A Theoretical proofs

In this appendix we detail the proofs of the theoretical results in the body text.

A.1 Complex analysis background

We recall here the minimal complex analysis background required to appreciate the theoretical results of this work. In the following, we recall the definitions of holomorphic and Wirtinger derivatives, the Cauchy-Riemann equations and the Cauchy formulas. We refer the reader to Chapter 4 of [S1] for proofs as well as an excellent introduction to complex analysis.

Definition 1 (Holomorphic function)**.** *Let U be an open set of $\mathbb{C}$ and $f : z \in U \mapsto f(z) \in \mathbb{C}$ a function. f is holomorphic at $a \in U$ if the limit*

$$\lim_{z \to a} \frac{f(z) - f(a)}{z - a}$$

exists. This limit is then noted $f'(a)$. f is holomorphic on U if it is holomorphic $\forall a \in U$.

Though this definition looks like the definition of differentiability in $\mathbb{R}$, it brings constraints on the underlying function $\tilde{f} : (x, y) \in \mathbb{R}^2 \mapsto (\mathrm{Re}(f(x + \mathrm{i}y), \mathrm{Im}(f(x + \mathrm{i}y))$. The added constraints are the Cauchy-Riemann equations, which can be compactly written after defining Wirtinger derivatives:

Definition 2 (Wirtinger derivatives)**.** *Noting $\partial/\partial x$ and $\partial/\partial y$ the usual partial derivatives in $\mathbb{R}^2$, the Wirtinger derivatives are defined by:*

$$\frac{\partial}{\partial z} := \frac{1}{2}\left(\frac{\partial}{\partial x} - \mathrm{i}\frac{\partial}{\partial y}\right), \qquad \frac{\partial}{\partial \bar{z}} := \frac{1}{2}\left(\frac{\partial}{\partial x} + \mathrm{i}\frac{\partial}{\partial y}\right).$$

In this way, z and its complex conjugate $\bar{z}$ can be thought of as independent variables. We can then state the Cauchy-Riemann equations as:

Theorem 2 (Cauchy-Riemann equations)**.** *if f is holomorphic at $a \in U$, then:*

$$\frac{\partial f}{\partial z}(a) = f'(a), \qquad \frac{\partial f}{\partial \bar{z}}(a) = 0. \tag{11}$$

These constraints ensure that f is locally expandable everywhere in U into a converging power series. In particular, it is differentiable at any order and the derivatives can be computed with the:

Theorem 3 (Cauchy formulas)**.** *Let f be holomorphic on U, let γ be any piece-wise continuously differentiable closed curve in U going around $a \in U$ once and counterclockwise, then:*

$$f^{(n)}(a) = \frac{n!}{2\mathrm{i}\pi} \oint_\gamma \frac{f(z)}{(z - a)^{n+1}} \mathrm{d}z. \tag{12}$$

A.2 Proof of Lemma 1

Here, we give a more detailed proof of holomorphic EP. We recall the:

Lemma 1 (Holomorphic Equilibrium Propagation)**.** *Let F be a scalar function governing the dynamics, so that the holomorphic implicit function theorem can be applied to the fixed point equation $\partial_{\mathbf{s}} F(\boldsymbol{\theta}, \mathbf{s}_0^*, 0) = 0$, then the gradient formula of equilibrium propagation (Eq.* (2)*) holds in the sense of complex differentiation.*

Proof. We first detail precisely the set of equations on which the holomorphic implicit function theorem is applied. At the free fixed point $(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \beta = 0)$ that which exists by assumption, we have the following set of equations:

$$\frac{\partial F}{\partial s_j}(\boldsymbol{\theta}_0, \mathbf{s}_0^*, 0) = 0, \quad 1 \leq j \leq n,$$

where n is the number of units in the system. The functions $\partial_{s_j} F$ are holomorphic by assumption. If we further assume that the Hessian of F with respect to $\mathbf{s}$ is invertible in $(\boldsymbol{\theta}_0, \mathbf{s}_0^*, 0)$, i.e.:

$$\det\left(\frac{\partial^2 F}{\partial s_i \partial s_j}(\boldsymbol{\theta}_0, \mathbf{s}_0^*, 0)\right)_{i,j} \neq 0,$$

then the holomorphic version of the implicit function theorem [S2] can be applied and there exists an open neighbourhood of $(\boldsymbol{\theta} = \boldsymbol{\theta}_0, \beta = 0)$ in the complex domain where the implicit map $(\boldsymbol{\theta}, \beta) \mapsto \mathbf{s}^*_{\boldsymbol{\theta},\beta}$ is holomorphic, and where the fixed point equations hold:

$$\frac{\partial F}{\partial \mathbf{s}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) = 0.$$

At such fixed points, we have that the total derivatives of F with respect to either β or $\boldsymbol{\theta}$ are equal to the partial derivatives, which can be seen by applying the chain rule of complex differentiation using Wirtinger derivatives. There are now in principle three contributions to the total derivative of F with respect to β:

$$\frac{\mathrm{d}F}{\mathrm{d}\beta}(\boldsymbol{\theta}, \mathbf{s}_{\boldsymbol{\theta},\beta}, \beta) = \frac{\partial F}{\partial \beta}(\boldsymbol{\theta}, \mathbf{s}_{\boldsymbol{\theta},\beta}, \beta) + \underbrace{\frac{\partial F}{\partial \mathbf{s}}}_{=0 \text{ at a fixed point}} \cdot \frac{\partial \mathbf{s}}{\partial \beta}(\boldsymbol{\theta}, \beta) + \underbrace{\frac{\partial F}{\partial \overline{\mathbf{s}}}}_{=0 \text{ by Cauchy-Riemann (Eq. (11))}} \cdot \frac{\partial \overline{\mathbf{s}}}{\partial \beta}(\boldsymbol{\theta}, \beta), \tag{13}$$

where $\overline{\mathbf{s}}$ denotes the complex conjugate of $\mathbf{s}$. At the fixed point however, the second term on the right hand side cancels by definition. The third term is zero because F is holomorphic, i.e., its derivative with respect to the conjugate variable $\overline{\mathbf{s}}$ is zero according to the Cauchy-Riemann condition [S1]. The same argument holds for the total derivative with respect to $\boldsymbol{\theta}$.

