{13}$$

We prove Lemma 4 in Appendix A 2. As a consequence of this lemma, the signal strengths tend to decrease as the index of the layer (ℓ) increases. This is the reason why bias units and VCVSs are used: the VCVSs amplify the input signals by a gain $A^{(0)}$ to compensate for the decay in amplitude and ensure that the signals at the output layer do not vanish.

Another difference between DRNs and neural networks is that the "weights" of a DRN are non-negative—a conductance is non-negative—meaning $g_{jk}^{(\ell)} \ge 0$. This is the reason why, in addition to excitatory units, we also use inhibitory units, so the network can communicate "negative signals" too. This is also the reason why we have doubled the number of input VCVSs in the input layer (two VCVSs for each input voltage source).

C. Approximating a ReLU neural network with a DRN

Given a ReLU neural network, we show that we can build a DRN, parametrized by a number $\gamma > 0$, that approximates the neural network up to $O(\gamma)$. To this end, let L be the number of layers of the neural network, and M_ℓ the number of units in layer ℓ, for $0 \le \ell \le L$. We denote by $w_{jk}^{(\ell)}$ and $b_k^{(\ell)}$ the weights and biases. We also denote by $s_k^{(\ell)}$ the state of the kth unit in layer ℓ of the neural network, given an input vector $x = (x_1, \ldots, x_{M_0})$.

We build the DRN as follows. First we define its architecture. The DRN consists of L layers. Input and hidden layers ($0 \le \ell \le L-1$) each contain $N_\ell := 2M_\ell + 2$ units, including M_ℓ excitatory units, M_ℓ inhibitory units and two bias units. The output layer contains $N_L := M_L$ linear units. To fully determine the DRN, it remains to choose the values of the conductances ($g_{jk}^{(\ell)}$), as well as the amplification factors ($A^{(\ell)}$) for input and hidden layers. The key idea of our construction is that for each hidden unit $s_k^{(\ell)}$ of the ReLU neural network, we associate two DRN units $v_{2k}^{(\ell)}$

and $v_{2k+1}^{(\ell)}$, choosing the conductances and amplification factors in such a way that $v_{2k}^{(\ell)} = -v_{2k+1}^{(\ell)} \approx A^{(\ell)} s_k^{(\ell)}$.

To do that, we denote the biases of the neural network for convenience by $w_{0k}^{(\ell)} := b_k^{(\ell)}$. We also introduce a dimensionless number $\gamma > 0$; later we will let $\gamma \to 0$ to establish the equivalence of the DRN and the ReLU neural network. For every ℓ such that $1 \le \ell \le L-1$, given pairs of units $(v_{2j}^{(\ell-1)}, v_{2j+1}^{(\ell-1)})$ and $(v_{2k}^{(\ell)}, v_{2k+1}^{(\ell)})$, we define the conductances of the resistors interconnecting them as follows:

$$g_{2j,2k}^{(\ell)} := \max\left(0, \gamma^\ell w_{jk}^{(\ell)}\right), \tag{14}$$

$$g_{2j,2k+1}^{(\ell)} := \max\left(0, -\gamma^\ell w_{jk}^{(\ell)}\right), \tag{15}$$

$$g_{2j+1,2k}^{(\ell)} := \max\left(0, -\gamma^\ell w_{jk}^{(\ell)}\right), \tag{16}$$

$$g_{2j+1,2k+1}^{(\ell)} := \max\left(0, \gamma^\ell w_{jk}^{(\ell)}\right). \tag{17}$$

Similarly for the output layer, for every (j,k) such that $0 \le j \le M_{L-1} - 1$ and $0 \le k \le M_L - 1$, we define

$$g_{2j,k}^{(L)} := \max\left(0, \gamma^L w_{jk}^{(L)}\right), \tag{18}$$

$$g_{2j+1,k}^{(L)} := \max\left(0, -\gamma^L w_{jk}^{(L)}\right). \tag{19}$$

The conductances thus defined are all non-negative. The purpose of the parameter $\gamma \ll 1$ is to ensure that, at each hidden layer ℓ, the signals coming from the next layer $(\ell+1)$, which scale approximately as $\gamma^{\ell+1}$, are negligible compared to the signals coming from the previous layer $(\ell-1)$, which scale approximately as γ^ℓ. This guarantees that the network effectively operates in a feedforward manner.

It remains to define the layerwise amplification factors and the unitwise leak conductances. For every ℓ such that $1 \le \ell \le L$, we define

$$a^{(\ell)} := \max_{1 \le k \le M_\ell} \left(\sum_{j=0}^{M_{\ell-1}} |w_{jk}^{(\ell)}| \right), \tag{20}$$

and for every (ℓ, k) such that $1 \le \ell \le L$ and $1 \le k \le M_\ell$, we define the "leak conductances" of units $v_{2k}^{(\ell)}$ and $v_{2k+1}^{(\ell)}$ as

$$g_{2k}^{(\ell)} := g_{2k+1}^{(\ell)} := \gamma^\ell \left(a^{(\ell)} - \sum_{j=0}^{M_{\ell-1}} |w_{jk}^{(\ell)}| \right). \tag{21}$$

The conductances $g_{2k}^{(\ell)}$ and $g_{2k+1}^{(\ell)}$ are also non-negative, by definition of $a^{(\ell)}$. Finally, we define the layerwise amplification factors (for input and hidden layers) as

$$A^{(\ell)} := a^{(\ell+1)} \times a^{(\ell+2)} \times \cdots \times a^{(L)}, \quad 0 \le \ell \le L-1. \tag{22}$$

Theorem 3 (Approximation of a ReLU neural network with a DRN). Suppose that, given an input vector $x = (x_1, x_2, \ldots, x_{M_0})$, we set the input voltage sources to

$$v_{2k}^{(0)} := A^{(0)} x_k, \; v_{2k+1}^{(0)} := -A^{(0)} x_k, \quad 1 \le k \le M_0. \tag{23}$$

Denote $\|x\|_\infty := \max_{1 \le k \le M_0} |x_k|$. Then, in the limit $\gamma \to 0$, for every (ℓ, k) such that $1 \le \ell \le L-1$ and $1 \le k \le M_\ell$, we have

$$v_{2k}^{(\ell)} = A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty), \tag{24}$$

$$v_{2k+1}^{(\ell)} = -A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty). \tag{25}$$

For the output layer, we have

$$v_k^{(L)} = s_k^{(L)} + O(\gamma \|x\|_\infty), \quad 1 \le k \le M_L. \tag{26}$$

See Appendix A 3 for a proof. Namely, when $\gamma \to 0$, the states of the excitatory units of the DRN tend to the states of the units of the ReLU neural network rescaled by $A^{(\ell)}$. In particular, the states of output units of the DRN are equal to those of the neural network, up to $O(\gamma \|x\|_\infty)$. In other words, the function implemented by the DRN approximates the function implemented by the ReLU neural network.

Using a symmetry argument, one can prove that the DRN thus built has the property that $v_{2k+1}^{(\ell)} = -v_{2k}^{(\ell)}$ for every (ℓ, k). Although it might seem wasteful and redundant to have two units encode the same piece of information, we recall that doubling the number of units per layer (using excitatory and inhibitory units) is also what allows us to overcome the constraint of non-negative conductances in resistive networks.

Technically, the reason why the DRN behaves essentially like a feedforward (ReLU) network is that we scale the conductances of each layer ℓ by γ^ℓ, using $\gamma \ll 1$. While this trick is mathematically convenient to derive Theorem 5, it would also likely be impractical for deep networks as the range of conductance values will span multiple orders of magnitude, proportional to the number of layers L. It is noteworthy, however, that ReLU neural networks with a single hidden layer ($L = 2$) are also universal approximators (Theorem 2). The DRNs corresponding to such neural networks have a single hidden layer, therefore the range spanned by their conductance values is smaller.

IV. SIMULATIONS

In this section, we illustrate the expressivity of nonlinear resistive networks in simulations. We show that DRNs can

TABLE I. Comparison of deep resistive networks (DRNs) and their equivalent-size ReLU neural networks (ReLU NNs). We train DRNs and ReLU NNs with one, two, and three hidden layers on MNIST, F-MNIST, and K-MNIST. For each experiment, we perform five runs and we report the mean values and std values of the test error rates (in %). See Appendix C for the architectural details of the DRN and ReLU neural network models, as well as the algorithms used to train them.

