Abstract

We introduce *frequency propagation*, a learning algorithm for nonlinear physical networks. In a resistive electrical circuit with variable resistors, an activation current is applied at a set of input nodes at one frequency, and an error current is applied at a set of output nodes at another frequency. The voltage response of the circuit to these boundary currents is the superposition of an ‘activation signal’ and an ‘error signal’ whose coefficients can be read in different frequencies of the frequency domain. Each conductance is updated proportionally to the product of the two coefficients. The learning rule is local and proved to perform gradient descent on a loss function. We argue that frequency propagation is an instance of a *multi-mechanism learning* strategy for physical networks, be it resistive, elastic, or flow networks. Multi-mechanism learning strategies incorporate at least two physical quantities, potentially governed by independent physical mechanisms, to act as activation and error signals in the training process. Locally available information about these two signals is then used to update the trainable parameters to perform gradient descent. We demonstrate how earlier work implementing learning via *chemical signaling* in flow networks ([1]) also falls under the rubric of multi-mechanism learning.

Frequency propagation: Multi-mechanism learning in nonlinear physical networks

I. INTRODUCTION

Advancements in artificial neural networks (ANN) ([2]) have inspired a search for adaptive physical networks that can be optimized to achieve desired functionality ([1, 3–9]). Similar to ANNs, adaptive physical networks modify their learning degrees of freedom to approximate a desired input-to-output function ; but unlike ANNs, they achieve this using physical laws. In a physical network, the input is typically an externally applied boundary condition, and the output is the network’s response to this input, or a statistic of this response. For instance, in a resistive network, an input signal can be fed in the form of applied currents or voltages, and the output may be the vector of voltages across a subset of nodes of the network. The learning degrees of freedom of the network are, for example, the conductances of the resistors (assuming variable resistors). Ideally, these learning parameters must be updated using only locally available information. Otherwise, the network would require additional channels to transmit the gradient information. Moreover, these parameter updates should preferably follow the direction of gradient descent in the loss function landscape, as is the case for ANNs.

Existing learning algorithms for adaptive physical networks include *equilibrium propagation* ([3, 10]) and *coupled learning* ([5]). These algorithms are based on the idea of *contrastive learning* ([11]) and proceed as follows. In a first phase, an input is presented to the network, either in the form of boundary currents or voltages, and the network is allowed to settle to equilibrium (the ‘free state’), where a supervisor checks the output of the system. Then the supervisor *nudges* the output towards the desired output. This perturbation causes the system to settle to a new (‘perturbed’) equilibrium state, which is a slightly more accurate approximation of the function that one wants to learn. The supervisor then compares the perturbed state with the free state to make changes in the learning degrees of freedom in such a way that the network spontaneously produces an output that is slightly closer to the desired output. In the limit of infinitesimal nudging, this procedure performs gradient descent on the squared prediction error ([12]).

The above procedure is not entirely ‘physical’ in nature, as it requires storing the free state to compare it with the perturbed state. For example, in the experimental work of [7], the authors use two copies of the same network to compute the two states. Alternatively, [13] use a single network, but the authors make use of additional SRAM to store the two states before performing the weight updates. Another idea proposed by [3] is to use a capacitor (sample-and-hold amplifier) at each

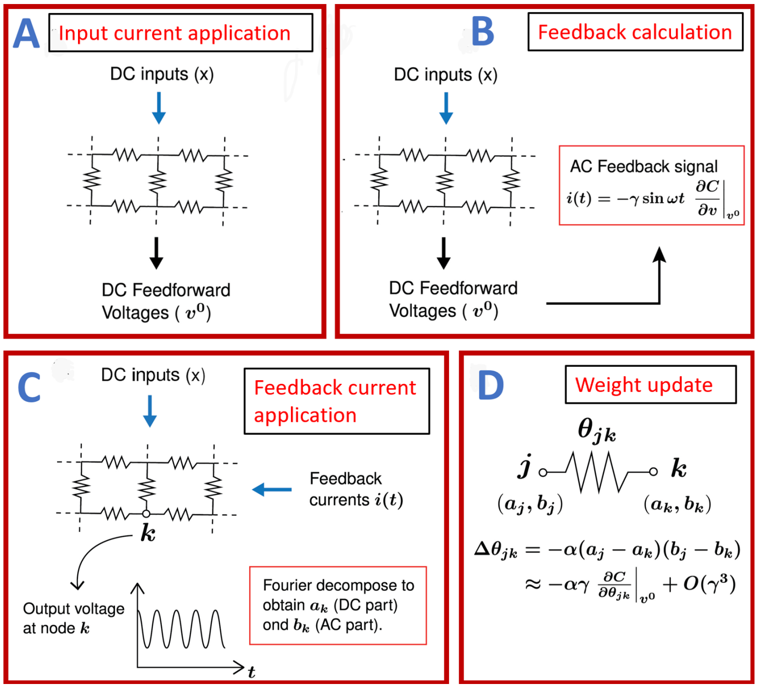
A
Input current application
DC inputs (x)
DC Feedforward Voltages (v^0)
B
Feedback calculation
DC inputs (x)
AC Feedback signal
$i(t) = -\gamma \sin \omega t \left.\frac{\partial C}{\partial v}\right|_{v^0}$
DC Feedforward Voltages (v^0)
C
Feedback current application
DC inputs (x)
Feedback currents $i(t)$
k
Output voltage at node k
Fourier decompose to obtain a_k (DC part) and b_k (AC part).
t
D
Weight update
θ_{jk}
j
k
(a_j, b_j)
(a_k, b_k)
$\Delta\theta_{jk} = -\alpha(a_j - a_k)(b_j - b_k)$
$\approx -\alpha\gamma \left.\frac{\partial C}{\partial \theta_{jk}}\right|_{v^0} + O(\gamma^3)$

FIG. 1. *A graphical summary of Frequency Propagation.*

node/unit/neuron to store the free state values, but this idea has not been verified experimentally. In this work, we propose an alternative *multi-mechanism learning* approach to overcome this hurdle. Our approach incorporates two physical quantities, each driven by their own respective mechanisms: one quantity acting as an *activation* signal and the other acting as an *error* signal. This concept is motivated by biological systems implementing functionality via multiple biophysical routes or mechanisms. Such functionality can be chemical, electrical or even mechanical in nature with potential interactions between such mechanisms. For instance, in the brain, activity can propagate from one cell to another via electrical and chemical synapses, as opposed to just one mechanism, if you will ([14]). Given this modularity in functionality in biology, it would be remiss not to explore such richness in how adaptive physical networks learn. Alternatively, as we shall soon see, the modularity is not necessarily in terms of mechanical versus chemical versus electrical signals, but distinguishable signals.