Finally, the cross-derivatives of F with respect to complex β and $\boldsymbol{\theta}$ can be exchanged, which is a consequence of the Schwarz theorem applied to the function $(\boldsymbol{\theta}, \beta) \mapsto G(\boldsymbol{\theta}, \beta) := F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta)$. Therefore we have that:

$$\begin{aligned}
\frac{\partial^2 G}{\partial \beta \partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \beta) &= \frac{\partial^2 G}{\partial \boldsymbol{\theta} \partial \beta}(\boldsymbol{\theta}, \beta), \\
\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) &= \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\frac{\mathrm{d}}{\mathrm{d}\beta} F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta), \\
\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{\partial}{\partial \boldsymbol{\theta}} F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) &= \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\frac{\partial}{\partial \beta} F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta), \quad \text{by Eq. (13).}
\end{aligned}$$

By then applying this equality in $\beta = 0$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, we obtain the EP gradient formula (Eq. (2)) for complex differentiation:

$$\left.\frac{\mathrm{d}}{\mathrm{d}\beta}\right|_{\beta=0} \left(\frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta)\right) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\frac{\partial F}{\partial \beta}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}},$$

which concludes the proof. □

A.3 Proof of Theorem 1

Theorem 1 (Exact gradient from finite teaching signals)**.** *Assuming that the conditions of Lemma 1 are met and let $|\beta| > 0$ be the radius of a circular path around 0 in $\mathbb{C}$ contained in the open set U on which the fixed point $\mathbf{s}^*_{\boldsymbol{\theta},\beta}$ is defined. Further assume that this path is parameterized by $t \in [0, T] \mapsto \beta(t) = |\beta| e^{2\mathrm{i}\pi t/T}$, where i is the imaginary unit. Then the loss gradient is given by:*

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{1}{T|\beta|}\int_0^T \frac{\partial F}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta(t)}, \beta(t)\right) e^{-2\mathrm{i}\pi t/T}\mathrm{d}t \quad . \tag{14}$$

Proof. By assumption the fixed point $\beta \mapsto \mathbf{s}^*_{\boldsymbol{\theta},\beta}$ is defined on an open set U (by the holomorphic implicit function theorem) containing the disk of radius $|\beta|$ centered around 0. In particular, the function $\beta \in U \mapsto \partial_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta)$, is also holomorphic by composition. The left hand side of

Eq. (4) can thus be computed with the Cauchy formulas (Eq. (12) with $f = \partial_{\boldsymbol{\theta}} F$, $n = 1$, $a = 0$), and γ an arbitrary closed path leading around zero once and counterclockwise in U:

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\bigg|_{\beta=0} \left(\frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) \right) = \frac{1}{2\mathrm{i}\pi} \oint_\gamma \frac{1}{\beta^2} \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) \mathrm{d}\beta \quad . \tag{15}$$

To obtain Eq. (14), we choose γ as a circular path in the complex plane with radius $|\beta| > 0$ parameterized by time $t \in [0, T] \mapsto \beta(t) = |\beta| e^{2\mathrm{i}\pi t/T}$, where T is a full period. After the change of variable $\mathrm{d}\beta = \left(2\mathrm{i}\pi\beta(t)/T\right)\mathrm{d}t$ in Eq. (15), and using Lemma 1, the loss gradient is given by:

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} &= \frac{1}{2\mathrm{i}\pi} \oint_\gamma \frac{1}{\beta^2} \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta}, \beta) \mathrm{d}\beta \\
&= \frac{1}{2\mathrm{i}\pi} \int_0^T \frac{1}{\beta(t)^2} \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta(t)}, \beta(t)) \left(\frac{2\mathrm{i}\pi\beta(t)}{T} \right) \mathrm{d}t \\
&= \frac{1}{T} \int_0^T \frac{1}{\beta(t)} \frac{\partial F}{\partial \boldsymbol{\theta}} \left(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta(t)}, \beta(t) \right) \mathrm{d}t \\
&= \frac{1}{T|\beta|} \int_0^T \frac{\partial F}{\partial \boldsymbol{\theta}} \left(\boldsymbol{\theta}, \mathbf{s}^*_{\boldsymbol{\theta},\beta(t)}, \beta(t) \right) e^{-2\mathrm{i}\pi t/T} \mathrm{d}t.
\end{aligned}$$