Dataset	Network Model	Test Error (%)
MNIST	DRN-1H	1.51 ± 0.04
	ReLU NN-1H	1.67 ± 0.03
	DRN-2H	1.46 ± 0.05
	ReLU NN-2H	1.34 ± 0.01
	DRN-3H	1.47 ± 0.08
	ReLU NN-3H	1.48 ± 0.03
Kuzushiji MNIST	DRN-1H	7.67 ± 0.13
	ReLU NN-1H	8.83 ± 0.11
	DRN-2H	8.27 ± 0.02
	ReLU NN-2H	8.14 ± 0.13
	DRN-3H	8.39 ± 0.24
	ReLU NN-3H	7.99 ± 0.23
Fashion MNIST	DRN-1H	10.31 ± 0.05
	ReLU NN-1H	10.20 ± 0.11
	DRN-2H	9.88 ± 0.11
	ReLU NN-2H	9.42 ± 0.22
	DRN-3H	9.79 ± 0.25
	ReLU NN-3H	9.55 ± 0.14

be trained to achieve comparable performance to neural networks on the MNIST, Kuzushiji-MNIST, and Fashion-MNIST datasets. We train DRNs with one, two, and three hidden layers, denoted by DRN-1H, DRN-2H, and DRN-3H, respectively. We compare their performances to their corresponding ReLU neural networks. Each DRN has 1568 input units ($2 \times 28 \times 28$), 10 output units corresponding to the ten classes of each of the three datasets, and 1024 units per hidden layer. Each ReLU neural network has 784 input units (28×28), 10 output units, and 512 units per hidden layer. We use the algorithm of [33] to compute the DRN steady states necessary to extract the weight (conductance) gradients to perform stochastic gradient descent. A full description of the models, algorithms, and hyperparameters used for training is provided in Appendix C. Table I reports the results. Our code to reproduce the results is available at [34].

TABLE II. Conductance values of the DRN models after training (DRN-1H, DRN-2H, and DRN-3H). We report the mean, standard deviation (std), and maximum values of the conductance matrices ($g^{(k)}$) for each of the three DRN models.

		$g^{(1)}$	$g^{(2)}$	$g^{(3)}$	$g^{(4)}$
DRN-1H	mean	0.0077	0.0091		
	std	0.0102	0.0327		
	max	0.2069	0.3295		
DRN-2H	mean	0.0071	0.0074	0.0024	
	std	0.0082	0.0102	0.0249	
	max	0.0996	0.1459	0.4897	
DRN-3H	mean	0.0076	0.0079	0.0053	0.0034
	std	0.0086	0.0113	0.0124	0.0230
	max	0.1276	0.4104	0.8345	0.4125

As discussed earlier, one limiting assumption in our mathematical derivation of Theorem 5 is the requirement to scale the weights (conductances) of layer ℓ by a factor γ^ℓ, where $\gamma \ll 1$. Interestingly, however, our simulation results suggest that this scaling is not necessary in practice. In Table II, we report for each DRN model and for each layer ℓ the mean and standard deviation of the conductance matrix $g^{(\ell)}$ after training. As one can see, the difference in conductance values across layers are not significant, demonstrating that conductance matrices do not actually need to span over multiple orders of magnitude.

V. DISCUSSION

Since the seminal work of Cybenko [35], the concept of universal function approximation has been extensively studied in the context of neural networks. On the other hand, the computational capabilities of analog physical systems that are not isomorphic to neural networks remain largely unexplored, despite recent efforts, for example, in optical systems [10]. In this study, we have examined the expressive power of resistor networks, which use voltage sources as inputs and variable resistors as trainable weights. These networks can be trained by gradient descent using local learning rules [14,15], and prototypical self-learning resistor networks based on such principles have been built [16,17]. Our study advances our understanding of the computational properties of these resistor networks, and further highlights their potential. We have shown that, under suitable assumptions, resistor networks equipped with diodes and voltage-controlled voltage sources —referred to as nonlinear resistive networks— can approximate any continuous function. Central to our proof is the demonstration that a subclass of these electrical networks—so-called "deep resistive networks" —can approximate arbitrary ReLU neural networks with any desired precision. Our result thus offers a method for translating a ReLU neural network into an approximately equivalent DRN.

Nonetheless, our theoretical results rely on several assumptions, some of which are impractical. First, to derive our results, we have assumed ideal diodes, but real-world diodes deviate significantly from our idealized model. Future works could attempt to adapt our approach to networks with more realistic current-voltage (i-v) characteristics, for example, by adapting the steps from Eqs. (A3) to (A6) in our derivation (Appendix A). Second, our construction uses VCVSs with very large gains ($A^{(0)} \gg 1$) to amplify the DRN's input signals,

which would be impractical in an actual circuit. A solution proposed by Kendall *et al*. [14] is to use layerwise "bidirectional amplifiers" with smaller gains, as a replacement for the input VCVSs (see Appendix B for details). Third, we have assumed that the conductances of the variable resistors can be adjusted to arbitrary non-negative values—arbitrarily large or arbitrarily small—but in practice, real-world memristors have quantized conductance values and operate within a finite range. Fourth, our model is deterministic, whereas real-world analog electrical networks are inherently noisy. Despite these assumptions, our universal approximation theorem represents the first theoretical result of its kind for analog resistor networks trainable via EP, and it could serve as a basis for deriving more practical theorems requiring fewer or more realistic assumptions.

We also note that nonlinear resistive networks are closely related to continuous Hopfield networks (CHNs) [36] (for a comparison, see [33, Appendix E]). In particular, the idea of rescaling the weights layerwise, so the DRN behaves approximately as a ReLU neural network, is reminiscent of the work by Xie and Seung [37] where it is shown that layered neural networks can be approximated by layered CHNs using a similar technique. The similarity between nonlinear resistive networks and CHNs is appealing, particularly as recent studies have shown through simulations that CHNs can be effectively trained using equilibrium propagation on datasets like CIFAR10, CIFAR100, and ImageNet 32×32, yielding promising results [38–40].

ACKNOWLEDGMENTS

The authors thank Jack Kendall, Maxence Ernoult, Mohammed Fouda, Suhas Kumar, Vidyesh Anisetti, Andrea Liu, Axel Laborieux and Tim De Ryck for discussions.

APPENDIX A: PROOFS

In this appendix, we prove the theoretical results, apart from Theorem 2 which is well known—we refer to [30] for a proof of it. We proceed as follows. First, we prove Lemma 3. Next, we prove Lemma 4 using Lemma 3. Then, we prove Theorem 5 using Lemmas 3 and 4. Finally, we prove Theorem 1 using Theorems 2 and 5.

1. Proof of Lemma 3

For clarity, we repeat the lemma.

Lemma 3 (Equations of a DRN). The electrical potentials of hidden units satisfy:

$$v_k^{(\ell)} = \begin{cases} \max\left(0, p_k^{(\ell)}\right) & \text{if } k \text{ is even (excitatory unit)}, \\ \min\left(0, p_k^{(\ell)}\right) & \text{if } k \text{ is odd (inhibitory unit)}, \end{cases} \tag{8}$$

where

$$p_k^{(\ell)} = \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{9}$$

As for the output units' electrical potentials,

$$v_k^{(L)} = \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} v_j^{(L-1)}}{g_k^{(L)} + \sum_{j=0}^{N_{\ell-1}} g_{jk}^{(L)}}. \tag{10}$$

Proof of Lemma 3. Consider unit k in layer ℓ, with $0 < \ell \leq L$ and $0 \leq k \leq N_\ell - 1$. First we consider the case $\ell < L$ of a hidden layer (we will consider the case $\ell = L$ of the output layer later). We denote by $i_k^{(\ell)}$ the current flowing through the diode to the unit's node. The current flowing from ground to the unit's node through the resistor is $-g_k^{(\ell)} v_k^{(\ell)}$. We also denote by $i_{jk}^{(\ell)}$ the current flowing from the j th unit of layer $\ell - 1$ to the kth unit of layer ℓ. Kirchhoff's current law applied to the kth unit of layer ℓ tells us that

$$i_k^{(\ell)} - g_k^{(\ell)} v_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} i_{jk}^{(\ell)} - \sum_{j=2}^{N_{\ell+1}-1} i_{kj}^{(\ell+1)} = 0. \tag{A1}$$

Furthermore, by Ohm's law we have $i_{jk}^{(\ell)} = g_{jk}^{(\ell)} \left(v_j^{(\ell-1)} - v_k^{(\ell)}\right)$ and $i_{kj}^{(\ell+1)} = g_{kj}^{(\ell+1)} \left(v_k^{(\ell)} - v_j^{(\ell+1)}\right)$. Therefore

$$i_k^{(\ell)} - g_k^{(\ell)} v_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} \left(v_j^{(\ell-1)} - v_k^{(\ell)}\right) - \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} \left(v_k^{(\ell)} - v_j^{(\ell+1)}\right) = 0. \tag{A2}$$

Solving (A2) for $v_k^{(\ell)}$ yields

$$v_k^{(\ell)} = \frac{i_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{A3}$$

