Self-Learning Machines Based on Hamiltonian Echo Backpropagation

Víctor López-Pastor[1] and Florian Marquardt[1,2]

[1]*Max Planck Institute for the Science of Light, 91058 Erlangen, Germany*
[2]*Physics Department, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany*

(Received 26 May 2021; revised 4 June 2023; accepted 11 July 2023; published 18 August 2023)

A physical self-learning machine can be defined as a nonlinear dynamical system that can be trained on data (similar to artificial neural networks) but where the update of the internal degrees of freedom that serve as learnable parameters happens autonomously. In this way, neither external processing and feedback nor knowledge of (and control of) these internal degrees of freedom is required. We introduce a general scheme for self-learning in any time-reversible Hamiltonian system. It relies on implementing a time-reversal operation and injecting a small error signal on top of the echo dynamics. We show how the physical dynamics itself will then lead to the required gradient update of learnable parameters, independent of the details of the Hamiltonian. We illustrate the training of such a self-learning machine numerically for the case of coupled nonlinear wave fields and other examples.

DOI: 10.1103/PhysRevX.13.031020

Subject Areas: Complex Systems,
Interdisciplinary Physics,
Nonlinear Dynamics

I. INTRODUCTION

In the last decade, the field of machine learning (ML) has experienced an explosive growth, finding use in an ever-increasing number of applications in our everyday lives, from automatic driving to face recognition. To a large degree, this astonishing progress in ML can be attributed to the developments in artificial neural networks (ANN). Training deep ANNs has only recently become possible in practice [1], thanks to both the availability of large data sets and the continued improvement in digital electronic hardware and the advent of fast graphical processing units (GPU) and other specialized hardware. However, while the demand for faster and more efficient information processing will only grow in order to address the needs of increasingly larger and complex ANNs, the exponential growth in the power of electronic devices that we enjoyed in the last half century appears to be coming to a halt.

What is more, the von Neumann architecture that is currently employed by electronic devices is known to be highly inefficient for most ML applications. In a von Neumann computer, the memory and processing units are separated, and the necessary transfer of data between them can severely constrain the overall performance. The field of neuromorphic computing [2,3] aims to improve the efficiency of specialized hardware for ML by imitating the structure of biological neural networks. The hope is to realize devices that are as efficient and massively parallel as the brain, while using much faster physical processes to carry out the information processing.

In particular, the idea of constructing neuromorphic computing devices based on light has recently attracted a lot of attention, as it promises to unlock all the benefits of optical computing: broad bandwidth, small latency, low power consumption, and natural parallelism [4]. Apart from optics, other physical platforms have been considered too, such as spin-based devices or memristor circuits [2].

Nonetheless, *training* such machines efficiently is still a challenging problem. The simplest method, shifting each learning parameter individually and providing feedback based on the outcomes, is very inefficient, scaling linearly in the number of parameters. Recently, a method using a combination of a physical forward pass but digitally simulated backpropagation was used successfully, but it requires relatively faithful models of the physical dynamics [5]. In another recent line of development, the concept of equilibrium propagation was introduced [6], leading to a local update rule for adapting weights based on a system's response under externally forced conditions, and it was recently pointed out how to implement such ideas in electronic systems by designing suitable components [7].

The best solution would be to realize a self-learning machine—i.e., a learning machine that is trained by means of entirely autonomous physical processes (Fig. 1), without the use of feedback for updating the learning parameters

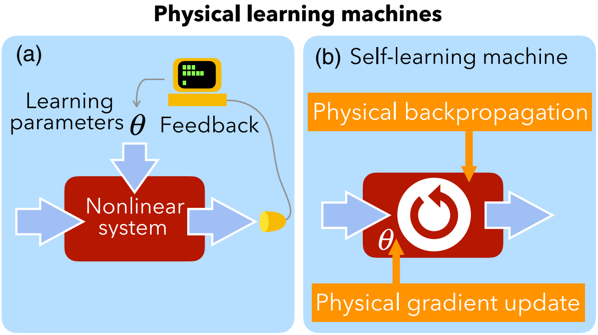

FIG. 1. Different types of physical learning machines. (a) Physical learning machine requiring feedback based on the outcome, to tune the internal "learning" parameters θ inside the physical device. (b) Self-learning machine that does not involve feedback. It updates the learning parameters in a fully autonomous way (for example, using physical backpropagation and a physical gradient update; see main text).

and without any kind of external processing of information (except possibly that needed for feeding the training data). A first step in this direction was already suggested theoretically in a visionary paper by Psaltis *et al.* [8] and later implemented to some degree [9–11]. In Psaltis *et al.* [8], it was shown that it is possible to approximately realize the backpropagation algorithm [12] in an optical neural network based on volume holograms. However, the nonlinear elements must be engineered so that the transmittance for the backpropagating signal matches the derivative of the transmittance for the forward propagating signal, in addition to requiring a carefully designed geometric arrangement [13]. To this day, such stringent requirements have prevented a fully developed practical implementation of the backpropagation algorithm in an optical learning machine [14].

Other learning machines based on Hebbian learning or spiking timing-dependent synaptic plasticity (STDSP) would also fall under our definition of a self-learning machine, and restricted versions have been implemented, e.g., optically [17]. Still, both Hebbian learning and STDSP are motivated mainly by their plausibility in the context of biological neural networks, and in each case, we see how well they optimize some training objective. In soft-matter systems, ideas based on equilibrium propagation have recently been explored [18].

In this work, we consider SL machines based on classical Hamiltonian systems. We present a new training procedure based on time-reversal operations, which we call Hamiltonian echo backpropagation (HEB). As opposed to existing approaches, we do not need to consider a particular implementation of a SL machine or carefully selected nonlinear elements; instead, we present a completely general procedure to train SL machines in a wide class of physical systems. Indeed, HEB can realize gradient descent and update the learning parameters via the dynamics in *any* time-reversible Hamiltonian system, and it does so in an efficient way that exploits the parallelism of the learning machine. This new training procedure opens up many exciting possibilities, making it possible to construct self-learning machines in a wide range of physical platforms. On the one hand, HEB makes it feasible to realize SL devices based on already mature technologies, such as integrated photonic circuits. On the other hand, it expands the field of SL machines to new, interesting, physical platforms, such as clouds of cold atoms or trains of optical pulses in a fiber loop. Given how broad the sufficient conditions are for HEB to work, we believe that it will instigate the discovery of completely new physical learning machines. Moreover, our training procedure is independent of the Hamiltonian that describes the dynamics of the learning machine. Surprisingly, one does not even need to know the dynamics of the physical device in order to train it.