□

A.4 Roles of real and imaginary parts in the learning rule

Recall that for the continuous Hopfield network case the partial derivative of F with respect to a parameter w_{ij} is the product of pre and post activation (Eq. (6)), so that applying Eq. (4) yields:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w_{ij}} = \frac{\mathrm{d}}{\mathrm{d}\beta}\bigg|_{\beta=0} \underbrace{\left(\frac{\partial F}{\partial w_{ij}}(\boldsymbol{\theta}, \mathbf{s}^*_\beta, \beta) \right)}_{= -\sigma(s^*_{i,\beta})\sigma(s^*_{j,\beta})} = - \left. \frac{\mathrm{d}\left(\sigma(s^*_{i,\beta})\sigma(s^*_{j,\beta}) \right)}{\mathrm{d}\beta} \right|_{\beta=0},$$

which can further be expressed as:

$$\left. \frac{\mathrm{d}\left(\sigma(s^*_{i,\beta})\sigma(s^*_{j,\beta}) \right)}{\mathrm{d}\beta} \right|_{\beta=0} = \left(\sigma(s^*_{i,\beta}) \frac{\mathrm{d}\sigma(s^*_{j,\beta})}{\mathrm{d}\beta} \right)\bigg|_{\beta=0} + \left(\sigma(s^*_{j,\beta}) \frac{\mathrm{d}\sigma(s^*_{i,\beta})}{\mathrm{d}\beta} \right)\bigg|_{\beta=0}. \tag{16}$$

Using the same assumptions as Section 3, the map $\beta \in U \mapsto s^*_{i,\beta}$ is holomorphic, and so is the map $\beta \in U \mapsto \sigma(s^*_{i,\beta})$ by composition. We can thus expand it in a power series around zero:

$$\sigma(s^*_{i,\beta}) = \sum_{k=0}^{\infty} \frac{\beta^k}{k!} \left. \frac{\mathrm{d}^k \sigma(s^*_{i,\beta})}{\mathrm{d}\beta^k} \right|_{\beta=0}.$$

We can then separate the sum into the real and imaginary parts because the series converge absolutely. Assuming that $\beta = |\beta| e^{2\mathrm{i}\pi t/T}$, and applying the Euler formula, we obtain:

$$\begin{aligned}
\mathrm{Re}\left(\sigma(s^*_{i,\beta}) \right) &= \sum_{k=0}^{\infty} \cos\left(\frac{2k\pi t}{T} \right) \frac{|\beta|^k}{k!} \left. \frac{\mathrm{d}^k \sigma(s^*_{i,\beta})}{\mathrm{d}\beta^k} \right|_{\beta=0}, \\
\mathrm{Im}\left(\sigma(s^*_{i,\beta}) \right) &= \sum_{k=1}^{\infty} \sin\left(\frac{2k\pi t}{T} \right) \frac{|\beta|^k}{k!} \left. \frac{\mathrm{d}^k \sigma(s^*_{i,\beta})}{\mathrm{d}\beta^k} \right|_{\beta=0}.
\end{aligned} \tag{17}$$

Therefore, the first derivative ($k = 1$) with respect to β in Eq. (16) can be obtained by either projecting the real part against the cosine function, or imaginary part against the sine function:

$$\begin{aligned}
\left. \frac{\mathrm{d}\sigma(s^*_{i,\beta})}{\mathrm{d}\beta} \right|_{\beta=0} &= \frac{2}{|\beta|T} \int_0^T \mathrm{Re}\left(\sigma(s^*_{i,\beta}) \right) \cos\left(\frac{2\pi t}{T} \right) \mathrm{d}t, \\
&= \frac{2}{|\beta|T} \int_0^T \mathrm{Im}\left(\sigma(s^*_{i,\beta}) \right) \sin\left(\frac{2\pi t}{T} \right) \mathrm{d}t,
\end{aligned}$$

by orthogonality of the family $\left((t \mapsto \cos\left(\frac{2k\pi t}{T}\right))_{k\geq 0}, (t \mapsto \sin\left(\frac{2k\pi t}{T}\right))_{k\geq 0}\right)$ in $L^2[0,T]$. Note that the higher order derivatives with respect to β can be obtained as well by projecting against the corresponding cosine or sine function. The same holds for index j by symmetry. As an interesting final note, if we define $\mathrm{Re}_1(\sigma(s^*_{i,\beta}))$ and $\mathrm{Im}_1(\sigma(s^*_{i,\beta}))$, the first order contributions in β to the real and imaginary parts of the neural activity, we find that they are the only ones to contribute to the gradient computation. We can appreciate that they are related through $\mathrm{Im}_1(\sigma(s^*_{i,\beta})) = -\frac{T}{2\pi}\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Re}_1(\sigma(s^*_{i,\beta}))$, where the time derivative is at the scale of the teaching signal.

A.5 Derivation of the bias term

Recall the definition of $\beta_k := |\beta| e^{2\mathrm{i}\pi k/N}$, for $k \in [0, ..., N-1]$, $N \geq 2$, and the gradient estimate (Eq. (8)):

$$\hat{\nabla}(N) := \frac{1}{N|\beta|}\sum_{k=0}^{N-1} \frac{\partial F}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}, \mathbf{s}^*_{\beta_k}, \beta_k\right) e^{-2\mathrm{i}\pi k/N}.$$