Now we use the characteristics of the diode between the unit's node and ground. First we consider the case of an excitatory unit ($v_k^{(\ell)} \geq 0$). We distinguish between two subcases: either $v_k^{(\ell)} > 0$ and $i_k^{(\ell)} = 0$ (the diode is in the "off state"), or $v_k^{(\ell)} = 0$ and $i_k^{(\ell)} \geq 0$ (the diode is in the "on

state"). In the first subcase (off state) we have, using (A3),

$$0 < v_k^{(\ell)} = \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{A4}$$

In the second subcase (on state) we have, using (A3) again,

$$0 = v_k^{(\ell)} \geq \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{A5}$$

From (A4) and (A5) it follows that both subcases are captured by a single formula:

$$v_k^{(\ell)} = \max\left(0, \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}\right). \tag{A6}$$

Similarly, an inhibitory unit satisfies

$$v_k^{(\ell)} = \min\left(0, \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}\right). \tag{A7}$$

The case of output nodes ($\ell = L$) is identical, with $N_{L+1} = 0$ (meaning the layer of index $L+1$ is empty) and $i_k^{(L)} = 0$ (no diode). Equation (A3) in this case is rewritten as

$$v_k^{(L)} = \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} v_j^{(L-1)}}{g_k^{(L)} + \sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)}}. \tag{A8}$$

■

2. Proof of Lemma 4

Lemma 4 (A maximum principle for DRNs). For every ℓ such that $0 \leq \ell \leq L$, denote the maximum amplitude of the node electrical potentials in layer ℓ by

$$v_{\max}^{(\ell)} := \max_{2 \leq k \leq N_\ell - 1} \left|v_k^{(\ell)}\right|. \tag{11}$$

Then we have

$$v_{\max}^{(\ell)} \leq \max\left(v_{\max}^{(\ell-1)}, A^{(\ell-1)}, \ldots, A^{(L)}\right), \quad 1 \leq \ell \leq L. \tag{12}$$

In particular, the electrical potentials of hidden and output units are bounded by those of input and bias units, that is, for every k and ℓ,

$$\left|v_k^{(\ell)}\right| \leq \max\left(\max_{2 \leq j \leq N_0 - 1} \left|v_j^{(0)}\right|, \max_{0 \leq \ell \leq L-1} \left|A^{(\ell)}\right|\right). \tag{13}$$

Proof of Lemma 4. To prove (12), we proceed by induction on the layer index ℓ, starting from the output layer ($\ell = L$), down to the first hidden layer ($\ell = 1$). We start with $\ell = L$. Using Lemma 3 we have for every output unit k,

$$|v_k^{(L)}| \leq \left|\frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} v_j^{(L-1)}}{g_k^{(L)} + \sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)}}\right| \tag{A9}$$

$$\leq \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} |v_j^{(L-1)}|}{g_k^{(L)} + \sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)}} \tag{A10}$$

$$\leq \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} |v_j^{(L-1)}|}{\sum_{j=1}^{N_{L-1}-1} g_{jk}^{(L)}} \tag{A11}$$

$$\leq \frac{\sum_{j=0}^{1} g_{jk}^{(L)} A^{(L-1)} + \sum_{j=2}^{N_{L-1}-1} g_{jk}^{(L)} v_{\max}^{(L-1)}}{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)}} \tag{A12}$$

$$\leq \max\left(v_{\max}^{(L-1)}, A^{(L-1)}\right). \tag{A13}$$

Since this holds for every output unit ($1 \leq k \leq N_L$), it follows that

$$v_{\max}^{(L)} \leq \max\left(v_{\max}^{(L-1)}, A^{(L-1)}\right). \tag{A14}$$

This completes the initialization step of the proof by induction.

Suppose now that (12) holds for layer $\ell + 1$ where $2 \leq \ell + 1 \leq L$, that is,

$$v_{\max}^{(\ell+1)} \leq \max\left(v_{\max}^{(\ell)}, A^{(\ell)}, \ldots, A^{(L-1)}\right), \tag{A15}$$

and let us show that the property holds for layer ℓ. To show this, we distinguish between two cases: either $v_{\max}^{(\ell)} \leq \max\left(A^{(\ell)}, \ldots, A^{(L-1)}\right)$ or $v_{\max}^{(\ell)} > \max\left(A^{(\ell)}, \ldots, A^{(L-1)}\right)$. The first case directly leads to $v_{\max}^{(\ell)} \leq \max\left(v_{\max}^{(\ell-1)}, A^{(\ell-1)}, \ldots, A^{(L-1)}\right)$, as desired. The second case implies that

$$v_{\max}^{(\ell+1)} \leq v_{\max}^{(\ell)}. \tag{A16}$$

Next, using Lemma 3 again,

$$|v_k^{(\ell)}| \leq \left|\frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}\right| \tag{A17}$$

$$\leq \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} |v_j^{(\ell-1)}| + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} |v_j^{(\ell+1)}|}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}} \tag{A18}$$

$$\leq \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} |v_j^{(\ell-1)}| + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} |v_j^{(\ell+1)}|}{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}} \tag{A19}$$

$$\leq \frac{\sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_{\max}^{(\ell+1)}}{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}} \tag{A20}$$

$$\leq \frac{\sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_{\max}^{(\ell)}}{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}, \tag{A21}$$

where we have used Eq. (A16). Next we select the index k ($2 \leq k \leq N_\ell - 1$) that achieves the largest magnitude, that is, $|v_k^{(\ell)}| = v_{\max}^{(\ell)}$. We get

$$v_{\max}^{(\ell)} \leq \frac{\sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_{\max}^{(\ell)}}{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}. \tag{A22}$$

Rearranging the terms, we successively obtain

$$\left(\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} \right) v_{\max}^{(\ell)} \leq \sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} v_{\max}^{(\ell)}, \tag{A23}$$

$$\left(\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} \right) v_{\max}^{(\ell)} \leq \sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)}, \tag{A24}$$

$$v_{\max}^{(\ell)} \leq \frac{\sum_{j=0}^{1} g_{jk}^{(\ell)} A^{(\ell-1)} + \sum_{j=2}^{N_{\ell-1}-1} g_{jk}^{(\ell)} v_{\max}^{(\ell-1)}}{\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)}} \leq \max\left(A^{(\ell-1)}, v_{\max}^{(\ell-1)}\right), \tag{A25}$$

where we have used that a weighted mean of terms with positive weights is bounded above by the max of the terms. Again this leads to

$$v_{\max}^{(\ell)} \leq \max\left(v_{\max}^{(\ell-1)}, A^{(\ell-1)}, \ldots, A^{(L-1)}\right), \tag{A26}$$

as desired. This completes the induction step, and therefore completes the proof. ■

3. Proof of Theorem 5

Theorem 5 (Approximation of a ReLU neural network with a DRN). Suppose that, given an input vector $x = (x_1, x_2, \ldots, x_{M_0})$, we set the input voltage sources to

$$v_{2k}^{(0)} := A^{(0)} x_k, \; v_{2k+1}^{(0)} := -A^{(0)} x_k, \quad 1 \leq k \leq M_0. \tag{23}$$

Denote $\|x\|_\infty := \max_{1 \leq k \leq M_0} |x_k|$. Then, in the limit $\gamma \to 0$, for every (ℓ, k) such that $1 \leq \ell \leq L-1$ and $1 \leq k \leq$

M_ℓ, we have

$$v_{2k}^{(\ell)} = A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty), \tag{24}$$

$$v_{2k+1}^{(\ell)} = -A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty). \tag{25}$$

For the output layer, we have

$$v_k^{(L)} = s_k^{(L)} + O(\gamma \|x\|_\infty), \quad 1 \le k \le M_L. \tag{26}$$

Proof of Theorem 5. We prove property (24) by induction on ℓ. It is true for $\ell = 0$ thanks to (23). Suppose that (24) is true for some $\ell - 1 \ge 0$ and let us prove it at the rank ℓ. Let k such that $1 \le k \le M_\ell$, and consider the excitatory unit $v_{2k}^{(\ell)}$ (the case of inhibitory unit $v_{2k+1}^{(\ell)}$, and the case of linear output unit $v_k^{(L)}$ if $\ell = L$, are similar). Recall from Lemma 3 that

$$v_{2k}^{(\ell)} = \max\left(0, \frac{\sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{2k,j}^{(\ell+1)} v_j^{(\ell+1)}}{g_{2k}^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{2k,j}^{(\ell+1)}}\right). \tag{A27}$$

First we calculate the denominator of expression (A27). By distinguishing between the two cases $w_{jk}^{(\ell)} \ge 0$ and $w_{jk}^{(\ell)} \le 0$, it is easy to see that we have in both cases

$$g_{2j,2k}^{(\ell)} + g_{2j+1,2k}^{(\ell)} = \gamma^\ell |w_{jk}^{(\ell)}|. \tag{A28}$$