In the following, after defining self-learning machines, we will introduce the Hamiltonian echo backpropagation procedure and present a mathematical proof of its core element. We will then discuss the ingredients of such a machine, illustrate the learning approach in two numerical examples, and finally comment on the prospects for different physical platforms.

II. SELF-LEARNING MACHINES

A. Definitions: Physical learning machines vs. physical self-learning machines

A physical learning machine can be defined as a physical device provided with an internal memory that can process an input to produce an output, such that the functional dependence between the input and the output is parametrized by the state of the internal memory. For example, a physical learning machine can be a photonic circuit made of beam splitters, nonlinear elements, and variable phase shifters. In this example, for any input signal that enters the circuit, the output will depend on the configuration of the phase shifters, which, in this case, represent the internal memory. In a physical learning machine, training is the process of finding the optimal state of the internal memory to realize some desired input-output relation.

There are several possible ways of training a physical learning machine. It could be done entirely externally (using numerical simulations) or by employing feedback, i.e., external processing of the machine's output to adapt the parameters. However, this step may spoil some of the advantages of using physical dynamics. Going beyond that, the most advanced version would consist in a machine that uses an internal physical process for training.

We may thus define a self-learning machine as a physical learning machine that, when presented with a data set, can train itself in a fully autonomous way. During training, a self-learning machine receives a sequence of inputs and some information about the target outputs. It is also legitimate to realize a preset sequence of external operations on the SL machine or to supply energy. What is not permissible is to give any kind of feedback dependent on the internal state of the device. In this sense, a SL machine is a black box: The user provides an input and obtains an output, but requires neither knowledge of nor access to the internal degrees of freedom.

In principle, physical (self-)learning machines could be used for any machine learning task. It is customary to classify machine learning tasks into three broad categories: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the task is to infer a function that maps an input to an output based on labeled training data, which consist of example pairs of an input object and an output (usually called the label). In unsupervised learning, the goal is to find patterns or model the probability distribution of the training data set, which, unlike the case of supervised learning, is not provided with labels. Finally, reinforcement learning comprises all tasks in which an intelligent agent has to take action in an environment in order to maximize some reward. In this paper, we concentrate on supervised learning, although there may also be exciting venues in the application of physical learning machines to unsupervised or reinforcement learning. Therefore, the self-learning machines that we consider throughout this paper are those that can learn autonomously when they are provided with pairs of input-output data.

To avoid confusion, here "self-learning" refers to an autonomous physical training procedure, as opposed to the computer science terms "unsupervised," "self-supervised," or "reinforcement" learning, which refer to the type of training task.

B. Different types of physical learning

The idea of constructing specialized physical hardware for machine learning, i.e., neuromorphic devices, has been explored in various physics-based platforms [2,3]. What we define as physical learning machines have been studied theoretically, and for some cases, they have already been demonstrated experimentally in the context of photonic hardware [9,10,15,16], memristor circuits [19–21], superconducting circuits [22], and spintronic devices [23–25], among others. However, as pointed out above, learning is still a challenge. In many platforms, the focus has been on demonstrating individual components and ingredients believed to be useful for both inference and learning, or on implementing larger-scale devices, but without *in situ* training.

The vast number of potential learning rules may be subdivided in several ways (Fig. 2). For our purposes, the two most significant questions are whether any given learning approach is based directly on optimization and whether the learning rule is local. Importantly, there is an overlap between these two categories, and we are interested in that overlap region since it has the ability to obtain learning updates by physical dynamics (local rule) and, at the same time, has a rigorous mathematical foundation (being based on optimization). Another question is whether updating the learning parameters requires external feedback and processing or can be implemented by physical dynamics autonomously.

There is a broad category of neuromorphic devices that make use of learning rules inspired by biological neural networks. This corpus of ideas started historically with Hebbian learning, and it can be summarized by the motto "neurons that fire together, wire together." A more sophisticated version in the context of spiking neurons is the so-called spike-timing-dependent synaptic plasticity (STDP). This has been suggested as a learning mechanism in several neuromorphic hardware platforms, e.g., for spintronic devices (see simulations in Ref. [26]), memristors (see implementation of a building block in Ref. [27]), superconducting circuits [22], and even optoelectronic-superconducting hybrid systems [28]. Importantly, these learning rules are always local in order to be biologically plausible (i.e., the update in the learning parameters is a function of only the nearest-neighbor neurons). However, the simplest versions of the Hebbian learning rule are, in general, not guaranteed to minimize some cost function, and they show a poor performance when applied to deep learning architectures.

Therefore, it is remarkable that there have been recent discussions about biologically inspired learning algorithms that were shown to approximately or exactly minimize a cost function that measures the deviation from the desired behavior. Thus, such approaches would fall in the particularly interesting region of overlap we mentioned above. Sometimes, these are more abstract conceptual ideas (like the free-energy principle of the brain [29]), while other approaches may be more concrete, relating to particular situations like recurrent networks of a certain structure [30]. There are interesting models that combine local rules with top-down signals in biologically plausible models, whose learning dynamics approximates backpropagation under certain conditions [31]. The area of backpropagation-like mechanisms that may be operating in the brain has been reviewed recently in Ref. [32]. In general, though, all these approaches, while very interesting, are not straightforward to implement physically in simple neuromorphic devices, and their hardware implementation would presumably require careful design and, often, tailored electronic components.

Optimization of a cost function is of course at the core of deep learning in artificial neural networks. There, the required gradients of the cost function with respect to the learning parameters inside the network can be efficiently calculated using the backpropagation algorithm. Unfortunately, this algorithm is nonlocal. The update of any weight has a complicated nonlocal dependence on any other neuron in the network. It is therefore not *a priori* obvious how to implement backpropagation physically.