For simplicity of notation, we rewrite $\partial_{\boldsymbol{\theta}} F(\beta) := \frac{\partial F}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}, \mathbf{s}^*_\beta, \beta)$. The function $\beta \mapsto \partial_{\boldsymbol{\theta}} F(\beta)$ is holomorphic on an open set U including zero, and so is $\beta \mapsto \mathbf{s}^*_\beta$ by the holomorphic implicit function theorem. We assume the β_k are included in U, so that we can expand $\partial_{\boldsymbol{\theta}} F(\beta_k)$ in a power series around zero:

$$\partial_{\boldsymbol{\theta}} F(\beta_k) = \sum_{p=0}^{\infty} \frac{\beta_k^p}{p!}\left[\frac{\mathrm{d}^p}{\mathrm{d}\beta^p}\partial_{\boldsymbol{\theta}} F\right](0),$$

we define $C_p := \left[\frac{\mathrm{d}^p}{\mathrm{d}\beta^p}\partial_{\boldsymbol{\theta}} F\right](0)$. The quantity of interest is C_1, since it is the gradient of the loss (Eq. (4))

$$\begin{aligned}
\partial_{\boldsymbol{\theta}} F(\beta_k) &= C_0 + \beta_k C_1 + \sum_{p=2}^{\infty}\frac{\beta_k^p}{p!}C_p \\
\frac{\partial_{\boldsymbol{\theta}} F(\beta_k)}{\beta_k} &= C_0\beta_k^{-1} + C_1 + \sum_{p=2}^{\infty}\frac{\beta_k^{p-1}}{p!}C_p \\
\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial_{\boldsymbol{\theta}} F(\beta_k)}{\beta_k} &= C_1 + C_0\frac{1}{N}\sum_{k=0}^{N-1}\beta_k^{-1} + \frac{1}{N}\sum_{k=0}^{N-1}\sum_{p=2}^{\infty}\frac{\beta_k^{p-1}}{p!}C_p.
\end{aligned}$$

The sum symbols on the right can be interchanged thanks to the absolute convergence of the power series.

$$\begin{aligned}
\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial_{\boldsymbol{\theta}} F(\beta_k)}{\beta_k} &= C_1 + C_0\frac{1}{N}\sum_{k=0}^{N-1}\beta_k^{-1} + \sum_{p=2}^{\infty}\frac{C_p}{p!}\frac{1}{N}\sum_{k=0}^{N-1}\beta_k^{p-1} \\
\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial_{\boldsymbol{\theta}} F(\beta_k)}{\beta_k} &= C_1 + C_0\frac{1}{N|\beta|}\sum_{k=0}^{N-1}e^{-2\mathrm{i}\pi k/N} + \sum_{p=1}^{\infty}\frac{C_{p+1}}{(p+1)!}\frac{|\beta|^p}{N}\sum_{k=0}^{N-1}e^{2\mathrm{i}\pi pk/N}.
\end{aligned}$$

It remains to evaluate the geometric sums of the form $\sum_{k=0}^{N-1} e^{2\mathrm{i}\pi pk/N}$ for $p = -1$ and $p \geq 1$. If N divides p, i.e $p \equiv 0\ (N)$, then we can write $p = Nq$ and we have:

$$\sum_{k=0}^{N-1} e^{2\mathrm{i}\pi pk/N} = \sum_{k=0}^{N-1} e^{2\mathrm{i}\pi qNk/N} = \sum_{k=0}^{N-1} e^{2\mathrm{i}\pi qk} = \sum_{k=0}^{N-1} 1 = N.$$

If N does not divide p, then the geometric sum of ratio $e^{2\mathrm{i}\pi p/N}$ can be computed:

$$\sum_{k=0}^{N-1} e^{2\mathrm{i}\pi pk/N} = \frac{1-(e^{2\mathrm{i}\pi p/N})^N}{1-e^{2\mathrm{i}\pi p/N}} = \frac{1-e^{2\mathrm{i}\pi p}}{1-e^{2\mathrm{i}\pi p/N}} = \frac{1-1}{1-e^{2\mathrm{i}\pi p/N}} = 0.$$

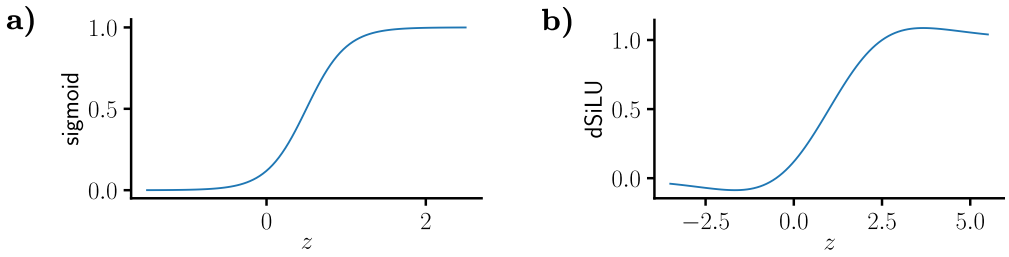

Figure 5: **a)** The shifted sigmoid we used in multi-layered perceptrons experiments. **b)** The dSiLU we used in CNNs experiments.

We thus have that:

$$\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial_{\boldsymbol{\theta}}F(\beta_k)}{\beta_k} = C_1 + C_0\frac{1}{N|\beta|}\underbrace{\sum_{k=0}^{N-1}e^{-2\mathrm{i}\pi k/N}}_{=0} + \sum_{p=1}^{\infty}\frac{C_{p+1}}{(p+1)!}\frac{|\beta|^p}{N}\underbrace{\sum_{k=0}^{N-1}e^{2\mathrm{i}\pi pk/N}}_{=0 \text{ when } p\not\equiv 0\ (N)}$$

$$\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial_{\boldsymbol{\theta}}F(\beta_k)}{\beta_k} = C_1 + \sum_{p\equiv 0\ (N)}^{\infty}\frac{C_{p+1}|\beta|^p}{(p+1)!},$$

which is the result of Eq. (9).