We introduce *frequency propagation* (Freq-prop), a physical learning algorithm falling under the umbrella concept of multi-mechanism learning. In Freq-prop, the activation and error signals are both sent through a single channel, but are encoded in different frequencies of the frequency domain ; we can thus obtain the respective responses of the system through frequency (Fourier) decomposition. This algorithm, which we show to perform gradient descent, can be used to train adaptive non-linear networks such as resistor networks, elastic networks and flow networks. Freq-prop thus has the potential to be an all-encompassing approach. See Fig. 1 for a graphical summary of Freq-prop. In the next section we present this idea of frequency propagation in the context of resistor networks, and in section III we show that frequency propagation is an example of multi mechanism learning and can be generalized to train various physical systems like flow and mechanical networks.

II. NONLINEAR RESISTIVE NETWORKS

A resistive network is an electrical circuit of nodes interconnected by resistive devices, which includes linear resistors and diodes. Let N be the number of nodes in the network, and denote v_j the electric potential of node j. A subset of the nodes are *input nodes*, where we can set input currents: we denote x_j the input current at input node j. For each pair of nodes j and k, we denote θ_{jk} the conductance of the resistor between these nodes (provided that the corresponding branch contains a linear resistor). We further denote $\theta = \{\theta_{jk} : \text{linear branch } (j,k)\}$ the vector of conductances, and $x = (x_1, x_2, \ldots, x_N)$ the vector of input currents, where by convention $x_j = 0$ if node j is not an input node. Finally, we denote $v = (v_1, v_2, \ldots, v_N)$ the configuration of the nodes' electric potentials, and $v(\theta, x)$ the equilibrium value of v as a function of the branch conductances (θ) and the input currents (x).

The following result, known since the work of Millar ([15]), provides a characterization of the equilibrium state – see also ([3]) for a proof of this result with notations closer to ours.

Theorem 1 *There exists a real-valued function $E(\theta, x, v)$ such that*

$$v(\theta, x) = \arg\min_{v} E(\theta, x, v). \tag{1}$$

Furthermore, $E(\theta, x, v)$ is of the form

$$E(\theta, x, v) = E_{\text{input}}(x, v) + E_{\text{nonlinear}}(v) \tag{2}$$

$$+ \sum_{\text{linear branch } (j,k)} \frac{1}{2}\theta_{jk} \left(v_j - v_k\right)^2, \tag{3}$$

where $E_{\text{input}}(x, v)$ is a function of x and v, and $E_{\text{nonlinear}}(v)$ is a function of v only.

$E(\theta, x, v)$ is the 'energy function' of the system, also called the *co-content* ([15]), and the equilibrium state $v(\theta, x)$ is a minimizer of the energy function. The energy function contains an energy term $E_{\text{input}}(x, v)$ associated to boundary input currents x. It also contains energy terms of the form $\theta_{jk} \left(v_j - v_k\right)^2$ representing the power dissipated in branch (j, k). The term $E_{\text{nonlinear}}(v)$ contains all nonlinearities of the system. In a *linear* resistance network (i.e. when $E_{\text{nonlinear}}(v) = 0$), it is well known that the equilibrium configuration of node electric potentials minimizes the power dissipation ; Theorem 1 generalizes this result to nonlinear networks. Below we explain how the different terms of $E(\theta, x, v)$ are constructed.

Constructing the energy function. Each branch is characterised by its current-voltage characteristic $i_{jk} = \widehat{i}_{jk}(v_j - v_k)$, where $\widehat{i}_{jk}(\cdot)$ is a real-valued function that returns i_{jk}, the current flowing from j to k in response to the voltage $v_j - v_k$. The energy term corresponding to branch (j, k), called the *co-content* of the branch ([15]), is by definition

$$E_{jk}(v_j - v_k) = \int_0^{v_j - v_k} \widehat{i}_{jk}(v')dv'. \tag{4}$$

In general, the characteristic function $\widehat{i}_{jk}(\cdot)$ is arbitrary, i.e. *nonlinear*. However, some branches are *linear*, meaning that their current-voltage characteristic is of the form $i_{jk} = \theta_{jk} \left(v_j - v_k\right)$, where θ_{jk} is the branch conductance [16]. For such linear branches, the energy term is

$$E_{jk}(v_j - v_k) = \frac{1}{2}\theta_{jk} \left(v_j - v_k\right)^2, \tag{5}$$

which is the power dissipated in branch (j, k).

We gather all the energy terms of nonlinear branches under a unique term:

$$E_{\text{nonlinear}}(v) = \sum_{\text{nonlinear branch } (j,k)} E_{jk}(v_j - v_k), \tag{6}$$

where we recall that $v = (v_1, v_2, \ldots, v_N)$.

As for the energy term $E_{\text{input}}(x, v)$, we present two ways to impose boundary conditions to the network to feed it with input signals x, either in the form of boundary currents or boundary electric potentials. Recall that we write $x = (x_1, x_2, \ldots, x_N)$ the vector of input signals, where $x_j = 0$ if node j is not an input node. In the case of boundary currents, the corresponding energy term is

$$E_{\text{input}}^{\text{current}}(x, v) = \sum_{j \in \{\text{input nodes}\}} x_j v_j, \tag{7}$$

whereas in the case of boundary electric potentials, the energy term is

$$E_{\text{input}}^{\text{voltage}}(x,v)=\begin{cases}0 & \text{if} \quad v_j=x_j, \\ & \forall j \in \{\text{input nodes}\}, \\ +\infty & \text{otherwise},\end{cases} \tag{8}$$

i.e. the electric potential v_j is clamped to x_j for every input node j (so that the energy remains finite).

Putting all the energy terms together, and denoting $E_{\text{input}}(x,v)$ the energy term of input signals (either $E_{\text{input}}^{\text{current}}(x,v)$ or $E_{\text{input}}^{\text{voltage}}(x,v)$ depending on the case), we get the energy function of Eq. (2-3).

III. MULTI-MECHANISM LEARNING VIA FREQUENCY PROPAGATION

Learning in a resistive network consists in adjusting the branch conductances (θ) so that the network exhibits a desired behavior, i.e. a desired input-output function $x \mapsto v(\theta,x)$. In machine learning, this problem is formalized by introducing a *cost function* C. Given an input-output pair (x,y), the quantity $C(v(\theta,x),y)$ measures the discrepancy between the 'model prediction' $v(\theta,x)$ and the desired output y. The learning objective is to find the parameters θ that minimize the expected cost $\mathbb{E}_{(x,y)}\left[C(v(\theta,x),y)\right]$ over input-output pairs (x,y) coming from the data distribution for the task that the system must solve.