In the domain of machine learning, the concept of so-called contrastive learning is at the basis of local update rules that could replace backpropagation. In general, for contrastive learning, the learning update is the difference between two contributions obtained in different learning phases. This was first employed to suggest an alternative to backpropagation in so-called "learning by recirculation" [33] and then more recently generalized and extended, eventually leading to the idea of equilibrium propagation [6]. In equilibrium propagation, one considers a system that relaxes into some steady state, given external forces that represent inputs and outputs for a given training sample. Two learning phases are distinguished, where the outputs are free or are nudged towards their desired values. The difference between the network activations in these two phases defines the weight update, in a local way. This procedure overcomes many problems of earlier similar ideas, and it implements backpropagation for the optimization of a cost function. While it was first suggested as a principled, local alternative to backpropagation, without immediate ideas about hardware realizations, there has been recent work on possible implementations of this (and related) ideas, both in electronic circuits [7] and in soft-matter systems [18]. In contrast to the method we describe, the basic assumption here is always that the system relaxes to an equilibrium steady state (or begins to relax towards that state), for each presented training example.

C. Problem of training

In some of the earlier attempts (like Ref. [11]), training was performed *in silico,* by means of numerical simulations. The problem with this approach is that the size of the learning machine is limited by the computational power required to simulate the physical hardware. Moreover, training is performed for an idealized model of the actual physical hardware. As a result, fabrication imperfections cannot be taken into account. This has been addressed recently by introducing a hybrid approach, combining physical forward propagation on actual hardware with backpropagation using simulation [5].

A general approach to remedy this shortcoming is to employ feedback. The simplest feedback-based training technique uses a finite-difference approximation to the gradient of the cost function. This can be realized physically in a very simple manner: (1) estimate the cost function C for the current state of the learning parameters, (2) change the ith parameter by a small amount and estimate C again, and (3) add the difference of these results to the ith parameter. However, this idea is far from optimal since the number of evaluations needed scales with the number of parameters.

D. Towards physical self-learning machines

As described above, ideally we would obtain the required training update of the learning parameters and, eventually, also implement that update using only autonomous physical dynamics.

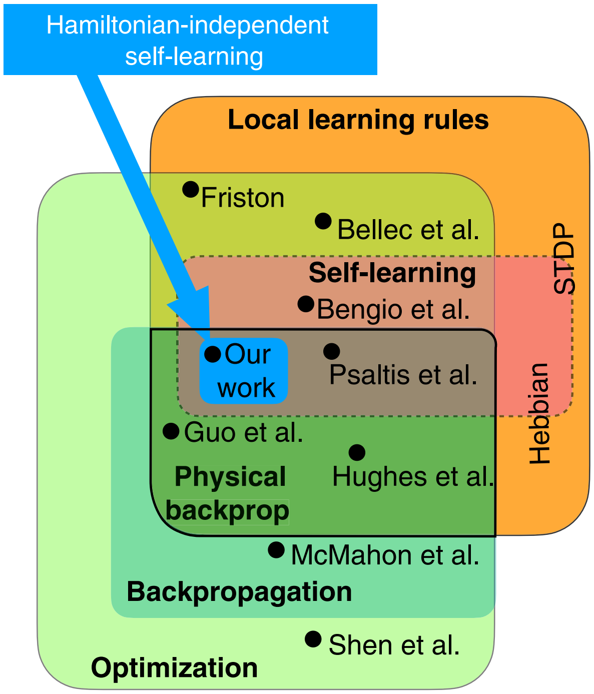

FIG. 2. Overview: landscape of physical learning machines. Our work falls into the broad class of optimization approaches. It belongs to the small class of works that involve both physical backpropagation and self-learning, but in contrast to other approaches, our scheme is universal, i.e., independent of the specific Hamiltonian (for some related works, see main text in Shen *et al.* [15], Guo *et al.* [16], Skinner *et al.* [11], Psaltis *et al.* [8,10]).

In this subsection, we review the previous developments along this line that we feel are most closely related to our idea. They happen to be in the optical domain—not surprisingly, given the coherent, time-reversible character of wave propagation.

We know that the backpropagation algorithm that forms the cornerstone of artificial neural network training can extract the required parameter update (the gradient of the cost function) using only two evaluations of the network, independent of the number of parameters. A crucial improvement of a physical learning machine therefore consists in realizing the backpropagation algorithm by physical means, even in cases where the eventual update is implemented by other means. This idea was first introduced in Wagner *et al.* [8], in the context of optical neural networks. In a typical optical neural network, an optical field propagates inside some nonlinear medium with elements that provide controllable phase shifts. The controllable phase shifts play the role of the learning weights. The optical field would be injected at the input and propagate towards the output. The idea advocated in Ref. [8] is then to create a physical error signal that propagates in the opposite direction, from output to input. The error signal is prepared according to the difference of the output of the device and the desired target output. Without going into any further detail (we refer to the original paper), in their approach, the backward-propagating weak error signal interferes with the strong forward-propagating optical field. In this way, for the specific choice of optical nonlinearities in their setup, it was possible to ensure that the error signal is (approximately) equal, at any point of the device, to the required gradient of the cost function.

Once the error signal is prepared, one can measure it and use the result to update the parameters via feedback. All learning parameters could be updated at once, in a fully parallel fashion. This idea was originally proposed for optical neural networks using Kerr nonlinearities [8], but it was recently extended to setups that employ saturable absorption for the activation function [16]. In a similar spirit, in Hermans *et al.* [34], the idea of physical backpropagation is used to train a physical linear system with controllable nonlinear feedback. An optical implementation based on this concept was demonstrated in a follow-up paper [35]. A related idea by Hughes *et al.* [36] uses an auxiliary circuit to backpropagate the error signal. A recent paper proposed a training method that combines *in situ* measurements with digital backpropagation [5]. Other implementations only train the last layer, in a similar fashion to reservoir computing.

Nonetheless, there is a potential drawback in using feedback to update the learning parameters. The error signal has to be measured in order to update the learning weights. If that is done by means of electronic sensors, it could potentially introduce a bottleneck. The potential advantage of using ultrafast physical dynamics for evaluating the network could be lost because of a relatively slow electronic response. In addition, some of these approaches require further computation steps. These problems can be addressed if the error signal directly influences the learning parameters via physical dynamics, without external feedback.