Summing from $j = 0$ to $j = M_{\ell-1}$, we get

$$\sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)} = \gamma^\ell \sum_{j=0}^{M_{\ell-1}} |w_{jk}^{(\ell)}|. \tag{A29}$$

Recalling that, by definition, $g_{2k}^{(\ell)} := \gamma^\ell \left(a^{(\ell)} - \sum_{j=0}^{M_{\ell-1}} |w_{jk}^{(\ell)}|\right)$, it follows that

$$g_{2k}^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)} = \gamma^\ell a^{(\ell)}. \tag{A30}$$

Thus, we get the following formula for the denominator in (A27):

$$\underbrace{g_{2k}^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)}}_{=\gamma^\ell a^{(\ell)}} + \underbrace{\sum_{j=2}^{N_{\ell+1}-1} g_{2k,j}^{(\ell+1)}}_{=O(\gamma^{\ell+1})} = \gamma^\ell a^{(\ell)} + O(\gamma^{\ell+1}). \tag{A31}$$

Next, we calculate the numerator of (A27). Similarly, whether $w_{jk}^{(\ell)} \ge 0$ or $w_{jk}^{(\ell)} \le 0$, we have in both cases, using the inductive hypothesis,

$$g_{2j,2k}^{(\ell)} v_{2j}^{(\ell-1)} + g_{2j+1,2k}^{(\ell)} v_{2j+1}^{(\ell-1)} = \gamma^\ell A^{(\ell-1)} w_{jk}^{(\ell)} s_j^{(\ell-1)} + O(\gamma^{\ell+1} \|x\|_\infty). \tag{A32}$$

This identity also holds for $j = 0$ by defining $s_0^{(\ell-1)} := 1$ to include the case of the biases. Summing from $j = 0$ to $j = M_{\ell-1}$, we get

$$\sum_{j=0}^{N_{\ell-1}-1} g_{j,2k}^{(\ell)} v_j^{(\ell-1)} + \underbrace{\sum_{j=2}^{N_{\ell+1}-1} g_{2k,j}^{(\ell+1)} v_j^{(\ell+1)}}_{=O(\gamma^{\ell+1}\|x\|_\infty) \text{ by Lemma 4}} = \gamma^\ell A^{(\ell-1)} \left(\sum_{j=0}^{M_{\ell-1}} w_{jk}^{(\ell)} s_j^{(\ell-1)}\right) + O(\gamma^{\ell+1} \|x\|_\infty). \tag{A33}$$

Here we have used the maximum principle (Lemma 4) to derive the bound $O(\gamma^{\ell+1}\|x\|_\infty)$ for the $v_j^{(\ell+1)}$ term. Finally, substituting (A31) and (A33) into (A27), we obtain the following equation for $v_{2k}^{(\ell)}$:

$$v_{2k}^{(\ell)} = \max\left(0, \frac{A^{(\ell-1)}}{a^{(\ell)}} \left(\sum_{j=1}^{M_{\ell-1}} w_{jk}^{(\ell)} s_j^{(\ell-1)} + b_k^{(\ell)}\right)\right) + O(\gamma \|x\|_\infty) = A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty), \tag{A34}$$

where we have used that $A^{(\ell-1)} = a^{(\ell)} A^{(\ell)}$, by definition. The case of $v_{2k+1}^{(\ell)}$ (inhibitory unit) is similar: we obtain

$$v_{2k+1}^{(\ell)} = -A^{(\ell)} s_k^{(\ell)} + O(\gamma \|x\|_\infty). \tag{A35}$$

Hence, property (24) is true at the rank ℓ. (The specific case of the linear output layer can be treated similarly.) This completes the proof by induction. ■

4. Proof of Theorem 1

Theorem 1 (A universal approximation theorem for nonlinear resistive networks). Given any continuous function $f : \mathbb{R}^q \to \mathbb{R}^r$, and given any compact subset $C \subset \mathbb{R}^q$ and any $\epsilon > 0$, there exists a nonlinear resistive network with q input voltage sources and r output branches such that, under the above assumptions of ideality, the function $F : \mathbb{R}^q \to \mathbb{R}^r$ that the nonlinear resistive network implements (i.e., the mapping between the inputs and outputs to the network) satisfies

$$\|F(x) - f(x)\| \le \epsilon, \quad \forall x \in C. \tag{1}$$

Proof of Theorem 1. Let $f : \mathbb{R}^q \to \mathbb{R}^r$ be a continuous function. Let C be a compact subset of $\mathbb{R}^q$, and let $\epsilon > 0$.

By Theorem 2, there exists a ReLU neural network $\mathcal{N}$ with q inputs and r outputs such that the function $G_{\mathcal{N}}$ that it implements satisfies

$$\|G_{\mathcal{N}}(x) - f(x)\| \leq \frac{\epsilon}{2}, \quad \forall x \in C. \tag{A36}$$

Let $\mathcal{R}_\gamma$ be the DRN approximator of $\mathcal{N}$ with parameter γ, as defined in Sec. III C. By Theorem 5, the function $F_{\mathcal{R}_\gamma}$ implemented by $\mathcal{R}_\gamma$, satisfies

$$\|F_{\mathcal{R}_\gamma}(x) - G_{\mathcal{N}}(x)\| = O(\gamma \|x\|_\infty); \tag{A37}$$

thus, there exists $\gamma > 0$ such that

$$\|F_{\mathcal{R}_\gamma}(x) - G_{\mathcal{N}}(x)\| \leq \frac{\epsilon}{2}, \quad \forall x \in C. \tag{A38}$$

Combining the two inequalities, we obtain

$$\begin{aligned}\|F_{\mathcal{R}}(x) - f(x)\| &\leq \|F_{\mathcal{R}}(x) - G_{\mathcal{N}}(x)\| \\ &\quad + \|G_{\mathcal{N}}(x) - f(x)\| \leq \epsilon, \quad \forall x \in C.\end{aligned} \tag{A39}$$

Hence the result. ■

APPENDIX B: BIDIRECTIONAL AMPLIFIERS

One constraint of deep resistive networks is the decay in amplitude across the layers of the network, as the depth increases, and therefore the need to amplify the input voltages by a large gain $A^{(0)} \gg 1$ to compensate for this decay. Instead of amplifying input voltages by a large factor $A^{(0)}$, another solution proposed by Kendall *et al.* [14] is to equip each "unit" (or node) with a *bidirectional amplifier*.

A bidirectional amplifier is a three-terminal device, with bottom (B), left (L), and right (R) terminals. The bottom terminal is linked to ground. The current and voltage states (v_L, i_L) and (v_R, i_R) of the left and right terminals satisfy $v_R = a\, v_L$ and $i_R = a\, i_L$, for some positive constant a, the *gain* of the bidirectional amplifier (see Fig. 4). Voltages are amplified in the forward direction by a factor $a > 1$, and currents are amplified in the backward direction by a factor $1/a$. In practice, a bidirectional amplifier can be formed by assembling a voltage-controlled voltage source and a current-controlled current source .

We then equip each unit with a bidirectional amplifier, placed before the resistor and the diode (Fig. 4). We define the unit's state as the voltage after amplification. All the units of a given layer ℓ have the same gain $a^{(\ell)}$ as defined by (20). Lemma 3 can be restated as follows: for every (ℓ, k) such that $1 \leq \ell \leq L-1$ and $0 \leq k \leq N_\ell - 1$, the equation for $v_k^{(\ell)}$ is

$$v_k^{(\ell)} = \max\left(0, \frac{\displaystyle\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} a^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} \frac{v_j^{(\ell+1)}}{a^{(\ell+1)}}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}\right)$$

if k is even (the unit is excitatory),

$$v_k^{(\ell)} = \min\left(0, \frac{\displaystyle\sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} a^{(\ell)} v_j^{(\ell-1)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)} \frac{v_j^{(\ell+1)}}{a^{(\ell+1)}}}{g_k^{(\ell)} + \sum_{j=0}^{N_{\ell-1}-1} g_{jk}^{(\ell)} + \sum_{j=2}^{N_{\ell+1}-1} g_{kj}^{(\ell+1)}}\right)$$

if k is odd (the unit is inhibitory). (B1)

As for the equation of the output unit $v_k^{(L)}$,

$$v_k^{(L)} = \frac{\sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)} a^{(L)} v_j^{(L-1)}}{g_k^{(L)} + \sum_{j=0}^{N_{L-1}-1} g_{jk}^{(L)}}, \quad 0 \leq k \leq N_L - 1. \tag{B2}$$

In these expressions, $((v_j^{(\ell+1)})/(a^{(\ell+1)}))$ represents the voltage of the jth unit of layer $\ell+1$ *before* amplification.