In deep learning, the main tool for this optimization problem is stochastic gradient descent (SGD) ([17]): at each step we pick at random an example (x,y) from the training set and update the parameters as

$$\Delta\theta=-\eta\frac{\partial\mathcal{L}}{\partial\theta}(\theta,x,y), \tag{9}$$

where η is a step size, and

$$\mathcal{L}(\theta,x,y)=C(v(\theta,x),y) \tag{10}$$

is the per-example *loss function*.

We now present *frequency propagation* (Freq-prop), a learning algorithm for physical networks whose update rule performs SGD. Freq-prop proceeds by modifying the energy of the network to push or pull away the network's output values from the desired outputs. In the case of a resistive network (Section II), we inject sinusoidal currents at the output nodes of the network, $i(t)=\gamma\sin(\omega t)\,\frac{\partial C}{\partial v}(v,y)$, where t denotes time, ω is a frequency, and γ is a small positive constant[18]. This amounts to augment the energy function of the system by a time-dependent sinusoidal energy term $\gamma\,\sin(\omega t)\,C(v,y)$. Due to this perturbation, the system's response $v(t)$ minimizing the energy at time t is

$$v(t)=\underset{v}{\arg\min}\;\left[E(\theta,x,v)+\gamma\,\sin(\omega t)\,C(v,y)\right]. \tag{11}$$

The response $v(t)$ is periodic of period $T=2\pi/\omega$, and for small perturbations (i.e. $\gamma \ll 1$), it is approximately sinusoidal. Next, we assume that we can recover the first two vectors of Fourier coefficients of $v(t)$, i.e. the vectors a and b such that

$$a=\frac{1}{T}\int_0^T v(t)dt, \qquad b=\frac{2}{T}\int_0^T v(t)\sin(\omega t)dt. \tag{12}$$

Finally, denoting $a=(a_1,a_2,\ldots,a_N)$ and $b=(b_1,b_2,\ldots,b_N)$, we update each parameter θ_{jk} according to the learning rule

$$\Delta\theta_{jk}=-\alpha(b_j-b_k)\cdot(a_j-a_k), \tag{13}$$

where α is a positive constant.

Theorem 2 *For every parameter θ_{jk}, we have*

$$\Delta\theta_{jk}=-\alpha\,\gamma\frac{\partial\mathcal{L}}{\partial\theta_{jk}}(\theta,x,y)+O(\gamma^3) \tag{14}$$

when $\gamma \to 0$.

Namely, the learning rule (13) approximates one step of gradient descent with respect to the loss, with learning rate $\alpha\,\gamma$. Note that this learning rule is local: it requires solely locally available information for each parameter θ_{jk}.

[Proof of Theorem 2] Let θ, x and y be fixed. For every $\beta \in \mathbb{R}$, we denote

$$v_\star^\beta = \arg\min_v \; \left[E(\theta, x, v) + \beta\, C(v, y) \right]. \tag{15}$$

With this notation, note that the response $v(t)$ of Eq. (11) rewrites $v(t) = v_\star^{\gamma \sin(\omega t)}$. Let us write the second-order Taylor expansion of $v_\star^\beta$ around $\beta = 0$:

$$v_\star^\beta = v_\star^0 + \beta \left. \frac{\partial v_\star^\beta}{\partial \beta} \right|_{\beta=0} + \frac{\beta^2}{2} \left. \frac{\partial^2 v_\star^\beta}{\partial \beta^2} \right|_{\beta=0} + O(\beta^3), \tag{16}$$

where $v_\star^0 = v(\theta, x)$ by definition (1), and $\left.\frac{\partial v_\star^\beta}{\partial \beta}\right|_{\beta=0}$ and $\left.\frac{\partial^2 v_\star^\beta}{\partial \beta^2}\right|_{\beta=0}$ denote the derivative and second-derivative of $v_\star^\beta$ at $\beta = 0$. Taking $\beta = \gamma \sin(\omega t)$ in the above formula, we get

$$v(t) = v_\star^{\gamma \sin(\omega t)} = v_\star^0 + \gamma \sin(\omega t) \left. \frac{\partial v_\star^\beta}{\partial \beta} \right|_{\beta=0} \tag{17}$$

$$+ \frac{\gamma^2}{2} \sin(\omega t)^2 \left. \frac{\partial^2 v_\star^\beta}{\partial \beta^2} \right|_{\beta=0} + O(\gamma^3), \tag{18}$$

uniformly in t. Therefore, the first two vectors of Fourier coefficients a and b of the periodic function $v(t)$, with time period $T = 2\pi/\omega$ are

$$a = \frac{1}{T} \int_0^T v(t) dt = v_\star^0 + \frac{\gamma^2}{4} \left. \frac{\partial^2 v_\star^\beta}{\partial \beta^2} \right|_{\beta=0} + O(\gamma^3), \tag{19}$$

$$b = \frac{2}{T} \int_0^T v(t) \sin(\omega t) dt = \gamma \left. \frac{\partial v_\star^\beta}{\partial \beta} \right|_{\beta=0} + O(\gamma^3). \tag{20}$$

Next, we know from the equilibrium propagation formula (Theorem 2.1 in ([10])) that the gradient of the loss $\mathcal{L}$ is equal to

$$\frac{\partial \mathcal{L}}{\partial \theta}(\theta, x, y) = \left. \frac{d}{d\beta} \right|_{\beta=0} \frac{\partial E}{\partial \theta}(\theta, x, v_\star^\beta). \tag{21}$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial \theta}(\theta, x, y) = \frac{\partial^2 E}{\partial \theta \partial v}(\theta, x, v_\star^0) \cdot \left. \frac{\partial v_\star^\beta}{\partial \beta} \right|_{\beta=0}. \tag{22}$$

Multiplying both sides by γ, and using (20), we get

$$\gamma \frac{\partial \mathcal{L}}{\partial \theta}(\theta, x, y) = \frac{\partial^2 E}{\partial \theta \partial v}(\theta, x, v_\star^0) \cdot b + O(\gamma^3). \tag{23}$$

Finally, given the form of the energy function (2), and using $b = O(\gamma)$ and $v_\star^0 = a + O(\gamma^2)$ from Eq. (19), we get for every parameter θ_{jk}

$$\gamma \frac{\partial \mathcal{L}}{\partial \theta_{jk}}(\theta, x, y) = (a_j - a_k) \cdot (b_j - b_k) + O(\gamma^3). \tag{24}$$

Therefore the learning rule

$$\Delta \theta_{jk} = -\alpha (b_j - b_k) \cdot (a_j - a_k) \tag{25}$$

satisfies

$$\Delta \theta_{jk} = -\alpha\, \gamma \frac{\partial \mathcal{L}}{\partial \theta_{jk}}(\theta, x, y) + O(\gamma^3). \tag{26}$$

Hence the result.