The possibility of using a physical mechanism to update the learning parameters autonomously was proposed for the first time in the above-mentioned work by Wagner and Psaltis [8]. In their paper, they consider a setup in which the learning parameters are recorded in the form of a hologram in a photorefractive material. In this way, the interference between the error signal and the forward-propagating beam could, in principle, provide a means to update the holographic recording in the right way. A simple version was partially demonstrated in a linear optical device in a follow-up paper by Li *et al.* [9]. However, that experimental setup actually still involved some form of feedback.

The ideas first advanced in Ref. [8] could potentially lead to ultrafast training and a high density of interconnections between neurons. By combining physical backpropagation with an autonomous process to update the parameters according to the error signal, it would be possible to realize self-learning machines. Nevertheless, all presently existing proposals for physical backpropagation are suitable only for some setups with very specific nonlinearities. In some cases, such as in Refs. [8,16], it is required that the forward propagating beams and the error signal experience a different dynamics at the nonlinear elements, which nonetheless has to be matched in a precise way. What is more, recording the learning parameters in a holographic pattern requires a very careful engineering of the geometry of the device [13]. In other cases, such as in Ref. [11], physical backpropagation is only realized approximately in a particular limit.

In this work, we show how to realize self-learning (i.e., the combination of physical backpropagation and the autonomous parameter update) in a much broader set of physical systems. Since our idea does not rely on a specific choice of system, it is not restricted to optical setups. Hence, it could be applied to many other possible platforms, ranging from cold atom clouds to solid-state spin networks. In fact, the training procedure that we propose is a sequence of operations that do not depend on the particular dynamics of the device, provided that some weak assumptions are met. Essentially, we only require that the physical learning machine is based on a time-reversible, nonlinear Hamiltonian system.

III. OVERVIEW OF HAMILTONIAN ECHO BACKPROPAGATION

The main contribution of this paper is to introduce a new technique for self-learning in Hamiltonian systems, which we term Hamiltonian echo backpropagation (HEB, for short). Before moving to the more technical sections, here we provide a high-level overview of HEB.

A. Requirements

Let us be more precise about the requirements for the implementation of a self-learning machine based on HEB.

Hamiltonian systems.—First, we only consider Hamiltonian systems where the dynamical degrees of freedom consist in the learning parameters and the variables needed to process the information. From the point of view of the physical realization, requiring Hamiltonian dynamics means that the timescale of dissipation in the self-learning device must be much larger than the time interval between the injection of the input and the production of the output (for a given training sample). Since we aim to use fast dynamical processes, this seems to be a reasonable requirement. In fact, many of the previous proposals for physical learning machines can be modeled as Hamiltonian systems, e.g., the optical neural networks using Kerr nonlinearity.

Time-reversal symmetry.—Second, we require that the self-learning device obeys time-reversal symmetry. It is worth noting that time-reversal symmetry is not a very restrictive property: All mechanical and optical systems with arbitrary nonlinear dynamics are time-reversal invariant in the absence of magnetic fields and in the absence of spontaneous time-reversal symmetry breaking (i.e., magnetization).

For the purposes of our paper, we define a Hamiltonian self-learning machine to be a self-learning machine that is Hamiltonian and time reversible.

Time-reversal operation.—Third, we need the ability to physically implement a time-reversal operation. In the case of wave fields, this would be a phase-conjugation mirror, an operation that has been demonstrated in nonlinear optics. This can be implemented outside the core of the device.

Decay step.—Fourth, we need the ability to implement another operation, the so-called decay step, to be explained below. Essentially, this is when the force imparted on the learning variables is converted into a shift of those variables. Again, this can be implemented outside the core of the device.

These are the only general requirements needed for the implementation of self-learning machines based on the Hamiltonian echo backpropagation technique. More detailed suggestions regarding physical implementations made in the remainder of this paper only serve to highlight opportunities for optimizing the performance—similar to discussions in deep learning where one should distinguish the essential concepts, i.e., gradient descent and backpropagation, from suggestions for improved network architectures.

B. Basic steps of Hamiltonian echo backpropagation

We explain here all the steps needed to implement HEB in what is probably the most practical scenario, which is when wave fields are transmitted through a nonlinear setup. In this setup, wave fields represent both learning variables, to be updated during training, and evaluation variables, participating in the information processing required to map input to output. Both types of wave fields enter a physical system (the nonlinear core), in which they interact to produce an output wave field. The only requirement that we impose on the nonlinear core is that the evolution of the fields inside it is described by a time-reversible Hamiltonian.

Apart from the nonlinear core, the self-learning device requires a few well-defined external operations that are applied on the wave fields that have been transmitted through the device. However, we need no control on the nonlinear core, which can be thought of as a black box. Since all the external operations depend only on the output fields and not on the internal degrees of freedom or the details of the dynamics inside the nonlinear core, we call this a Hamiltonian-independent self-learning machine.

Suppose that we consider the usual supervised learning scenario, with a data set consisting of pairs of inputs and target outputs, and suppose that we want to train the device to approximate the input-output relation given by these samples. We proceed to enumerate the steps in a single training iteration of HEB.

(1) The first step is to draw a random sample from the input data set and prepare the evaluation field accordingly.

(2) *Forward pass*: Send both the evaluation field and the learning field into the nonlinear core, and let them interact there. This is where the essence of the nonlinear information processing (and, later, learning process) will take place.

(3) Both fields have again emerged from the system, propagating outwards as wave fields. We now inject a small *error signal* on top of the evaluation field. This signal is proportional to the desired output of this training sample. Note that this prescription covers the case of the most well-known cost function, the mean square loss, and it mathematically corresponds to injecting a signal proportional to the gradient of the cost function with respect to the output field. Other cost functions can be implemented as well, as we will show.

(4) *Time-reversal operation*: This operation is performed by phase conjugating both wave fields, which, at this point, are outside the nonlinear core. More generally, this means reversing the momenta of the physical degrees of freedom.

(5) *Backward pass*: As a result of the time-reversal operation, the fields will propagate back into the nonlinear core, where they will interact again (we call this the backward pass).

(6) Now, the evaluation field leaves the device near the input port (it will be substituted by the next input). However, the learning field is retained, and a *second time-reversal operation* is now enacted on this learning field, again outside the device. This will eventually re-inject the learning field into the device, together with the next input sample.