Lemma 4 must also be adapted: with bidirectional amplifiers, the bounds depend not only on the boundary conditions (input and bias voltages) but also on the layer-wise gains of the bidirectional amplifiers ($a^{(\ell)}$). For simplicity, let us assume no bias voltages. Then the equation for $v_{\max}^{(\ell)} := \max_{2 \leq j \leq N_\ell - 1} |v_j^{(\ell)}|$ takes the form

$$v_{\max}^{(\ell)} \leq a^{(\ell)}\, v_{\max}^{(\ell-1)}, \quad 1 \leq \ell \leq L. \tag{B3}$$

Multiplying by $A^{(\ell)} = a^{(\ell+1)} \times \cdots \times a^{(L)}$ on both sides, we get

$$A^{(\ell)} v_{\max}^{(\ell)} \leq A^{(\ell-1)}\, v_{\max}^{(\ell-1)}, \quad 1 \leq \ell \leq L. \tag{B4}$$

Iterating, we obtain

$$\begin{aligned}A^{(\ell)} |v_k^{(\ell)}| &\leq A^{(0)} \max_{2 \leq j \leq N_0 - 1} |v_j^{(0)}|, \\ &0 \leq \ell \leq L,\ 2 \leq k \leq N_\ell - 1.\end{aligned} \tag{B5}$$

APPENDIX C: SIMULATION DETAILS

In this appendix, we provide the details of our simulations (Sec. IV). We train DRNs with one, two, and three hidden layers, and we compare their performances to their corresponding ReLU neural networks with one, two, and three hidden layers. Our code to reproduce the results is available at [34].

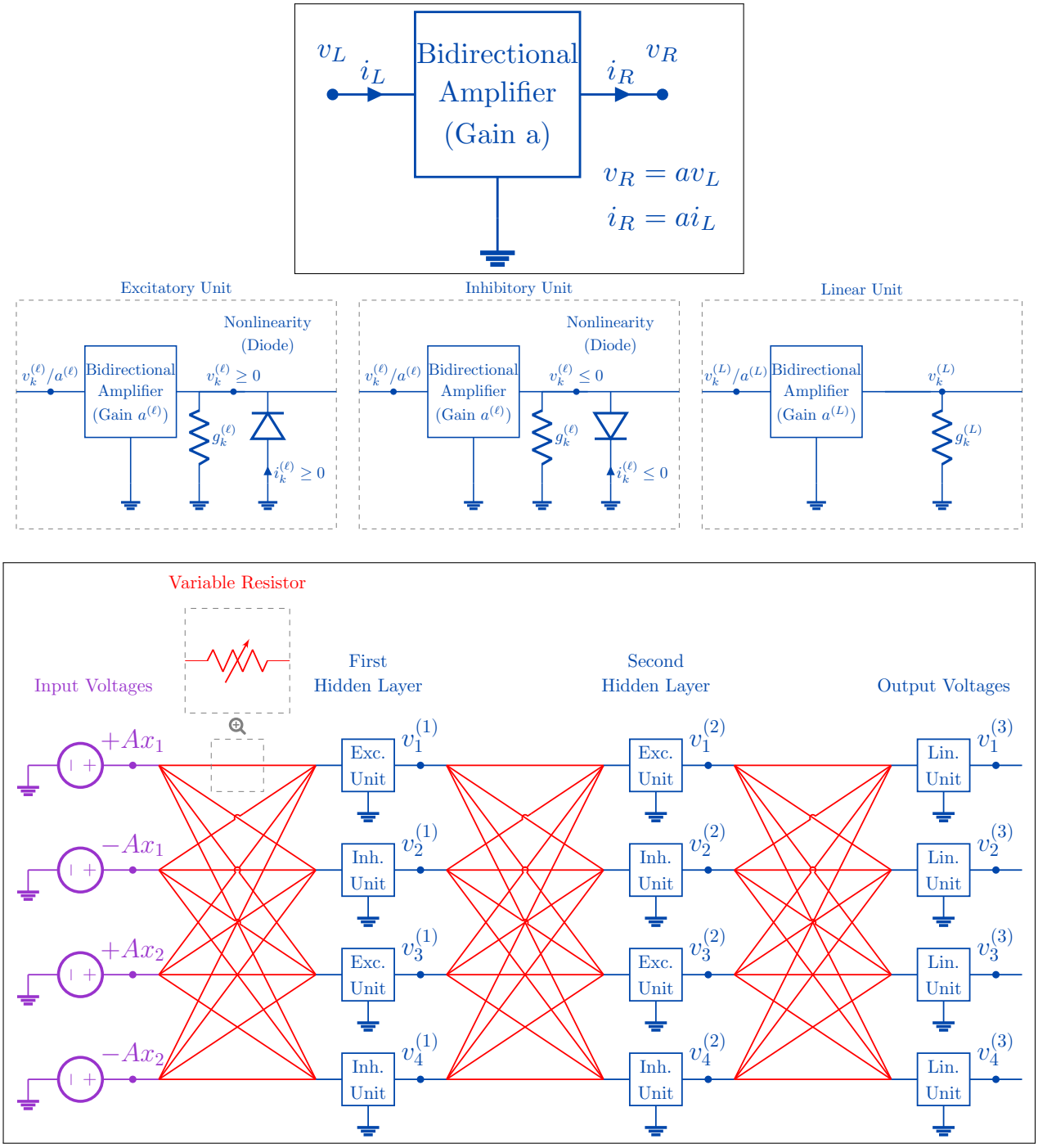


FIG. 4. (Top) Bidirectional amplifier with gain a. (Middle) A unit is composed of a bidirectional amplifier (with amplification factor a), followed by a resistor and a diode between the unit's node and ground. Depending on the orientation of the diode, units come in two flavors: excitatory units and inhibitory units. (Bottom) A deep resistive network with bidirectional amplifiers.

1. Datasets

We perform simulations on the MNIST, Kuzushiji-MNIST, and Fashion-MNIST datasets.

The MNIST dataset [41] consists of 60 000 training images and 10 000 test images. Each image is a 28×28 pixel grayscale representation of a handwritten digit ranging from 0 to 9. Each image is associated with a label indicating the digit it represents.

The Kuzushiji-MNIST dataset [42] is a dropin replacement for the MNIST dataset, also consisting of 60 000 training images and 10 000 test images, each a 28×28 pixel grayscale image. The images represent handwritten

kuzushiji (cursive Japanese) hiragana characters. Like MNIST, each image in the Kuzushiji-MNIST dataset comes with an associated label, corresponding to Japanese characters.

The Fashion-MNIST dataset [43] also features 60 000 training images and 10 000 test images. These images are also 28×28 pixels in grayscale, depicting various fashion items categorized into ten types such as shirts, trousers, and shoes.

2. Optimizer and scheduler

We optimize the mean squared error. We perform stochastic gradient descent with momentum $\mu = 0.9$. No weight decay is used. We also use an "exponential learning rate scheduler" with a decay rate of 0.99 for the learning rates at each epoch of training. We use mini batches of size 32 and we perform 100 epochs of training.

3. Computational resources

Our code for the simulations uses PYTORCH 1.13.1 and TORCHVISION 0.14.1. [44]. The simulations were carried on three NVidia A100 GPUs (one for each of the three sets of simulations corresponding to the three datasets). Each run took between 1 and 5 h to complete, depending on the network type (DRN or ReLU neural network) and network architecture (1H, 2H, or 3H).