A.6 Derivation of the online estimate

Recall the formula of the online estimate (Eq. (10)):

$$\widetilde{\nabla}(T_{\text{plas}}) := -\frac{1}{T_{\text{plas}}|\beta|}\int_0^{T_{\text{plas}}}\sigma_i(t)\sigma_j(t)e^{-2\mathrm{i}\pi t/T_{\text{osc}}}\mathrm{d}t.$$

If $T_{\text{dyn}} \ll T_{\text{osc}}$, the product of activities can be replaced by its value at the fixed point, and an exact gradient is computed after each period (Eq. (7)). Then if $T_{\text{osc}} \ll T_{\text{plas}}$, the integral can be divided into an integer amount of completed periods plus a remainder: $T_{\text{plas}} = kT_{\text{osc}} + T_{\text{rem}}$, where $k \in \mathbb{N}$ and $T_{\text{rem}} < T_{\text{osc}}$. We then have by periodicity that:

$$\widetilde{\nabla}(T_{\text{plas}}) = \underbrace{\frac{kT_{\text{osc}}}{kT_{\text{osc}}+T_{\text{rem}}}}_{\to 1 \text{ when } T_{\text{plas}}\to\infty}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w_{ij}} - \underbrace{\frac{1}{kT_{\text{osc}}+T_{\text{rem}}}}_{\to 0 \text{ when } T_{\text{plas}}\to\infty}\frac{1}{|\beta|}\int_0^{T_{\text{rem}}}\sigma_i^*(t)\sigma_j^*(t)e^{-2\mathrm{i}\pi t/T_{\text{osc}}}\mathrm{d}t.$$

In this way, when averaging over large T_{plas}, the number of completed cycles outweighs the current period. Thus, by simply averaging over many oscillation cycles allows estimating gradients without explicit separate phases.

B Detailed architecture

B.1 Dynamics for multi-layer perceptrons

Assuming a number of L layers, we note $\mathbf{s}_l$ the subset of units in layer l, with $\mathbf{s}_0 = \mathbf{x}$ and $\mathbf{y}$ the one hot class label. We note $\mathbf{W}_l$, and $\mathbf{b}_l$ the weight and biases of layer $l \geq 1$. The energy function F for a MLP optimizing the cross entropy loss reads:

$$F(\boldsymbol{\theta}, \mathbf{s}, \beta, \mathbf{y}) = \sum_{l=1}^{L-1}\left[\frac{1}{2}\|\mathbf{s}_l\|^2 - \sigma(\mathbf{s}_{l-1})^\top\cdot\mathbf{W}_l\cdot\sigma(\mathbf{s}_l) - \mathbf{b}_l^\top\cdot\sigma(\mathbf{s}_l)\right] - \beta\mathbf{y}^\top\cdot\log(\mathbf{s}_L).$$

The activation function used for multi-layer perceptrons is the shifted sigmoid $z \mapsto 1/(1+e^{-4z+2})$ (Fig. 5a). We use the layer-wise discrete dynamics introduced by [S3, S4], which read:

$$\begin{cases} \mathbf{s}_l & \leftarrow & \sigma\left(\mathbf{W}_l \mathbf{s}_{l-1} + {\mathbf{W}_{l+1}}^\top \mathbf{s}_{l+1} + \mathbf{b}_l + \boldsymbol{\eta}_l\right), & \text{for } 1 \leq l \leq L-2 \\ \mathbf{s}_{L-1} & \leftarrow & \sigma\left(\mathbf{W}_{L-1} \mathbf{s}_{L-2} + \beta {\mathbf{W}_L}^\top (\mathbf{y} - \mathbf{s}_L) + \mathbf{b}_{L-1} + \boldsymbol{\eta}_{L-1}\right), & \\ \mathbf{s}_L & \leftarrow & \mathrm{Softmax}\left(\mathbf{W}_L \mathbf{s}_{L-1} + \mathbf{b}_L\right), & \end{cases}$$

where $\boldsymbol{\eta}_l$ is an optional Gaussian noise added for Fig. 3c and Table 1. The noise was sampled at each time step.

B.2 Dynamics for convolutional neural networks

The activation function used for CNNs is a sigmoid-weighted linear unit [S5] (Fig. 5b):

$$\mathrm{dSiLU}(z) := \left(\frac{z}{2}\right) \frac{1}{1+e^{-z}} + \left(1 - \frac{z}{2}\right) \frac{1}{1+e^{-z+2}}.$$

We denote by $\mathcal{P}$ the pooling operation, and $\tilde{\mathcal{P}}$ the corresponding unpooling operation. '$*$' denotes the convolution when preceded by $\mathbf{W}$ and transpose convolution when preceded by $\mathbf{W}^\top$. The energy function F for a CNN optimizing the cross entropy loss reads:

$$F(\boldsymbol{\theta}, \mathbf{s}, \beta, \mathbf{y}) = \sum_{l \in \{\text{Conv}\}} \left[\frac{1}{2}\|\mathbf{s}_l\|^2 - \sigma(\mathbf{s}_{l-1})^\top \cdot \mathcal{P}(\mathbf{W}_l * \sigma(\mathbf{s}_l)) - \mathbf{b}_l^\top \cdot \sigma(\mathbf{s}_l)\right]$$
$$\sum_{l \in \{\text{FC}\}} \left[\frac{1}{2}\|\mathbf{s}_l\|^2 - \sigma(\mathbf{s}_{l-1})^\top \cdot \mathbf{W}_l \cdot \sigma(\mathbf{s}_l) - \mathbf{b}_l^\top \cdot \sigma(\mathbf{s}_l)\right] - \beta \mathbf{y}^\top \cdot \log(\mathbf{s}_L).$$