(7) *Decay step*: In a Hamiltonian setting, the degrees of freedom of any field can be decomposed into the field amplitude and its canonically conjugate momentum; for the optical case, these correspond to the two field quadratures. As we prove here, the canonical momentum of the learning field after the steps described above is proportional to the gradient of the cost function, i.e., exactly what is needed in a gradient-descent learning algorithm. The purpose of this final step is to convert the canonical momentum (i.e., the field quadrature whose sign we have flipped during the time-reversal operation) into a change of the field amplitude and quench the momentum to zero in the process. We call this the decay step since it is the only step in which dissipation is introduced. As this step takes place outside the nonlinear physical system (just like the time-reversal operation

itself), it can be realized in an engineered way without destroying the Hamiltonian independence advertised above. Furthermore, since it is realized outside the nonlinear core, one can control the learning rate in a simple manner.

We discuss all of these steps in the following and provide proofs for the claims made here. Nevertheless, we can already offer insight into why the full procedure works. Consider that we want to evaluate the gradient by the finite-difference method. In this case, we have to slightly shift one of the learning fields and query the resulting response. Such a shift would result in a weak perturbation of the fields during the forward pass that would ultimately result in a small change of the value of the cost function with respect to the case of unperturbed fields. Now, we have to repeat this process for every learning degree of freedom to obtain the full learning gradient, which is one of the reasons why, as we explained above, the finite-difference method is hugely inefficient. However, since we consider systems with time-reversal symmetry, we can instead inject an error signal into the output, followed by a time-reversal operation. In this case, the wave fields evolve to a time-reversed replica of the input that is perturbed by a weak error signal. Because of the reciprocity principle [37], the linear response of the (nonlinear) system in the forward pass is in fact related to the linear response of the system in the backward pass. Therefore, we can obtain the required gradient by injecting a signal from the output, instead of perturbing the input. This is the essence of the idea underlying our approach.

Like the approach of equilibrium propagation [6], the general idea of our method is to exploit the difference between two contributions: In our case, this would be the difference of the force imposed on the learning field between the forward pass and the backward pass. The difference is automatically produced by the physical dynamics since we employ a time-reversal operation. In contrast to equilibrium propagation, we do not have to further engineer the dynamics inside the device to produce an update according to such a difference. Another distinction is that we work with a system out of equilibrium so we do not have to wait for it to relax to any steady state (nor would we want to since our method relies on dissipation being absent or at least very small).

Like the approach of Psaltis and Wagner [8], we employ a time-reversal operation. However, in our case, we apply a time-reversal operation to the full nonlinear wave field, as opposed to only an injected weak error signal. This difference ultimately means that the forward and backward transmissions had to be engineered inside the device components in a particular way, while our approach is completely general.

At this point, we observe that none of the external procedures that are employed during a HEB iteration depends on the Hamiltonian of the nonlinear core. Therefore, we introduce the following definition. A Hamiltonian-independent self-learning machine is a Hamiltonian self-learning machine in which the training procedure is independent of the Hamiltonian. By this statement, we mean that the training procedure can minimize the cost function for any Hamiltonian that is time reversible. This is another important difference from equilibrium propagation as well as the approach of Wagner and Psaltis.

To the best of our knowledge, our proposal constitutes the first example of an approach to construct Hamiltonian-independent self-learning machines. Since the training procedure does not depend on the particular form of the Hamiltonian, the self-learning machine can be regarded as a black box. As opposed to other physical learning machines in the literature, its implementation in the laboratory does not require detailed knowledge of the internal dynamics of the device in order to make it work. There are only a few mechanisms that could disrupt an experimental implementation: a mechanism that in some way breaks time-reversal symmetry of the Hamiltonian (including an unwanted dissipation or noise channel) or significant imperfections in the implementation of the time-reversal operation or the decay step.

IV. HAMILTONIAN ECHO BACKPROPAGATION

A. Dynamical variables and Hamiltonian

Our setup for a SL machine consists of two basic ingredients: dynamical variables that are used to process the information (the evaluation variables) and dynamical variables that contain the learning parameters.

The evaluation variables will be combined into a single dynamical vector $\Psi(t)$. In a similar way, the collection of all the learning parameters is denoted as a vector $\Theta(t)$. Such vectors can represent any collection of physical degrees of freedom in a Hamiltonian system, from interacting spins to optical fields. For the present discussion, we imagine discrete models with a countable number of degrees of freedom. However, the approach also covers the case of continuous fields (with straightforward slight changes in the notation).

The input to the device is given by the initial configuration of the evaluation variables, $\Psi(-T)$, whereas the output is given by its state at a later time, $\Psi(0)$. We choose the initial time to be $-T$ for reasons that will become clear later. The learning parameters $\Theta(t)$ will interact with the evaluation variables and, overall, change slowly during training.

For example, consider a self-learning machine where Ψ and Θ are nonlinear wave fields. These could be realized as a lattice of optical cavities or spins. We can imagine the operation of the SL machine as a nonlinear scattering experiment, where the input $\Psi(-T)$ is a wave packet. As the wave packet propagates inside the device, it evolves under the effect of the nonlinear self-interaction and the interaction with the learning parameters. Finally, an output wave packet is obtained at $t = 0$. The output state of Ψ is a nonlinear function of the input, and it has a parametric dependence on the initial configuration of the learning parameters, $\Theta(-T)$, much like an artificial neural network (ANN).

We can capture the dynamics of the SL machine during both the forward and backward pass in a single Hamiltonian:

$$H_{\mathrm{SL}} = H_{\psi} + H_{\mathrm{int}} + H_{\theta}. \tag{1}$$

Let us inspect, one by one, what each term means. Some of the technical details will be more fully explained later.

Here, H_{ψ} is the Hamiltonian that describes the evolution of Ψ alone, in the absence of interaction with Θ. In general, it will have self-interaction (nonquadratic) terms. The second term, H_{int}, contains the interaction of Θ and Ψ. In general, $H_{\psi} + H_{\mathrm{int}}$ must be a time-reversible Hamiltonian, but there is no further constraint. In accordance with previous proposals for physical learning machines, we may assume that, for example, Ψ is an optical field inside a nonlinear medium. The nonlinear response of such a medium may correspond, among many other possible choices, to a Kerr-like nonlinearity.