Remark 1. For simplicity, we have omitted the time of relaxation to equilibrium in our analysis. However, a practical circuit has an effective capacitance C_{eff} and therefore will equilibrate in time $\tau_{\text{relax}} \sim R_{\text{eff}} C_{\text{eff}}$, where R_{eff} is the effective resistance of the circuit. Our learning algorithm will work as long as the circuit equilibrates much faster than the timescale of oscillation ($\tau_{\text{relax}} \ll 1/\omega$). Our analysis thus requires that C_{eff} be small enough for the assumption $\tau_{\text{relax}} \ll 1/\omega$ to hold. If this is not the case, there will be a tradeoff between how fast one can train the network with Freq-Prop vs how accurate the approximation is for gradient. We leave the study of the regime where C_{eff} is non negligible for future work. We note however that the effective capacitance of the circuit is expected to grow linearly with the size of the network (the total amount of wire), so that inference time grows linearly with the size of the network, too. We also note that the same is true for deep neural networks: in a feedforward network, both inference (the forward pass) and training (the backward pass of backpropagation) grow linearly with the size of the network.

Remark 2. While our nudging method (11) is inspired by the one of *equilibrium propagation* ([3, 12]), it is also possible to apply the nudging variant of *coupled learning* ([5]) which might be easier to implement in practice ([7]). To do this, we denote v_O^F the 'free' equilibrium value of the output nodes of the network (where the prediction is read), without nudging. Then, at time t, we *clamp* the output nodes to $v_O^C(t) = v_O^F + \gamma \sin(\omega t)(y - v_O^F)$. This nudging method can be achieved via AC voltage sources at output nodes. We note however that Theorem 2 does not hold with this alternative nudging method.

Remark 3. Measuring b_j for every node j as per Eq. (12) requires that we use the same reference time $t = 0$ for all nodes, i.e. it requires global synchronization of the measurements for all nodes. However, in practice, there may be a time delay t_j between nudging and measurement, leading to a measured response $v_j(t) = a_j + b_j \sin(\omega(t+t_j)) + O(\gamma^3)$ at node j. Without any information about t_j, we can only obtain the absolute value of the coefficient b_j, not its sign. We propose a solution to this issue in Appendix A.

IV. DISCUSSION

We have introduced frequency propagation (Freq-prop), a physical learning algorithm that falls in the category of Multi-mechanism Learning (MmL). In MmL, separate and "distinguishable" activation and error signals both contribute to a local learning rule, such that trainable parameters (e.g. conductances of variable resistors) perform gradient descent on a loss function. In Freq-prop, the activation and error signals are implemented using different frequencies of a single physical quantity (e.g. voltages or currents) and are thus distinguishable. We note however that the 'distinguishability' of the signals does not mean that they are mathematically 'independent': in Freq-prop, the error signal depends on the activation signal via the Hessian of the network. Other potential MmL algorithms may involve independent physical mechanisms, such as an electrical activation signal and a chemical error signal or vice versa. Multi-mechanism learning algorithms, such as Freq-prop, may have implications towards designing fast and low-power, or high-efficiency, hardware for AI, as they are rooted in physical principles. For the time being, inroads are being made by using backpropagation to train controllable physical systems in a hybrid *in silico-in situ* approach ([19]). As we work towards a fully in situ approach, Freq-prop is a natural candidate. And while the *in situ* realization of a nonlinear resistor network is an obvious starting point, there are potential limitations, particularly in terms of timescales. Consider the time of relaxation to equilibrium (τ_{relax}), the time scale of the sinusoidal nudging signal ($T = 2\pi/\omega$), and the time scale of learning (τ_{learning}). Our learning methodology requires that $\tau_{\text{relax}} \ll T < \tau_{\text{learning}}$. More specifically,

1. Once input is applied, the network reaches equilibrium in time τ_{relax}.
2. Based on the network's output, a sinusoidal nudging signal of frequency ω is applied at the output nodes. The time scale of evolution of this sinusoidal nudging wave is $T = 2\pi/\omega$. Assuming that $\tau_{\text{relax}} \ll T$, the network is at equilibrium at every instant t.
3. We observe the network's response $v(t)$ for a time $\tau_{\text{obs}} > T$ to extract the coefficients a and b of Eq. (12). Updating the conductances of the resistors takes a time $\tau_{\text{learning}} \sim \tau_{\text{obs}}$ using the values of a and b to determine the magnitude and sign of these updates.

Finally, could something like Freq-prop occur in the brain? Earlier work analyzing local field potentials recorded simultaneously from different regions in the cortex suggested that feedforward signaling is carried by gamma-band (30–80 Hz) activity, whereas feedback signaling is mediated by alpha-(5–15Hz) or beta- (14–18 Hz) band activity, though local field potentials are not actively relayed between regions ([20]). More recent work in the visual cortex argues that feedforward and feedback signaling rely on separate "channels" since correlations in neuronal population activity patterns, which are actively relayed between regions, are distinct during feedforward- and feedback-dominated periods ([21]). Freq-prop is also related in spirit to the idea of frequency multiplexing in biological neural networks ([22–24]), which uses the simultaneous encoding of two or more signals. While Freq-prop here uses only two separate signals – an activation signal and an error signal – one can envision multiple activation and error signals being encoded to accommodate vector inputs and outputs and to accommodate multiple, competing cost functions. With multiple activation and error signals one can also envision coupling learning via chemical signaling (Appendix D) with Freq-prop, for example, to begin to capture the full computational *creativity* of the brain.

ACKNOWLEDGMENTS

The authors thank Sam Dillavou, Nachi Stern, Andrea Liu, Doug Durian, and Jack Kendall for discussion. JMS acknowledges financial support from NSF-DMR-1832002 and NSF-DMR-2204312.

[1] V. R. Anisetti, B. Scellier, and J. M. Schwarz, "Learning by non-interfering feedback chemical signaling in physical networks," *arXiv preprint arXiv:2203.12098*, 2022.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] 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*, 2020.