We use the layer-wise discrete dynamics introduced by [S3, S4], which read:

$$\begin{cases} \mathbf{s}_l & \leftarrow & \sigma\left(\mathcal{P}(\mathbf{W}_l * \mathbf{s}_{l-1}) + \mathbf{W}_{l+1}^\top * \tilde{\mathcal{P}}(\mathbf{s}_{l+1}) + \mathbf{b}_l\right), & \text{for } l \in \{\text{Conv layers}\} \\ \mathbf{s}_l & \leftarrow & \sigma\left(\mathbf{W}_l \mathbf{s}_{l-1} + {\mathbf{W}_{l+1}}^\top \mathbf{s}_{l+1} + \mathbf{b}_l\right), & \text{for } l \in \{\text{FC layers}\} \\ \mathbf{s}_{L-1} & \leftarrow & \sigma\left(\mathbf{W}_{L-1} \mathbf{s}_{L-2} + \beta {\mathbf{W}_L}^\top (\mathbf{y} - \mathbf{s}_L) + \mathbf{b}_{L-1}\right), & \\ \mathbf{s}_L & \leftarrow & \mathrm{Softmax}(\mathbf{W}_L \mathbf{s}_{L-1} + \mathbf{b}_L). & \end{cases}$$

We used Softmax pooling [S6] with a tunable temperature τ, instead of the non-holomorphic Max pooling. The output y of Softmax pooling of an input $\mathbf{x}$ is defined for a kernel neighbourhood $\mathbf{R}$ by:

$$y = \sum_{i \in \mathbf{R}} \left(\frac{e^{x_i/\tau}}{\sum_{j \in \mathbf{R}} e^{x_j/\tau}}\right) x_i.$$

Note that Softmax pooling interpolates between Average pooling ($\tau \to \infty$) and Max pooling ($\tau \to 0$).

C Layer-wise comparison of the gradient in a deep network

Here we show in Fig. 6 the complete layer-wise cosine similarities between the estimates of holomorphic EP for various N and the true gradient computed by automatic differentiation.

D Dynamical stability in the complex plane

We show in Fig. 7 how the area in $\mathbb{C}$ where the fixed point empirically exists varies with different architecture choices. The data used for each panel is a digit from the MNIST dataset. As in Fig.2b), dark blue means that the fixed point exists, whereas light areas denote divergence. These diverging areas could be due to the poles of the activation functions used. For example, the sigmoid function

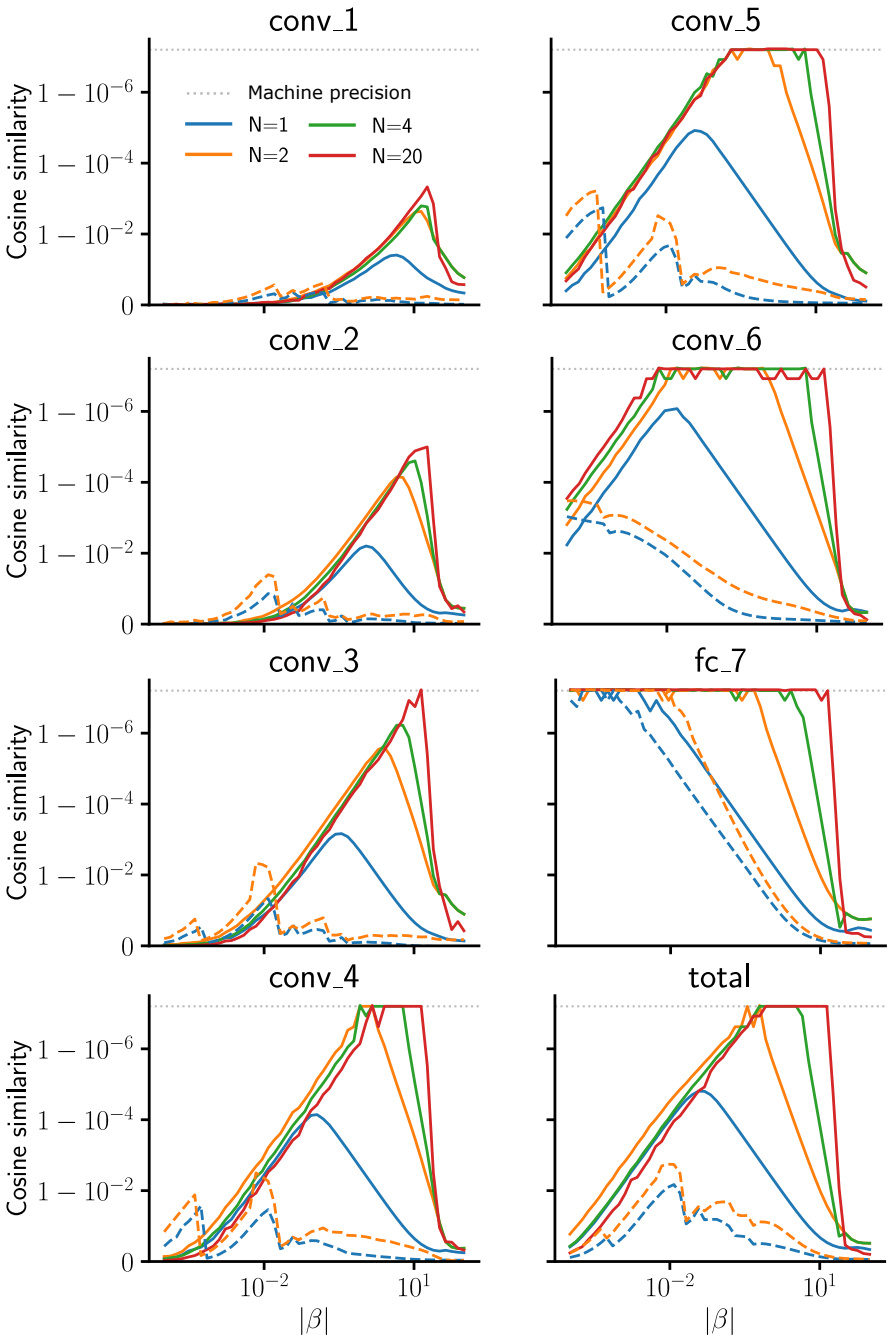

Figure 6: The complete layer-wise cosine similarity of Fig. 4a).