The third term, H_{θ}, describes the dynamics of Θ in the absence of interaction with Ψ. For the sake of generality, we consider that Θ may have an arbitrary self-interaction. Therefore, H_{SL} is completely arbitrary.

B. Gradient descent as a goal

Suppose that Θ is initialized in some random configuration $\Theta(-T)$. Now, we prepare Ψ in the input state, and we let it interact with Θ. After a time interval T, we obtain the output state. The error in the output is quantified by means of a cost function. In general, a cost function can be any function,

$$C(\Psi(0), \Psi_{\mathrm{target}}) = C(\Psi(-T), \Psi_{\mathrm{target}}, \Theta(-T)), \tag{2}$$

of the output $\Psi(0)$ and its corresponding target configuration that is minimized when both are equal. The function displayed here is the sample-specific cost function. The training procedure attempts to minimize its sample-averaged version [averaging over all training input samples $\Psi(-T)$ that are provided alongside their respective target Ψ_{target}]. After averaging, the cost function still depends on the learning variables $\Theta(-T)$ since these determine the mapping from input to output.

For reasons that will become apparent later, we take our cost function to have units of energy (and, in fact, it will represent a physical Hamiltonian later in this section). In order to improve the performance of the SL machine, we have to minimize the cost function. In principle, the problem of optimizing the cost function is a very hard nonlinear optimization problem, but in the context of ML, it is usually sufficient to find a local minimum. In the context of ANNs, this is normally performed using the stochastic gradient descent method. In this method, the gradient of the cost function is computed (for one or several randomly chosen training samples), and the learning parameters are subsequently updated, according to the rule

$$\theta(T_f) = \theta(-T) - \eta \frac{\partial C}{\partial \theta(-T)}. \tag{3}$$

C. Introducing Hamiltonian echo backpropagation

Our goal is to realize the stochastic gradient update rule in a fully autonomous way, via physical dynamics. The sample-specific cost function depends on the output and the target state. For this reason, if one hopes to implement gradient descent without external intervention, some information about the output must be sent backwards from the output to the input. This immediately suggests the use of a time-reversal operation, which was already recognized in early proposals [8], although it was employed in a manner different than what we describe now.

To understand the qualitative idea of our approach, we start from a general observation that is well known from discussions of backpropagation in ANNs. The change of the cost function induced by a small change in a learning parameter θ is given by

$$\delta C = 2\delta\theta \mathrm{Re}\left\{\frac{\partial \Psi(0)}{\partial \theta}\frac{\partial C}{\partial \Psi(0)}\right\}. \tag{4}$$

In other words, we need to know both the perturbation in the output configuration $\delta\Psi(0) = \partial\Psi(0)/\partial\theta$ produced by a small parameter change and the "error signal" $\partial C/\partial\Psi(0)$ at the output.

Since we consider weak perturbations, $\delta\Psi(0)$ can be understood in terms of the linear response of the system. In other words, $\delta\Psi(0)$ is in fact proportional to the Green's function associated with the linearized equations of motion. At this point, we can introduce the most important idea of our approach: If we physically time reverse the whole field Ψ, we can instead interpret the error signal $\partial C/\partial\Psi(0)$ as the source of a perturbation that is "riding on top of" the time-reversed nonlinear wave field (see Fig. 3). This is at the heart of our new procedure, HEB, to realize self-learning.

As we will see later, the idea is related to the fact that in a linear system of differential equations, there is a symmetry between the source of a field at t and its effect at t' (the so-called self-reciprocity principle) [37].

Let us go through the individual steps of the HEB procedure. First, Ψ is prepared in the input state, at time $t = -T$. Note that we choose to start the process at time

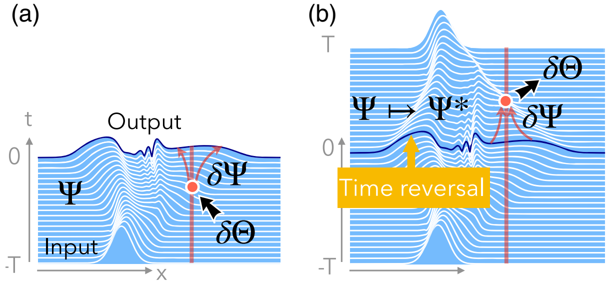

FIG. 3. Physical backpropagation and physical gradient update. (a) Forward pass (nonlinear evolution) of the "evaluation" wave field Ψ in the presence of a background potential landscape provided by a "learning field" Θ. Any small perturbation $\delta\Theta$ will affect the output, as indicated here, which in turn changes the value of the cost function. In panel (b), we show that, instead of testing the cost function response for many potential updates of the learning field Θ, Hamiltonian echo backpropagation automatically produces the required update in a single backward pass, via physical dynamics. After applying a time-reversal operation ($\Psi \mapsto \Psi^*$), the subsequent nonlinear wave-field evolution is a time-reversed version of the forward pass. A small variation $\delta\Psi$ injected on top can now be used to backpropagate the error signal (variation of the cost function at the output); see main text. This then leads to the required update of the learning field θ, via the physical interaction between Ψ and θ, realizing a fully autonomous learning machine.

$t = -T$ to make explicit the time-reversal symmetry. Thus, the output is produced at $t = 0$, and the echo of the input will form at $t = T$. In order to simplify the calculations, let us also assume first that the canonical momentum of θ is initially zero. After the interaction between Ψ and Θ, we obtain an output field $\Psi(0)$.

In the next instant, we need a way to inject the error signal required for the subsequent backward pass. We achieve this by imposing on Ψ an interaction Hamiltonian proportional to the (sample-specific) cost function, given the desired target field configuration. We assume that the duration of this interaction, ϵ, is small enough. In practice, this implies that the field is slightly changed according to $\Psi(0) \to \Psi(0) - i\epsilon\partial_{\Psi^*(0)}C$.

Before going further in the discussion of HEB, let us introduce a useful notation that we employ in the rest of the paper. In order to make our equations more compact, we condense each pair of canonical coordinates in a single complex variable. We reserve the capital Greek letters for the complex variables, as follows: $\Psi = \psi + i\pi_\psi$, $\Theta = \theta + i\pi_\theta$, where the canonical coordinates and momenta have been rescaled to obtain the same dimension (nominally, the square root of an action). Thus, the notation for the time-reversal is especially convenient, as it corresponds to a complex conjugation of the dynamical variables. Furthermore, one can easily check that Hamilton's equations can now be written in a particularly convenient manner:

$$\dot{\Psi} = -i\partial_{\Psi^*}H,$$

where one has to formally treat Ψ, Ψ^* as independent variables (or more technically, ∂_{Ψ^*} represents a Wirtinger partial derivative).