[4] M. Stern, C. Arinze, L. Perez, S. E. Palmer, and A. Murugan, "Supervised learning through physical changes in a mechanical system," Jun 2020.

[5] M. Stern, D. Hexner, J. W. Rocks, and A. J. Liu, "Supervised learning in physical networks: From machine learning to learning machines," *Physical Review X*, vol. 11, no. 2, p. 021045, 2021.

[6] V. Lopez-Pastor and F. Marquardt, "Self-learning machines based on hamiltonian echo backpropagation," *arXiv preprint arXiv:2103.04992*, 2021.

[7] S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, "Demonstration of decentralized, physics-driven learning," *arXiv preprint arXiv:2108.00275*, 2021.

[8] B. Scellier, S. Mishra, Y. Bengio, and Y. Ollivier, "Agnostic physics-driven deep learning," *arXiv preprint arXiv:2205.15021*, 2022.

[9] M. Stern and A. Murugan, "Learning without neurons in physical systems," 2022.

[10] B. Scellier, *A deep learning theory for neural networks grounded in physics*. PhD thesis, Université de Montréal, 2021.

[11] P. Baldi and F. Pineda, "Contrastive learning and neural oscillations," *Neural computation*, vol. 3, no. 4, pp. 526–545, 1991.

[12] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, p. 24, 2017.

[13] S.-i. Yi, J. D. Kendall, R. S. Williams, and S. Kumar, "Activity-difference training of deep neural networks using memristor crossbars," *Nature Electronics*, vol. 6, no. 1, pp. 45–51, 2023.

[14] A. E. Pereda, "Electrical synapses and their functional interactions with chemical synapses," *Nature Reviews Neuroscience*, vol. 15, no. 4, pp. 250–263, 2014.

[15] W. Millar, "Cxvi. some general theorems for non-linear systems possessing resistance," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 333, pp. 1150–1160, 1951.

[16] To avoid any confusion, we stress that θ_{jk} is a scalar, whereas $\hat{\imath}_{jk}(\cdot)$ is a real-valued function. Thus, $\theta_{jk}\,(v_j - v_k)$ denotes the product of θ_{jk} and $v_j - v_k$, whereas $\hat{\imath}_{jk}(v_j - v_k)$ denotes the function $\hat{\imath}_{jk}$ applied to the voltage $v_j - v_k$.

[17] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[18] In practical situations such as the squared error prediction, the cost function C depends only on the state of output nodes ; therefore nudging requires injecting currents at output nodes only.

[19] 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*, vol. 601, no. 7894, pp. 549–555, 2022.

[20] A. M. Bastos, J. Vezoli, C. A. Bosman, J.-M. Schoffelen, R. Oostenveld, J. R. Dowdall, P. De Weerd, H. Kennedy, and P. Fries, "Visual areas exert feedforward and feedback influences through distinct fre-

quency channels," *Neuron*, vol. 85, no. 2, pp. 390–401, 2015.

[21] J. D. Semedo, A. I. Jasper, A. Zandvakili, A. Krishna, A. Aschner, C. K. Machens, A. Kohn, and B. M. Yu, "Feedforward and feedback interactions between visual cortical areas use different population activity patterns," *Nature communications*, vol. 13, no. 1, pp. 1–14, 2022.

[22] R. Naud and H. Sprekeler, "Sparse bursts optimize information transmission in a multiplexed neural code," *Proceedings of the National Academy of Sciences*, vol. 115, no. 27, pp. E6329–E6338, 2018.

[23] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nature neuroscience*, vol. 24, no. 7, pp. 1010–1019, 2021.

[24] T. Akam and D. M. Kullmann, "Oscillatory multiplexing of population codes for selective communication in the mammalian brain," Jan 2014.

[25] C. Cherry, "Cxvii. some general theorems for non-linear systems possessing reactance," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 333, pp. 1161–1177, 1951.

[26] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons.," *Proceedings of the national academy of sciences*, vol. 81, no. 10, pp. 3088–3092, 1984.

[27] A. Laborieux and F. Zenke, "Holomorphic equilibrium propagation computes exact gradients through finite size oscillations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12950–12963, 2022.

[28] N. Zucchet and J. Sacramento, "Beyond backpropagation: implicit gradients for bilevel optimization," *arXiv preprint arXiv:2205.03076*, 2022.

Appendix A: Choice of the nudging signal

We have seen in section III that, using a sinusoidal nudging signal $\gamma\, \sin(\omega t)\, C(v, y)$, the measured response at node j will be of the form $v_j(t) = a_j + b_j \sin(\omega(t + t_j)) + O(\gamma^3)$, where t_j is the time delay between nudging and measurement. Unfortunately, it is not possible to recover the sign of b_j without any knowledge of t_j. This problem can be overcome by using a different nudging signal.

In general, if we nudge the system by an energy term $\gamma f(t)\, C(v, y)$, where $f(t)$ is an arbitrary function such that $\sup_t |f(t)| < \infty$, then the system's response at node j will be of the form $v_j(t) = a_j + b_j f(t + t_j) + O(\gamma^2)$. Our goal is to choose a f so that we can obtain for every node j the values of a_j and b_j by measuring only $v_j(t)$, without knowing t_j.

Clearly, this is not possible for all functions f. For example, if $f(\cdot)$ is a constant, then $v_j(\cdot)$ is also a constant, and we cannot recover the values of a_j and b_j from $v_j(\cdot)$ alone. As seen above, another example for which this is not possible is $f(t) = \sin(\omega t)$. This is because a time delay $t_j = \pi/\omega$ will change the sign of the signal, $\sin(\omega(t + t_j)) = -\sin(\omega t)$; therefore the sign of b_j cannot be recovered without any knowledge of t_j.

An example of a nudging signal for which we can infer the values of a_j and b_j (up to $O(\gamma^2)$) is $f(t) = |\sin(\omega(t))|$. To do this, we observe the response at node j

$$v_j(t) = a_j + b_j |\sin(\omega(t + t_j))| + O(\gamma^2) \tag{A1}$$

for a duration τ_{obs} greater than the time period of the signal $T = 2\pi/\omega$. The coefficients a_j and b_j can be obtained by identifying the times where the signal's derivative is zero or is discontinuous. Specifically, denoting $\partial_+ v_j(t)$ and $\partial_- v_j(t)$ the left and right derivatives of the signal at time t, we have

$$a_j = v_j(t_1) + O(\gamma^2) \text{ where } \partial_+ v(t_1) \neq \partial_- v(t_1), \tag{A2}$$

$$b_j = v_j(t_2) - v_j(t_1) + O(\gamma^2) \text{ where } \partial v(t_2) = 0. \tag{A3}$$

More generally, we will show that, in principle, it is possible to recover the coefficients a_j and b_j if and only if the function f has the property that there is no τ such that $f(t) = \sup f + \inf f - f(t+\tau)$ for every t. In other words, no amount of time delay converts the signal's 'upright' form to its 'inverted' form or vice versa.