4. Simulation details of DRNs

Table III contains the hyperparameters used to obtain the results presented in Table I.

a. DRN architecture

We train deep resistive networks with one, two, and three hidden layers, termed DRN-1H, DRN-2H, and DRN-3H, respectively. Each DRN has 1568 input units ($2 \times 28 \times 28$), 10 output units and 1024 units per hidden layer. In our implementation for the simulations, we did not include resistors between the units' nodes and ground (parallel to the diodes).

b. Input amplification factor

Input voltages are amplified by a (nontrainable) gain factor $A^{(0)}$. The value of $A^{(0)}$ is reported in Table III for each DRN architecture (DRN-1H, DRN-2H, and DRN-3H).

c. Initialization of the conductances of the resistors

Given two consecutive layers of indices ℓ and $\ell+1$, denoting by N_ℓ the number of units in layer ℓ, we initialize the matrix $g^{(\ell+1)}$ of conductances between these two layers as

$$g_{jk}^{(\ell+1)} = \max\left(0, h_{jk}^{(\ell+1)}\right),$$
$$h_{jk}^{(\ell+1)} \sim \mathcal{U}(-c, +c),\ c := \sqrt{\frac{1}{N_\ell}}. \tag{C1}$$

d. Computing the steady state

We compute the steady state of the DRN with the following algorithm, introduced in [33]. First we set the state of input units ($\ell = 0$) to their input values, and we initialize the state of the hidden layers and output layer to zero. Then, for $\ell = 1$ to $\ell = L-1$, we perform the following operations. For every k such that $1 \le k \le N_\ell$, we update the state of unit $v_k^{(\ell)}$ as follows:

$$v_k^{(\ell)} \leftarrow \max\left(0, p_k^{(\ell)}\right) \quad \text{if } k \text{ is even},$$
$$v_k^{(\ell)} \leftarrow \min\left(0, p_k^{(\ell)}\right) \quad \text{if } k \text{ is odd},$$
$$\text{where } p_k^{(\ell)} := \frac{\sum_{j=1}^{N_{\ell-1}} g_{jk}^{(\ell)} v_j^{(\ell-1)} + \sum_{j=1}^{N_{\ell+1}} g_{kj}^{(\ell+1)} v_j^{(\ell+1)}}{\sum_{j=1}^{N_{\ell-1}} g_{jk}^{(\ell)} + \sum_{j=1}^{N_{\ell+1}} g_{kj}^{(\ell+1)}}. \tag{C2}$$

For the output layer ($\ell = L$), the update rule for the states of the output units $v_k^{(L)}$ is

$$v_k^{(L)} \leftarrow \frac{\sum_{j=1}^{N_{\ell-1}} g_{jk}^{(L)} v_j^{(L-1)}}{\sum_{j=1}^{N_{\ell-1}} g_{jk}^{(L)}}. \tag{C3}$$

TABLE III. Hyperparameters used for initializing and training the deep resistive networks (DRNs) and reproducing the results of Table I. LR means learning rate.

	DRN-1H	DRN-2H	DRN-3H
Input amplification factor ($A^{(0)}$)	480	2000	4000
No. of iterations at inference (T)	4	5	6
No. of iterations during training (K)	4	5	6
LR weight 1 & bias 1 (η_1)	0.006	0.002	0.005
LR weight 2 & bias 2 (η_2)	0.006	0.006	0.02
LR weight 3 & bias 3 (η_3)		0.005	0.08
LR weight 4 & bias 4 (η_4)			0.005
Nudging parameter (β), used in Table IV	0.5	1.0	2.0

Importantly, we may update all the units of a given layer simultaneously, rather than sequentially, thus taking advantage of the parallelism of GPUs. Updating all the layers once, from inputs to output, constitutes one "iteration." We perform T iterations of this process, until convergence to steady state. This algorithm is an instance of exact block coordinate descent.

e. Training procedure

We train our networks by stochastic gradient descent. At each training step, we proceed as follows. First we pick a minibatch x of examples in the training set, and their corresponding labels y. Then we perform T iterations of the layer updates until convergence to the steady state. Once at steady state, we compute (or measure) the training error rate for the current minibatch, to monitor training. Next, we perform another K iterations of the layer updates, and we compute the gradient of the cost function with respect to the weights (conductances) through the trajectory. This corresponds to a form of truncated backpropagation, where we perform $T + K$ steps in total, but we only backpropagate through the last K steps. Then, we update all the conductances simultaneously, proportionally to their gradient. Finally, we clip all the negative conductance values to zero, to ensure that all the conductances remain non-negative.

We also performed simulations using equilibrium propagation for extracting the conductance gradients, in place of (truncated) backpropagation (see Appendix C 6).

5. Simulation details of ReLU neural networks

As a baseline for our DRNs, we train ReLU neural networks with one, two, and three hidden layers, termed ReLU NN-1H, ReLU NN-2H, and ReLU NN-3H, respectively.

a. ReLU neural network architecture

Each of the three ReLU neural network models has 784 input units (28×28), 10 output units, and 512 units per hidden layer.

TABLE IV. This table provides a more extensive presentation of the results of Table I. We compare the performance of DRNs trained with equilibrium propagation (EP), DRNs trained with truncated backpropagation (TBP), and their equivalent-size ReLU neural networks trained with backpropagation (BP). We train DRNs and ReLU neural networks with one, two, and three hidden layers on MNIST, Kuzushiji-MNIST, and Fashion-MNIST. For each setting, we perform five runs and report the mean and std values (%) of the training and test error rates.

Dataset	Network Model (Learning Algorithm)	Test Error (%)	Train Error (%)
MNIST	DRN-1H (EP)	1.53 ± 0.05	0.07 ± 0.00
	DRN-1H (TBP)	1.51 ± 0.04	0.04 ± 0.00
	ReLU NN-1H (BP)	1.67 ± 0.03	0.22 ± 0.01
	DRN-2H (EP)	1.60 ± 0.05	0.20 ± 0.01
	DRN-2H (TBP)	1.46 ± 0.05	0.20 ± 0.01
	ReLU NN-2H (BP)	1.34 ± 0.01	0.03 ± 0.01
	DRN-3H (EP)	1.77 ± 0.08	0.36 ± 0.01
	DRN-3H (TBP)	1.47 ± 0.08	0.30 ± 0.01
	ReLU NN-3H (BP)	1.48 ± 0.03	0.02 ± 0.00
Kuzushiji MNIST	DRN-1H (EP)	7.57 ± 0.11	0.08 ± 0.01
	DRN-1H (TBP)	7.67 ± 0.13	0.06 ± 0.00
	ReLU NN-1H (BP)	8.83 ± 0.11	0.12 ± 0.01
	DRN-2H (EP)	8.40 ± 0.10	0.39 ± 0.03
	DRN-2H (TBP)	8.27 ± 0.02	0.42 ± 0.01
	ReLU NN-2H (BP)	8.14 ± 0.13	0.05 ± 0.01
	DRN-3H (EP)	9.30 ± 0.17	0.63 ± 0.03
	DRN-3H (TBP)	8.39 ± 0.24	0.63 ± 0.01
	ReLU NN-3H (BP)	7.99 ± 0.23	0.03 ± 0.01
Fashion MNIST	DRN-1H (EP)	10.38 ± 0.11	6.14 ± 0.15
	DRN-1H (TBP)	10.31 ± 0.05	5.91 ± 0.19
	ReLU NN-1H (BP)	10.20 ± 0.11	5.13 ± 0.01
	DRN-2H (EP)	10.29 ± 0.14	6.20 ± 0.06
	DRN-2H (TBP)	9.88 ± 0.11	4.88 ± 0.05
	ReLU NN-2H (BP)	9.42 ± 0.22	2.78 ± 0.03
	DRN-3H (EP)	11.23 ± 0.17	8.19 ± 0.05
	DRN-3H (TBP)	9.79 ± 0.25	4.58 ± 0.04
	ReLU NN-3H (BP)	9.55 ± 0.14	1.82 ± 0.05

b. Initialization of the weights

We use the "Kaiming uniform" weight initialization scheme. Given two consecutive layers of indices ℓ and $\ell+1$, denoting by N_ℓ the number of units in layer ℓ, we initialize the matrix of conductances between these two layers as

$$w_{jk}^{(\ell+1)} \sim \mathcal{U}(-c,+c), \quad c := \sqrt{\frac{1}{N_\ell}}. \tag{C4}$$

c. Computing the weight gradients

At each step of stochastic gradient descent, we calculate the weight gradients using backpropagation. We use $\eta = 0.005$ as a learning rate for all the weights and biases.

6. Extended simulation results

Table I presents the results we obtained using truncated backpropagation (TBP) for training DRNs. Table IV extends the results of Table I to include the results we obtained using equilibrium propagation (EP) for training. Table IV also provides the error rates on the training set (training error rate), in addition to the test error rate.

In all cases, DRNs trained with TBP achieve comparable test performance to their equivalent-size ReLU neural networks. On the other hand, we observe that the training error rate in DRNs is often higher (sometimes significantly higher) than in ReLU neural networks, hinting at an optimization issue for DRNs. We note, however, that our Theorem 5 only states the existence of a configuration of conductance values that approximates the behavior of a ReLU neural network at inference, but it does not provide any guarantee that this configuration of conductance values can easily be reached through gradient descent optimization. We also observe that DRNs trained with EP often achieve lower performance than with TBP. Again, understanding these differences in performance is beyond the scope of our work. In this work, we provide a proof of existence, and we leave these open questions for future works to investigate.

APPENDIX D: EQUILIBRIUM PROPAGATION

The deep resistive networks studied in this work were introduced within the Equilibrium propagation training framework [3,14]. This appendix highlights the advantages that DRNs and EP offer over analog electrical implementations of neural networks using backpropagation. We begin by discussing the challenges associated with analog implementations of BP.