As a concrete example, let us assume that our cost function is simply the overlap between the target state and the output state. Then, the prescription above tells us that we just have to inject a weak perturbation proportional to the target state: $\delta\Psi(0) \propto i\epsilon\Psi_{\text{target}}$.

After this step, we time-reverse, which, in our complex notation, means phase conjugating the fields, e.g., $\Psi \mapsto \Psi^*$. In the limit $\epsilon \to 0$, Ψ will evolve backwards precisely to the input configuration because the Hamiltonian is time-reversal invariant. However, when ϵ is small but finite, we obtain a small variation in the fields, both in Ψ and Θ. One can understand such variation as a perturbation traveling on top of the time-reversed field, from output to input. Once the backpropagation is complete, one obtains a slightly perturbed echo of both Ψ and Θ, at $t = T$. Since we have perturbed the field at $t = 0$, the echo is not perfect: There is a small variation given by $\Theta(T) = \Theta^*(-T) + \delta\Theta(T)$. In order to have the correct sign for the momentum of the learning field required for our approach, we time reverse again, $\Theta(T) \mapsto \Theta^*(T)$. Using the self-reciprocity principle, one can then show that the final configuration of Θ in the echo step is simply given by

$$\Theta(T) = \theta(-T) - i\epsilon \frac{\partial C}{\partial\Theta^*(-T)}. \tag{5}$$

We show below how to prove that result, which encapsulates the central idea of our proposal, mathematically (Sec. V).

We note that the procedure introduced here differs in an important way from previous approaches to self-learning dynamics [8], which never considered a full time reversal of the nonlinear wave field and instead treated a weak perturbation traveling against a strong forward-propagating field. It is a consequence of this choice that these approaches work only for some specific, carefully chosen, wave dynamics, and not in the general, Hamiltonian-independent way that we outline here.

The final step of HEB will be to update the learning field, as we now discuss.

D. Decay step (physical learning field update)

At this step, the dynamics of the learning field Θ almost looks like gradient descent but not completely: We have been able to impart a momentum kick to Θ that is proportional to the gradient of the cost function, but what we really want to do is update the position θ. We now need to convert the update in the canonical momentum into a shift in the position, and at the same time, we need to let the momentum decay to zero. For this purpose, we need to realize some form of dissipation during this part of the training process, which we call the decay step. This can be done by coupling Θ to a reservoir.

While we need dissipation during the decay step, the dynamics during the forward and backward pass must be time reversible. The conceptually simplest way to resolve this conflict is to have the ability to switch dissipation on and off. In principle, we can do so by controlling the coupling of Θ to the reservoir, and many physical implementations (e.g., all approaches related to laser cooling ideas) are possible. The fact that we need dissipation during the decay step is not surprising because we have only used Hamiltonian evolution and time reversals so far, while the overall gradient descent procedure is contractive.

In general, therefore, during the decay step, we need to switch on a dissipative interaction

$$H_{\mathrm{SL,decay}} = H_{\mathrm{SL}} + H_{\theta,\mathrm{decay}}, \tag{6}$$

where $H_{\theta,\mathrm{decay}}$ is a term that couples θ to the dissipative bath. Additionally, $H_{\theta,\mathrm{decay}}$, in general, needs to contain other simple nondissipative terms that control the evolution during the dissipative phase.

We now explain this step by means of a simplified example that contains all the key ingredients needed. In Sec. VI C, we provide a more detailed prescription for how the desired effect outlined here can be obtained in the concrete case of wave fields.

In our example, we assume that the dynamics of Θ during the decay step corresponds to a free-particle Hamiltonian with damping:

$$\dot{\theta} = \Omega\pi_\theta,$$
$$\dot{\pi}_\theta = -\Gamma(t)\pi_\theta.$$

(Note that, in our convention, π_θ^2 has units of action, which implies that Ω has units of frequency). Furthermore, we assume that the effect of dissipation may be modeled by a damping force of the form $-\Gamma(t)\pi_\theta$, where $\Gamma(t)$ can be controlled in a time-dependent fashion. The stable manifold would be given by the particle at rest at any location, with $\pi_\theta = 0$.

During this final step of the HEB procedure, we switch on the dissipation, $\dot{\pi}_\theta = -\Gamma\pi_\theta$, and apart from that we let Θ evolve freely. If we wait long enough, the final configuration of Θ is given by $\theta(-T)+\epsilon[\partial C/\partial\pi_\theta(-T)]-\epsilon(\Omega/\Gamma)[\partial C/\partial\theta(-T)]$ (to first order in ϵ), while the canonical momentum decays to zero. The update step finishes at the end of the decay step, at time $T_f \gg \Gamma^{-1}$. In the limit of large decay times, i.e., $(\Omega/\Gamma) \gg [\partial C/\partial\pi_\theta(-T)][\partial C/\partial\theta(-T)]^{-1}$, we recover the usual gradient descent update rule

$$\theta(T_f) = \theta(-T) - \eta\frac{\partial C}{\partial\theta(-T)}, \tag{7}$$

with a learning rate given by $\eta = \epsilon(\Omega/\Gamma)$. Repeating this procedure in a sequential way, for many injected training samples, will realize stochastic gradient descent (SGD). Apparently, the learning rate is proportional to ϵ, which represents the duration during which the "cost function Hamiltonian" is active.

We must remark that in the general case, when the dynamics of Θ is not necessarily equivalent to a free particle with damping, the relation of η to ϵ may be different. However, we can always expect these two quantities to be proportional due to the assumption of injecting only a weak perturbation.

In practice, the way in which the decay step is performed would depend on the architecture of the self-learning machine. We can envisage two main possibilities, which are illustrated in Fig. 4.

Before we discuss these possibilities, we emphasize that it is important to have an overall dynamics where the degeneracy between different values of the learning field is only broken by the training samples and not by the intrinsic dynamics of the device. After all, the learning parameters Θ must be able to change continuously between different states during training.