Let $f(t)$ denote the nudging signal. Assuming that f is bounded, recall that, for every j, the measured response $v_j(t)$ at node j is of the form $v_j(t) = a_j + b_j f(t + t_j) + O(\gamma^2)$, where a_j and b_j are the numbers that we wish to recover (up to $O(\gamma^2)$) to implement the parameter update, and t_j is an unknown time delay. Our goal is to obtain for every node j the values of a_j and b_j by measuring only $v_j(t)$, without any knowledge of t_j.

We now establish a necessary and sufficient condition on the nudging signal $f(t)$ so that one can, at least in principle, uniquely obtain the values of a_j and b_j for every node j. We are concerned with quantities that depend only on a single node and hence we will drop the node index with the understanding that all of the analysis applies to any arbitrary node.

Let F denote the set of all real-valued, bounded functions, and let f be an element of F. Let $\mathcal{C}_f : \mathbb{R}^3 \to F$ be the function that maps the parameters (a, b, t_0) to the function $v(\cdot) = a + bf(\cdot + t_0)$. We define the following equivalence relation on F: two functions $g, h \in F$ are equivalent if they differ by a time translation, i.e., $g \sim h$ if and only if there exists a $t_0 \in \mathbb{R}$ such that $g(t) = h(t + t_0)$ for all $t \in \mathbb{R}$. Let $\tilde{F} = F/\sim$ be the quotient of F under this equivalence relation and let $[g]$ be the equivalence class that contains the function g. The map $\mathcal{C}_f$ can be lifted to yield $\tilde{\mathcal{C}}_f : \mathbb{R}^2 \to \tilde{F}$ such that $\tilde{\mathcal{C}}_f(a, b) = [a + bf]$. In order to be able to uniquely extract a and b from any equivalence class of the form $[a + bf]$, the function $\tilde{\mathcal{C}}_f$ has to be injective. This can be re-expressed as a direct condition on the nudging signal f.

Proposition 3 *The following statements are equivalent:*

P1: The function $\tilde{\mathcal{C}}_f : \mathbb{R}^2 \to \tilde{F}$ defined by $\tilde{\mathcal{C}}_f(a, b) = [a + bf]$ is injective.
P2: There exists no $\tau \in \mathbb{R}$ such that for all $t \in \mathbb{R}$,
$f(t) = \sup f + \inf f - f(t + \tau)$.

where $\sup f = \sup_t f(t)$ and $\inf f = \inf_t f(t)$ denote the supremum and infimum values of the nudging signal f respectively.

We establish this by proving that the negation of the two statements are equivalent, i.e., the following statements are equivalent:

N1: There exists two distinct pairs of real numbers (a_1, b_1) and (a_2, b_2) such that $[a_1 + b_1 f] = [a_2 + b_2 f]$.
N2: There exists a $\tau \in \mathbb{R}$ such that for all $t \in \mathbb{R}$,
$f(t) = \sup f + \inf f - f(t + \tau)$.

Suppose that N2 is true: there is a $\tau \in \mathbb{R}$ such that for all $t \in \mathbb{R}$, $f(t) = \sup f + \inf f - f(t + \tau)$. This means that f and $\sup f + \inf f - f$ are related by a time translation, i.e. $[f] = [\sup f + \inf f - f]$. Therefore, N1 is true, with $(a_1, b_1) = (0, 1)$ and $(a_2, b_2) = (\sup f + \inf f, -1)$.

Conversely, suppose that N1 is true: there exists two distinct pairs of real numbers (a_1, b_1) and (a_2, b_2) and a $\tau \in \mathbb{R}$ such that

$$\forall t \in \mathbb{R}, \qquad a_1 + b_1 f(t) = a_2 + b_2 f(t + \tau). \tag{A4}$$

The numbers b_1 and b_2 cannot be both zero, otherwise the above equation implies that $a_1 = a_2$, a contradiction. If $b_1 = 0$ and $b_2 \neq 0$, the above equation implies that f is a constant, in which case N2 is clearly true. Otherwise $b_1 \neq 0$ and we can re-write the above equality as

$$\forall t \in \mathbb{R}, \qquad f(t) = a + bf(t + \tau) \tag{A5}$$

with $a = (a_2 - a_1)/b_1$ and $b = b_2/b_1$. Now there are two possibilities: either $b > 0$ or $b < 0$.

First, let us suppose that $b > 0$. The above equality imposes the following conditions on the minimum and maximum values of the function f:

$$\sup f = a + b \sup f, \tag{A6}$$
$$\inf f = a + b \inf f. \tag{A7}$$

Subtracting (A7) from (A6) and reorganizing the terms we get $(1 - b)(\sup f - \inf f) = 0$. If $b = 1$, then $a = 0$, contradicting our assumption that (a_1, b_1) and (a_2, b_2) are distinct pairs. Therefore $\sup f = \inf f$, f is constant and N2 is true.

Second, let us suppose that $b < 0$. As before we have

$$\sup f = a + b \inf f, \tag{A8}$$
$$\inf f = a + b \sup f, \tag{A9}$$

and again $(1 + b)(\sup f - \inf f) = 0$. Either f is a constant, or $b = -1$, impliying in turn that $a = \sup f + \inf f$. Therefore, coming back to (A5), we have $f(t) = \sup f + \inf f - f(t + \tau)$ for all $t \in \mathbb{R}$, which is the statement of N2.

Appendix B: General Applicability of Frequency Propagation

Freq-prop applies to arbitrary physical networks: not only resistive networks, but also flow networks, capacitive networks and inductive networks, among others. In these networks, the notion of current-voltage characteristics will be replaced by current-pressure characteristics, current-flux characteristics, and charge-voltage characteristics, respectively. The mathematical framework for nonlinear elements (Section II) also applies to these networks, where the energy functions minimized at equilibrium are the co-content, the inductive energy and the capacitive co-energy, respectively ([15, 25]).