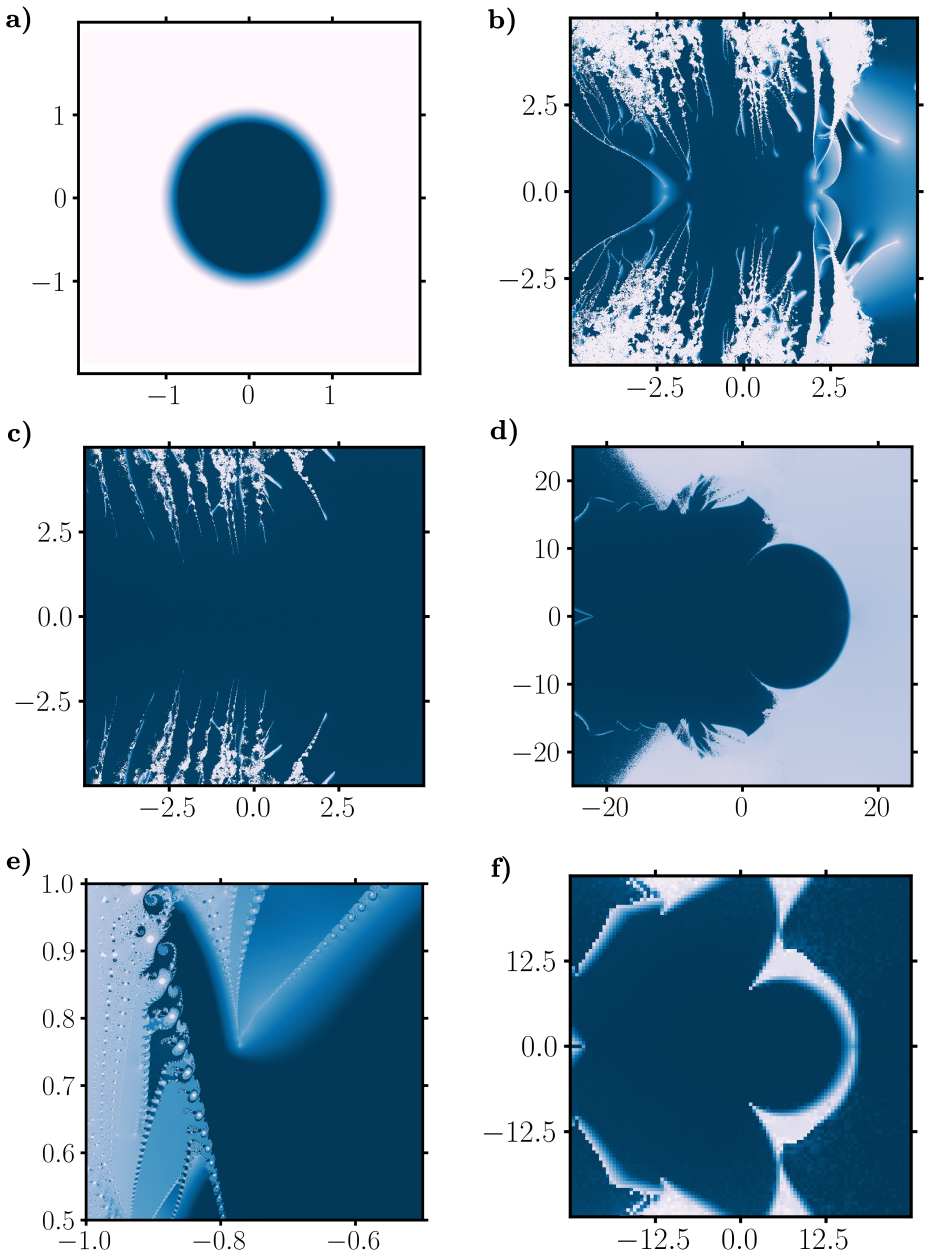

Figure 7: Map of convergence to a fixed point for complex β in various settings. **a)** MLP with linear activation function and low weight initialization. **b)** MLP with shifted sigmoid activation and default weight initialization. **c)** Same as b) but with reduced weight initialization. **d)** Same as b) but with dSiLU activation function. **e)** Zoom at a frontier between stable and unstable regions. **f)** Small CNN with dSiLU activation and Softmax pooling.

$z \mapsto 1/(1+e^{-z})$ has $\{(2k+1)\mathrm{i}\pi; k \in \mathbb{Z}\}$ as a set of poles where it diverges. Although we did not systematically study this phenomenon in this work, we strongly suspect that these unstable areas are partly the result of the teaching signal being too strong or the weights being poorly distributed, thereby driving the complex neural activities near to the poles. To some extent, the poles can be brought farther by introducing a coefficient in the exponential, but it results in flatter activation functions on the real axis, so a trade-off should be found. In practice, we found that choosing reasonably the activation function, weight initialization, and teaching radius $|\beta|$ lead to enough stable areas around 0 to compute the gradient.