1. Analog backpropagation

There has been a long-standing effort to develop analog versions of neural networks and BP. Most electrical implementations employ crossbar arrays of memristors to perform matrix-vector multiplications (MVMs) [2], which constitute the core arithmetic operations of neural networks and BP. However, two additional operations are required: computing the nonlinear activation function, and computing its derivative. Based on whether these operations are executed in the analog or digital domain, analog implementations of BP can be broadly divided into two categories.

The first category aims to implement BP entirely in the analog domain, not just the MVMs. For instance, [45] utilized voltage followers to implement the ReLU activation function during the forward pass and analog switches to compute its derivative during the backward pass. However, this approach faces a significant challenge: analog devices cannot perfectly replicate the desired operations, leading to a mismatch between the activation function and its derivative. In practice, this mismatch will cause discrepancies between the calculated and actual weight gradients, leading to suboptimal optimization and reduced accuracy.

The second category employs mixed-signal (analog/digital) circuits, where MVMs are performed in the analog domain while nonlinear activation functions and their derivatives are computed digitally [2]. Such mixed-signal implementations of neural networks can mitigate the mismatch issue; however, the energy gains achieved by performing MVMs in the analog domain are offset by the power demands of analog-to-digital conversion required before the activation function [46].

Thus, it remains unclear whether BP is the most suitable training algorithm for analog computing platforms. This has led to the exploration of alternative approaches, such as EP, which may be better suited for these systems.

2. Equilibrium propagation

Equilibrium propagation , introduced in [3], has emerged as an alternative to BP for training analog systems, including nonlinear resistive networks [14]. For a more general, physics-oriented overview of EP, see [47]. Here, we focus on EP's application in DRNs, and explain how it addresses the mismatch problem found in analog BP.

EP is broadly applicable to energy-based systems that minimize an "energy" (or Lyapunov) function. For a DRN with ideal components, the energy function corresponds to power dissipation:

$$E_{\mathrm{DRN}}^{\mathrm{ideal}}(v) = \sum_{\ell=1}^{L}\sum_{j=1}^{N_\ell}\sum_{k=1}^{N_{\ell+1}} g_{jk}^{(\ell)}\left(v_j^{(\ell)} - v_k^{(\ell+1)}\right)^2, \tag{D1}$$

where $g_{jk}^{(\ell)}$ represents a memristor conductance, $v_j^{(\ell)}$ denotes a node electrical potential, and v is the vector of

all node potentials. As shown by [33], these ideal DRNs reach a steady state $v_\star$, given by

$$v_\star = \underset{v \in \mathcal{S}_{\text{DRN}}^{\text{ideal}}}{\arg\min}\, E_{\text{DRN}}^{\text{ideal}}(v), \tag{D2}$$

where the feasible set $\mathcal{S}_{\text{DRN}}^{\text{ideal}}$ is defined by diode constraints

$$\mathcal{S}_{\text{DRN}}^{\text{ideal}} = \Big\{ v \in \mathbb{R}^{\sum_{\ell=1}^{L} N_\ell} \mid v_k^{(\ell)} \geq 0 \text{ if } k \text{ is even}, \; v_k^{(\ell)} \leq 0 \text{ if } k \text{ is odd}, 1 \leq \ell \leq L-1, 1 \leq k \leq N_\ell \Big\}. \tag{D3}$$

In the present work, we have assumed ideal diodes to simplify the mathematical analysis of DRNs and derive our universal approximation theorem (specifically, this assumption has enabled us to derive Lemma 3 and subsequently an approximate equivalence with ReLU neural networks, Theorem 5). Similarly, [33] assumed ideal diodes to derive a fast algorithm to simulate DRNs on digital computers. Importantly, however, EP does not rely on this assumption of ideality. In fact, the i-v characteristics of nonlinear untrainable elements (diodes) do not need to be explicitly known. Indeed, nonlinear resistive networks with nonideal elements possess an energy function called the "co-content" [48] or "pseudo power" [14]. For instance, the co-content for a DRN with linear resistors and nonideal diodes is of the form

$$E_{\text{DRN}}(v) = \sum_{\ell=1}^{L} \sum_{j=1}^{N_\ell} \sum_{k=1}^{N_{\ell+1}} g_{jk}^{(\ell)} \left(v_j^{(\ell)} - v_k^{(\ell+1)} \right)^2 + E_{\text{diodes}}(v), \tag{D4}$$

where the term E_{diodes} depends solely on the i-v characteristics of the diodes. The steady state is then

$$v_\star = \arg\min_{v} E_{\text{DRN}}(v). \tag{D5}$$

The goal of learning is to minimize a cost function $C(v_\star)$ with respect to the trainable weights (memristor conductances), given boundary input conditions from voltage sources. For example, using the mean squared error,

$$C(v) = \sum_{k} \left(v_k^{\text{out}} - y_k \right)^2, \tag{D6}$$

where v_k^{out} is the kth output node's potential, and y_k is the kth target output.

The core idea of EP is to interpret βC as power dissipation by resistors with conductance β linking output nodes to "target nodes" (whose electrical potentials are set to target values). The resulting energy function (co-content) is

$$F(v, \beta) = E_{\text{DRN}}(v) + \beta C(v), \tag{D7}$$

where $\beta \geq 0$ is termed the "nudging parameter." Each training step of EP consists of:

(1) sourcing input voltages, setting $\beta = 0$ and allowing the system to reach the "free state," $v_\star^0 = \arg\min_v F(0, v)$;
(2) increasing $\beta > 0$, and allowing the system to reach the "nudge state," $v_\star^\beta = \arg\min_v F(\beta, v)$.

The gradient of the cost function with respect to the conductances can be estimated using the formula [3]

$$\frac{d}{dg} C(v_\star) = \left. \frac{d}{d\beta} \right|_{\beta=0} \frac{\partial F}{\partial g}\left(v_\star^\beta, \beta\right) \approx \underbrace{\frac{1}{\beta}\left(\frac{\partial F}{\partial g}\left(v_\star^\beta, \beta\right) - \frac{\partial F}{\partial g}\left(v_\star^0, 0\right) \right)}_{=: \widehat{\nabla}_g(\beta)}. \tag{D8}$$

Using the analytical form of the energy function, we obtain

$$\begin{aligned} \widehat{\nabla}_{g_{jk}^{(\ell)}}(\beta) &= \frac{1}{\beta}\left(\frac{\partial F}{\partial g_{jk}^{(\ell)}}\left(v_\star^\beta, \beta\right) - \frac{\partial F}{\partial g_{jk}^{(\ell)}}\left(v_\star^0, 0\right) \right) \\ &= \frac{1}{\beta}\left[\left((v_j^{(\ell)})^\beta - (v_k^{(\ell+1)})^\beta \right)^2 - \left((v_j^{(\ell)})^0 - (v_k^{(\ell+1)})^0 \right)^2 \right]. \end{aligned} \tag{D9}$$

Finally, we update the weights (memristor conductances) using these gradient estimates, to perform (or approximate) one step of gradient descent on the cost function:

$$\Delta g = -\eta \widehat{\nabla}_g(\beta). \tag{D10}$$

The learning rule for each weight (memristor) is local, and notably, it is agnostic to the i-v characteristics of the diodes. Whether the network components are ideal or nonideal, the EP algorithm and its learning rules remain unchanged. This feature makes EP robust to variability of analog devices, particularly untrainable nonlinear elements (diodes), which is a key advantage of DRNs and EP over analog implementations of neural networks and BP. Intuitively, the robustness of EP arises because the same network is used for both phases of training (free and nudge), so that nonidealities affect both states similarly and thus cancel out in the gradient calculation. EP thus solves the mismatch problem encountered in BP between the nonlinear activation function and its derivative.

In addition to the resistive networks considered in this work, EP has been applied to various energy-based systems, such as continuous Hopfield networks [3], Ising machines [49], and coupled phase oscillators [50]. A variant of EP called "coupled learning" has been developed and used in other systems as well, such as flow and elastic networks [51,52]. Recent works have extended EP to quantum systems too [53–55].

While EP's learning rule does not require knowledge of the characteristics of untrainable elements, it does require knowledge of the partial derivatives of the energy function with respect to trainable weights ($\partial E/\partial g$). However, a more advanced version of EP called "agnostic equilibrium propagation" has been introduced, which does not require knowledge of these partial derivatives either, and allows the weight updates to be carried out using physical dynamics [4].

[1] A. Momeni, B. Rahmani, B. Scellier, L. G. Wright, P. L. McMahon, C. C. Wanjura, Y. Li, A. Skalli, N. G. Berloff, and T. Onodera, Training of physical neural networks. *arXiv preprint* arXiv:2406.03372.

[2] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, Analog architectures for neural network acceleration based on non-volatile memory, Appl. Phys. Rev. **7**, 031301 (2020).