To emphasize the generality of Freq-prop, we present it here in the context of *central force spring networks* (or 'elastic networks') ([5]), as well as Hopfield networks (the Ising model).

a. Central force spring networks. We consider an elastic network of N nodes interconnected by springs. The elastic energy stored in the spring connecting node i to node j is $E_{ij}(r_{ij}) = \frac{1}{2}k_{ij}\left(r_{ij} - \ell_{ij}\right)^2$, where k_{ij} is the spring constant, ℓ_j is the spring's length at rest, and r_{ij} is the distance between nodes i and j. Nonlinear springs are also allowed and their energy terms are gathered in a unique term $E_{\mathrm{nonlinear}}(r)$. Thus, the total elastic energy stored in the network, which is minimized, is given by

$$E(\theta, r) = \frac{1}{2}\sum_{i,j} k_{ij}\left(r_{ij} - \ell_{ij}\right)^2 + E_{\mathrm{nonlinear}}(r), \tag{B1}$$

where $\theta = \{k_{ij}, \ell_{ij}\}$ is the set of adjustable parameters, and $r = \{r_{ij}\}$ plays the role of state variable.

In this setting as in the case of resistive networks, we apply a nudging signal $\gamma\,\sin(\omega t)\,C(r, y)$ at the output part of the network, we observe the response $r(t)$, and we assume that we can recover the first two vectors of Fourier coefficients of $r(t)$, i.e. the vectors a and b such that $a = \frac{1}{T}\int_0^T r(t)dt$ and $b = \frac{2}{T}\int_0^T r(t)\sin(\omega t)dt$. Then, the learning rules for the spring constant k_{ij} and the spring's length at rest ℓ_{ij} read, in this context,

$$\Delta k_{ij} = -\alpha\, b_{ij}\,(a_{ij} - \ell_{ij}), \qquad \Delta\ell_{ij} = -\alpha\, k_{ij}\, b_{ij}. \tag{B2}$$

Theorem 2 generalizes to this setting ; the above learning rules perform stochastic gradient descent on the loss: $\Delta\theta = -\alpha\gamma\frac{\partial\mathcal{L}}{\partial\theta}(\theta, x, y) + O(\gamma^3)$.

b. Continuous Hopfield networks. Freq-prop also applies to Hopfield networks (the Ising model) ([11, 26]). In a Hopfield network of multiple units interconnected by synapses, the energy term between unit i and unit j is $E_{ij} = w_{ij}h_ih_j$, where w_{ij} is the synaptic weight, and h_i is the state of unit i. The total energy is

$$E(\theta, h) = \frac{1}{2}\sum_{i,j} w_{ij}h_ih_j, \tag{B3}$$

where $\theta = \{w_{ij}\}$ is the set of adjustable parameters, and $h = \{h_i\}$ plays the role of state variable. After applying a nudging signal $\gamma\,\sin(\omega t)\,C(h, y)$ at a set of output units, we observe the response $u(t)$ (the state of the units at equilibrium), we compute the vectors a and b such that $a = \frac{1}{T}\int_0^T u(t)dt$ and $b = \frac{2}{T}\int_0^T u(t)\sin(\omega t)dt$. The learning rules for the weight w_{ij} reads

$$\Delta w_{ij} = -\alpha(a_ib_j + a_jb_i), \tag{B4}$$

which performs stochastic gradient descent on the loss, up to $O(\gamma^3)$.

Appendix C: Related Work

Frequency propagation builds on learning via *chemical signaling* ([1]), which is another example of multi-mechanism learning (MmL) in physical networks. Whereas MmL via frequency propagation uses two different frequencies to play the role of the *activation* and *error* signals during training, MmL via chemical signaling uses two different chemical concentrations for these signals.

[1] presents learning via chemical signaling in the setting of linear flow networks, which we extend here to the nonlinear setting (Appendix D).

Freq-prop is also related to *equilibrium propagation* (EP) ([3, 12]) and *coupled learning* ([5]). To see the relationship with these algorithms, we consider the case of resistive networks (section II). Denote $v_{jk} = v_j - v_k$ the voltage across branch (j, k). Further denote $v^\beta = \arg\min_v \; [E(\theta, x, v) + \beta\, C(v, y)]$ for any $\beta \in \mathbb{R}$. Based on a result from [12], [3] proved that the learning rule

$$\Delta^{\text{EP}}\theta_{jk} = \frac{\alpha}{2}\left((v_{jk}^0)^2 - (v_{jk}^\beta)^2\right) \tag{C1}$$

performs gradient descent with step size $\alpha\beta$, up to $O(\beta^2)$. We note that the right-hand side of (C1) is also equal to $\alpha\, v_{jk}^0 \left(v_{jk}^0 - v_{jk}^\beta\right) + O(\beta^2)$, showing that the gradient information is contained in the physical quantities v^0 and $\frac{\partial v^\beta}{\partial \beta}\Big|_{\beta=0}$. These quantities correspond to the activation and error signals of Freq-prop, respectively. To avoid the use of finite differences to measure $\frac{\partial v^\beta}{\partial \beta}\Big|_{\beta=0}$, Freq-prop makes use of a time-varying nudging signal $\beta(t) = \gamma \sin(\omega t)$. With this method, the activation and error signals are encoded in the frequencies 0 and ω of the response signal $v(t) = v^0 + \gamma \sin(\omega t)\, \frac{\partial v^\beta}{\partial \beta}\Big|_{\beta=0} + O(\gamma^3)$. The required information can thus be recovered via frequency analysis.

The idea of using an oscillating nudging signal was also proposed by [11] and more recently (concurrently to our work) in ‘holomorphic EP’ ([27]). Our work differs from these two other works in several ways. First, our learning rule can be decomposed as ‘activation signal’ times ‘error signal’ ($a \times b$), whereas the learning rule of [11] takes the form $\Delta\theta_{jk} = \frac{\alpha}{2} \int \sin(\omega t) v_{jk}(t)^2 dt$, and similarly for holomorphic EP. Second, our learning rule is proved to approximate the gradient of the cost function, up to $O(\beta^3)$, unlike in [11]. In [27], the authors exploit the Cauchy formula of complex calculus to prove that their algorithm computes the exact gradient of the cost function, independently of the strength of the nudging signal. To achieve this, the authors allow the nudging factor to take complex values, i.e. $\beta = \gamma e^{i\omega t} \in \mathbb{C}$, and the domain of definition of the energy function $v \mapsto E(\theta, x, v)$ is extended to complex configurations $v \in \mathbb{C}^N$. However, it is not straightforward to see how this mathematical formalism can be directly mapped to physical systems such as resistive networks or spring networks, which is the motivation of our work.