E Hyperparameters

E.1 MNIST experiments

The digits were rescaled by 255 and flattened. The hyperparameters used for training are reported in Table 3, and the training errors are reported in Table 4.

Table 3: Hyperparameters used for the MNIST training experiment of Table 1.

Hyperparameter	Classic EP	hEP	Online hEP
Batch size	20	20	20
Learning rate	5e-2	5e-2	5e-2
Epochs	50	50	50
$\lvert\beta\rvert$	0.1 and 0.4	0.4	0.4
T_{free}	350	200	200*
T_{nudge}	350	50	N/A
T_{osc}	N/A	N/A	300
T_{plas}	N/A	N/A	900
N	N/A	10	10
Noise**	4e-2	4e-2	4e-2

* Only used for evaluation

** Standard deviation of the Gaussian noise for experiments with noise.

Table 4: MNIST training and validation errors for classic EP [10], hEP, and online hEP, with and without noise. All results are averages ($n = 3$) $\pm$ one standard deviation.

	Class. EP, $\lvert\beta\rvert = 0.1$		Class. EP, $\lvert\beta\rvert = 0.4$		hEP, $\lvert\beta\rvert = 0.4]$		Online hEP	
Noise	Train (%)	Val (%)	Train (%)	Val (%)	Train (%)	Val (%)	Train (%)	Val. (%)
No	0.05 $^{\pm 0.02}$	1.87 $^{\pm 0.01}$	0.19 $^{\pm 0.05}$	2.24 $^{\pm 0.05}$	0.02 $^{\pm 0.01}$	1.97 $^{\pm 0.08}$	0.11 $^{\pm 0.01}$	2.05 $^{\pm 0.02}$
Yes	88.8 $^{\pm 0.0}$	88.7 $^{\pm 0.0}$	1.96 $^{\pm 0.2}$	3.01 $^{\pm 0.1}$	0.14 $^{\pm 0.03}$	1.96 $^{\pm 0.07}$	0.13 $^{\pm 0.03}$	1.91 $^{\pm 0.16}$

E.2 Large-scale experiments

In the training experiments for CIFAR-10, CIFAR-100, and ImageNet 32×32, the training data was normalized, then augmented with 50% chance random horizontal flips, resized to 36×36 resolution with padding, and cropped randomly back to 32×32. The optimizer used was stochastic gradient descent with momentum. Pooling was applied at all layers for the five-layer CNN, and every other layer starting with the first layer in the seven-layer CNN.

F Simulations details

All simulations were performed on an in-house GPU cluster or workstations. Each simulation in Table 2 was run in parallel on four NVIDIA V100 GPUs. The training runs on ImageNet 32×32 took 5.5 days each for EP, and a few hours for BP. The runs on CIFAR-10 and CIFAR-100 took one day on average depending on the architecture (5 or 7 layers) and the number of time steps used for

Table 5: Hyperparameters used for the VGG training experiments of Table 2 and Fig.4c.

Hyperparameter	CIFAR-10	CIFAR-100	ImageNet 32×32	CIFAR-10 (Fig.4c)
Batch size	128	128	256	128
Channel sizes	[128, 256, 512, 512]			[128, 128, 256, 256, 512, 512]
Kernel sizes	[3, 3, 3, 3]			[3, 3, 3, 3, 3, 3]
Strides	[1, 1, 1, 1]			[1, 1, 1, 1, 1, 1]
Paddings	[1, 1, 1, 0]			[1, 1, 1, 0, 1, 0]
SoftPool window	2×2			2×2
SoftPool stride	2			2
SoftPool temp.	1			10
Initial LRs*	[25, 15, 10, 8, 5] $\times$ 1e-2			[5, 4, 4, 3, 3, 2, 2] $\times$ 1e-2
Final LRs	[25, 15, 10, 8, 5] $\times$ 1e-9			[5, 4, 4, 3, 3, 2, 2] $\times$ 1e-9
Weight decay	2e-3	1e-2	5e-4	[5, 5, 5, 5, 5, 5, 10] $\times$ 1e-4
Momentum	0.9	0.9	0.9	0.9
Epochs	90	90	90	90
$\lvert\beta\rvert$	1.0	1.0	1.0	1.0
T_{free}	250	250	250	260
T_{nudge}	60	60	60	60
N	2	2	2	2 and 4

* Learning rates were decayed with cosine annealing without restart [S7].

the dynamics. The use of complex numbers, although seamlessly implementable with Jax, results in longer simulation times due to the 64 bit-precision requirement (32 bit-precision for real and imaginary parts respectively).

Supplementary References

[S1] Walter Appel. Mathematics for physics and physicists. 2007.

[S2] Henri Cartan. *Théorie élémentaire des fonctions analytiques d'une ou plusieurs varibales complexes: Avec le concours de Reiji Takahashi*. Hermann, 1961.

[S3] Maxence Ernoult, Julie Grollier, Damien Querlioz, Yoshua Bengio, and Benjamin Scellier. Updates of equilibrium prop match gradients of backprop through time in an rnn with static input. *Advances in neural information processing systems*, 32, 2019.

[S4] Axel Laborieux, Maxence Ernoult, Benjamin Scellier, Yoshua Bengio, Julie Grollier, and Damien Querlioz. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:129, 2021.

[S5] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[S6] A Stergiou, R Poppe, and G Kalliatakis. Refining activation downsampling with softpool. arxiv 2021. *arXiv preprint arXiv:2101.00440*.

[S7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.