[3] B. Scellier and Y. Bengio, Equilibrium propagation: Bridging the gap between energy-based models and backpropagation, Front. Comput. Neurosci. **11**, 24 (2017).

[4] B. Scellier, S. Mishra, Y. Bengio, and Y. Ollivier, Agnostic physics-driven deep learning. *arXiv preprint* arXiv:2205.15021.

[5] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, Deep physical neural networks trained with backpropagation, Nature **601**, 549 (2022).

[6] V. Lopez-Pastor and F. Marquardt, Self-learning machines based on Hamiltonian echo backpropagation, Phys. Rev. X **13**, 031020 (2023).

[7] C. C. Wanjura and F. Marquardt, Fully nonlinear neuromorphic computing with linear wave scattering, Nat. Phys. **20**, 1 (2024).

[8] A. Momeni, B. Rahmani, M. Malléjac, P. Del Hougne, and R. Fleury, Backpropagation-free training of deep physical neural networks, Science **382**, 1297 (2023).

[9] M. Stern and A. Murugan, Learning without neurons in physical systems, Ann. Rev. Condens. Matter Phys. **14**, 417 (2023).

[10] G. Marcucci, D. Pierangeli, and C. Conti, Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons, Phys. Rev. Lett. **125**, 093901 (2020).

[11] H. Larocque and D. Englund, Universal linear optics by programmable multimode interference, Opt. Express **29**, 38257 (2021).

[12] Z. Gao, F.-K. Sun, R. Rohrer, and D. S. Boning, KirchhoffNet: A scalable ultra fast analog neural network. *arXiv preprint* arXiv:2310.15872.

[13] J. Harris, C. Koch, J. Luo, and J. Wyatt, in *Analog VLSI Implementation of Neural Systems*, edited by C. Mead and M. Ismail (Springer, Boston, MA, 1989), Vol. 80, pp. 27–55.

[14] J. Kendall, R. Pantone, K. Manickavasagam, Y. Bengio, and B. Scellier, Training end-to-end analog neural networks with equilibrium propagation. *arXiv preprint* arXiv:2006.01981.

[15] V. R. Anisetti, A. Kandala, B. Scellier, and J. Schwarz, Frequency propagation: Multimechanism learning in nonlinear physical networks, Neural Comput. **36**, 1 (2024).

[16] S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, Demonstration of decentralized physics-driven learning, Phys. Rev. Appl. **18**, 014040 (2022).

[17] S. Dillavou, B. D. Beyer, M. Stern, A. J. Liu, M. Z. Miskin, and D. J. Durian, Machine learning without a processor: Emergent learning in a nonlinear analog network, Proc. Natl. Acad. Sci. **121**, e2319718121 (2024).

[18] S.-i. Yi, J. D. Kendall, R. S. Williams, and S. Kumar, Activity-difference training of deep neural networks using memristor crossbars, Nat. Electron. **6**, 45 (2023).

[19] J. F. Wycoff, S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, Desynchronous learning in a physics-driven learning network, J. Chem. Phys. **156**, 144903 (2022).

[20] M. Stern, S. Dillavou, M. Z. Miskin, D. J. Durian, and A. J. Liu, Physical learning beyond the quasistatic limit, Phys. Rev. Res. **4**, L022037 (2022).

[21] M. Stern, S. Dillavou, D. Jayaraman, D. J. Durian, and A. J. Liu, Training self-learning circuits for power-efficient solutions, APL Mach. Learn. **2**, 016114 (2024).

[22] M. Stern, A. J. Liu, and V. Balasubramanian, Physical effects of learning, Phys. Rev. E **109**, 024311 (2024).

[23] F. Z. Kiraz, D.-K. G. Pham, and P. Desgreys, in *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)* (IEEE, Quebec City, QC, Canada, 2022), pp. 519–523.

[24] M. Watfa, A. Garcia-Ortiz, and G. Sassatelli, Energy-based analog neural network framework, Front. Comput. Neurosci. **17**, 1114651 (2023).

[25] S. Oh, J. An, S. Cho, R. Yoon, and K.-S. Min, Memristor crossbar circuits implementing equilibrium propagation for on-device learning, Micromachines **14**, 1367 (2023).

[26] S. Oh, R. Yoon, and K.-S. Min, Defect-tolerant memristor crossbar circuits for local learning neural networks, Nanomaterials **15**, 213 (2025).

[27] M. Guzman, F. Martins, M. Stern, and A. J. Liu, Microscopic imprints of learned solutions in adaptive resistor networks. *arXiv preprint* arXiv:2412.19356.

[28] M. Yildirim, N. U. Dinc, I. Oguz, D. Psaltis, and C. Moser, Nonlinear processing with linear optics, Nat. Photonics **18**, 1076 (2024).

[29] F. Xia, K. Kim, Y. Eliezer, S. Han, L. Shaughnessy, S. Gigan, and H. Cao, Nonlinear optical encoding enabled by recurrent linear scattering, Nat. Photonics **18**, 1067 (2024).

[30] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw. **6**, 861 (1993).

[31] D. Yarotsky, Error bounds for approximations with deep ReLU networks, Neural Netw. **94**, 103 (2017).
[32] X. Glorot, A. Bordes, and Y. Bengio, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (2011), pp. 315–323, https://proceedings.mlr.press/v15/glorot11a.html.
[33] B. Scellier, in *Proceedings of the 41st International Conference on Machine Learning*, Volume 235 of *Proceedings of Machine Learning Research* (PMLR, 2024), pp. 43477–43503, https://proceedings.mlr.press/v235/scellier24a.html.
[34] https://github.com/rain-neuromorphics/energy-based-learning.
[35] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control, Signals Syst. **2**, 303 (1989).
[36] J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. **81**, 3088 (1984).
[37] X. Xie and H. S. Seung, Equivalence of backpropagation and contrastive Hebbian learning in a layered network, Neural Comput. **15**, 441 (2003).
[38] A. Laborieux, M. Ernoult, B. Scellier, Y. Bengio, J. Grollier, and D. Querlioz, Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias, Front. Neurosci. **15**, 129 (2021).
[39] A. Laborieux and F. Zenke, in *Advances in Neural Information Processing Systems*, 35, 12950–12963 (2022), https://proceedings.neurips.cc/paper_files/paper/2022/hash/545a114e655f9d25ba0d56ea9a01fc6e-Abstract-Conference.html.
[40] B. Scellier, M. Ernoult, J. Kendall, and S. Kumar, in *Advances in Neural Information Processing Systems*, 36 (2024), https://proceedings.neurips.cc/paper_files/paper/2023/hash/a52b0d191b619477cc798d544f4f0e4b-Abstract-Conference.html.
[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).
[42] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, Deep learning for classical Japanese literature. *arXiv preprint* arXiv:1812.01718.
[43] H. Xiao, K. Rasul, and R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint* arXiv:1708.07747.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation in PyTorch (2017).
[45] M. Hermans, M. Burm, T. Van Vaerenbergh, J. Dambre, and P. Bienstman, Trainable hardware for dynamical computing using error backpropagation through physical media, Nat. Commun. **6**, 6729 (2015).
[46] B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, in *Proceedings of the 52nd Annual Design Automation Conference* (2015), pp. 1–6.
[47] B. Scellier, *A deep learning theory for neural networks grounded in physics*. PhD thesis, Université de Montréal, 2021.
[48] W. Millar, CXVI. Some general theorems for non-linear systems possessing resistance, Lond., Edinb., Dubl. Philos. Mag. J. Sci. **42**, 1150 (1951).
[49] J. Laydevant, D. Marković, and J. Grollier, Training an Ising machine with equilibrium propagation, Nat. Commun. **15**, 3671 (2024).
[50] Q. Wang, C. Wanjura, and F. Marquardt, Training coupled phase oscillators as a neuromorphic platform using equilibrium propagation, Neuromorphic Comput. Eng. (2024).
[51] M. Stern, D. Hexner, J. W. Rocks, and A. J. Liu, Supervised learning in physical networks: From machine learning to learning machines, Phys. Rev. X **11**, 021045 (2021).
[52] L. E. Altman, M. Stern, A. J. Liu, and D. J. Durian, Experimental demonstration of coupled learning in elastic networks, Phys. Rev. Appl. **22**, 024053 (2024).
[53] S. Massar and B. M. Mognetti, Equilibrium propagation: The quantum and the thermal cases, Quantum Stud.: Math. Found. **12**, 6 (2025).
[54] C. C. Wanjura and F. Marquardt, Quantum equilibrium propagation for efficient training of quantum systems based on onsager reciprocity. *arXiv preprint* arXiv:2406.06482.
[55] B. Scellier, Quantum equilibrium propagation: Gradient-descent training of quantum systems, *arXiv preprint* arXiv:2406.00879.