Another very recent work proposes an alternative approach to train physical systems by gradient descent called *agnostic equilibrium propagation* ([8]). However, this method imposes constraints on the nature of the parameters (θ), which must minimize the system’s energy (E), just like the state variables (v) do. This assumption does not allow us to view the conductances of resistors as trainable parameters in a resistive network. The method also requires control knobs with the ability to perform homeostatic control over the parameters. Our work can also be seen as a physical implementation of *implicit differentiation* in physical networks. We refer to ([28]) for a description of implicit differentiation where the authors use a mathematical formalism close to ours.

Lastly, other physical learning algorithms that make explicit use of time are being developed. For instance, recent work proposes a way to train physical systems with time reversible Hamiltonians ([6]). In this method called *Hamiltonian echo backpropagation* (HEB), the error signal is a time-reversed version – an “echo” – of the activation signal, with the cost function acting as a perturbation on this signal. However, HEB requires a feasible way to time-reverse the activation signal.

Appendix D: Multi-Mechanism Learning via Chemical Signaling

In this appendix, we generalize the learning algorithm via *chemical signaling* ([1]) to nonlinear networks. Learning via chemical signaling is another example of *multi-mechanism learning* in physical networks. It uses pressures and chemical concentrations to implement a local learning rule. This way of using multiple independent “mechanisms” is the central idea behind multi-mechanism learning.

Consider a flow network, i.e. a network of nodes interconnected by tubes. A flow network is formally equivalent to the resistive network of Section II, with v being the configuration of node pressures, and θ_{jk} being the conductance of the branch between nodes j and k.

Learning via chemical signaling proceeds as follows. In the first phase, given θ and input signals x, the configuration of node pressures stabilizes to its equilibrium value $v(\theta, x)$ given by

$$v(\theta, x) = \arg\min_v E(\theta, x, v). \tag{D1}$$

In the second phase, we inject chemical currents $e = -\beta \frac{\partial C}{\partial v}(v(\theta, x), y)$ at output nodes, where β is a (positive or negative) constant. As a result, a chemical concentration u develops at each node. Assuming that the configuration of node pressures $v(\theta, x)$ is not affected by the chemical, the chemical concentration u at equilibrium satisfies the relationship:

$$\frac{\partial^2 E}{\partial v^2}(\theta, x, v(\theta, x)) \cdot u = -\beta \frac{\partial C}{\partial v}(v(\theta, x), y). \tag{D2}$$

Indeed, diffusion along a tube follows the same equation as that of flow along the same tube, up to a constant factor (replacing node pressures and flow conductivity by chemical concentration and diffusion constant, respectively). When there is no ambiguity from the context, we write $v = v(\theta, x)$ for simplicity. We note that, although v is not affected by the chemical, u depends on v. In particular u also depends on θ and x through v.

Next, denoting $u = (u_1, u_2, \ldots, u_N)$, we update each parameter θ_{jk} according to the learning rule

$$\Delta\theta_{jk} = -\alpha(u_j - u_k) \cdot (v_j - v_k), \tag{D3}$$

where α is some constant. Note that this learning rule is local (just like the learning rule of Freq-prop), requiring only information about nodes j and k for each conductance θ_{jk}.

Theorem 4 *For every parameter θ_{jk}, it holds that*

$$\Delta\theta_{jk} = -\alpha\,\beta \frac{\partial \mathcal{L}}{\partial \theta_{jk}}(\theta, x, y). \tag{D4}$$

Namely, the learning rule of Eq. (D3) performs one step of gradient descent with respect to the loss, with step size $\alpha\beta$. We note that learning via chemical signaling comes in two variants, either with $\beta > 0$ and $\alpha > 0$, or with $\beta < 0$ and $\alpha < 0$. The procedure performs one step of gradient *descent* as long as the product $\alpha\beta$ is positive.

[Proof of Theorem 4] First, we write the first-order equilibrium condition for $v(\theta, x)$, which is

$$\frac{\partial E}{\partial v}(\theta, x, v(\theta, x)) = 0. \tag{D5}$$

We differentiate this equation with respect to θ:

$$\begin{aligned} &\frac{\partial^2 E}{\partial v^2}(\theta, x, v(\theta, x)) \frac{\partial v}{\partial \theta}(\theta, x) + \\ &\qquad\qquad \frac{\partial^2 E}{\partial v \partial \theta}(\theta, x, v(\theta, x)) = 0. \end{aligned} \tag{D6}$$

Multiplying both sides on the left by $u^\top$ we get

$$\begin{aligned} &u^\top \frac{\partial^2 E}{\partial v^2}(\theta, x, v(\theta, x)) \frac{\partial v}{\partial \theta}(\theta, x) + \\ &\qquad\qquad u^\top \frac{\partial^2 E}{\partial v \partial \theta}(\theta, x, v(\theta, x)) = 0. \end{aligned} \tag{D7}$$

On the other hand, multiplying both sides of (D2) on the left by $\frac{\partial v}{\partial \theta}(\theta, x)^\top$, we get

$$\begin{aligned} &\frac{\partial v}{\partial \theta}(\theta, x)^\top \frac{\partial^2 E}{\partial v^2}(\theta, x, v(\theta, x)) u \\ &\qquad = -\beta \frac{\partial v}{\partial \theta}(\theta, x)^\top \frac{\partial C}{\partial v}(v(\theta, x), y) \\ &\qquad = -\beta \frac{\partial \mathcal{L}}{\partial \theta}(\theta, x, y) \end{aligned} \tag{D8}$$

Comparing (D7) and (D8) we conclude that

$$u^\top \frac{\partial^2 E}{\partial v \partial \theta}(\theta, x, v(\theta, x)) = \beta \frac{\partial \mathcal{L}}{\partial \theta}(\theta, x, y). \tag{D9}$$

Finally, using the form of the energy function (2), we have for each parameter θ_{ij}

$$(u_i - u_j) \cdot (v_i - v_j) = \beta \frac{\partial \mathcal{L}}{\partial \theta_{ij}}(\theta, x, y). \tag{D10}$$

Therefore the learning rule

$$\Delta \theta_{jk} = -\alpha (u_i - u_j) \cdot (v_i - v_j) \tag{D11}$$

satisfies

$$\Delta \theta_{jk} = -\alpha \, \beta \frac{\partial \mathcal{L}}{\partial \theta_{jk}}(\theta, x, y). \tag{D12}$$

Hence the result.