Learning field transmitted through the nonlinear core.— The first possibility, illustrated in Fig. 4(a), is that both Θ

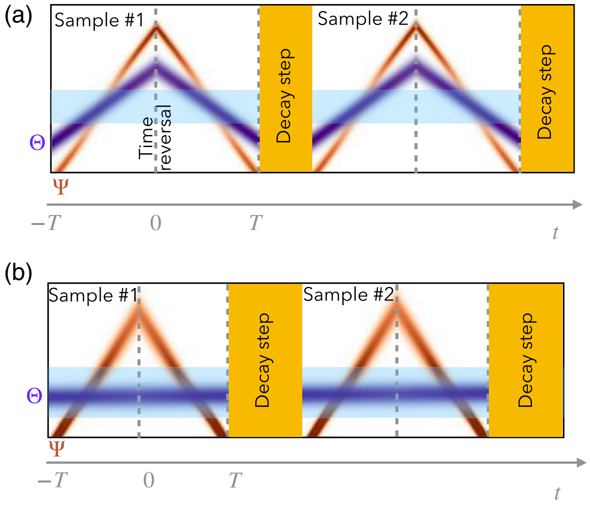

FIG. 4. Different possible architectures of the self-learning machine. We illustrate the different possible configurations of the learning and evaluation dynamical variables, for the case of a setup consisting of wave fields interacting in 1D. We consider that the SL device is in the highlighted region, which is where the nonlinear interaction given by H_{SL} takes place. If both the input and output of both fields are injected from outside, all the time reversals and decay steps can be performed externally, and there is freedom to choose $H_{\theta,\mathrm{bath}}$. If not, the Hamiltonian of the SL device, H_{SL}, has to be engineered to have a continuously degenerate ground state in Θ. In panel (a), both $\Psi(-T)$ and $\Theta(-T)$ are wave packets injected from outside the SL device. In panel (b), the input $\Psi(-T)$ is a wave packet injected from outside the device, but Θ is confined inside.

and Ψ are wave fields that enter in the self-learning device from outside, evolve according to $H_{\rm SL}$, and then finally produce an output wave field. In this case, the echo of both Θ and Ψ is produced outside the device. Therefore, all the operations needed to realize the decay step can be performed externally, and there is freedom to choose the most suitable form of $H_{\theta,{\rm decay}}$. In this situation, the procedure is completely Hamiltonian independent since the operations required to implement $H_{\theta,{\rm decay}}$ are the same for any time-reversible Hamiltonian self-learning device. In this case, even when the training is finished, one must keep time-reversing Θ and performing decay steps in order to keep it around the optimal configuration. In the absence of an error signal injected at the output, the time-reversed evolution during the backward pass will exactly undo any evolution during the forward pass, regardless of the Hamiltonian of the nonlinear core. Therefore, the learning field will remain stable once the training is finished. The only conditions to impose are on the Hamiltonian governing the dissipative dynamics outside the device, $H_{\theta,{\rm decay}}$, which must not break the degeneracy between different values of θ (unlike the Hamiltonian $H_{\rm SL}$, which can be arbitrary in this case).

For this architecture, since the decay step is performed outside the device, in principle, there is no constraint on the values of the learning rate or the particular form of dissipation. In this case, the learning rate can be tuned to maximize the speed of training. Just like in the case of training of artificial neural networks, the learning rate cannot be too high; otherwise, a gradient descent step is not guaranteed to decrease the value of the cost function.

Learning field localized inside the nonlinear core.—The second possibility, illustrated in Fig. 4(b), is that the evolution of Θ happens entirely inside the device. For example, this would be the case if the learning dynamical variables are given by optical fields confined in cavities placed inside the self-learning device or are stored inside some other persistent degrees of freedom. In that case, there is the danger that during the decay step, the dynamics prescribed by $H_{\rm SL}$ would induce some unwanted drift of the learning field. To avoid this, we need to require that the dynamics induced by $H_{\rm SL}$ results in a continuously degenerate ground-state manifold for Θ. Therefore, when the learning dynamical variables are confined inside the device, the form of H_θ is constrained. In Appendix G, we outline some possibilities to engineer such a Hamiltonian with a few simple nonlinear interactions. Nevertheless, we can still have arbitrary Hamiltonians for Ψ and for the interaction between Ψ and Θ, as long as $\Psi = 0$ inside the device during the decay step. This will be true if we think of Ψ as some sort of excitation (wave packet) having left the device by this time. Although this restricts the form of $H_{\rm SL}$, it has the advantage that it is no longer necessary to keep performing time reversals once the training is finished and as long as no evaluation passes are performed.

Overall, what our training procedure (in either of the two cases) does is impart an additional effective force on Θ proportional to the gradient of the cost function, as we explain in the next section. This breaks the degeneracy in such a way that the only surviving stable fixed points are the minima of the cost function.

Up to now, we have always considered that the evaluation field Ψ adopts the shape of wave packets traveling through the device. However, even there, other possibilities exist: One could adopt a scenario where the time reversal of Ψ is implemented entirely inside the device. In that case, we do not need to think of propagation along a particular direction nor an architecture reminiscent of the structure of layered neural networks. Rather, the dynamics of Ψ could resemble that of a wave-chaotic system inside a complex nonlinear cavity (see example in Appendix D).

E. Summarizing the general scheme of Hamiltonian echo backpropagation

We are now in a position to summarize the general approach (Fig. 5). The HEB procedure consists of a sequence of multiple events [Fig. 5(a)], but they can be subdivided into the following steps:

(1) *Echo step*. We randomly draw an input sample from the training data set, and we use it to inject $\Psi(-T)$. Following the nonlinear dynamics of the interacting system, the output is obtained at $t = 0$. At that point, the evaluation field is weakly perturbed according to $\Psi(0) \rightarrow \Psi(0) - i\epsilon\partial_{\Psi^*(0)}C$, which can be brought about via the dynamics induced by an extra interaction Hamiltonian (the cost function Hamiltonian), as explained above. Immediately afterwards, both Ψ and the learning field Θ are time reversed (i.e., phase conjugated). At $t = T$, a perturbed echo of the initial configuration is obtained. We then time reverse Θ again at $t = T$. The final result of this whole process is equivalent to the following update of the learning field:

$$\Theta(T) = \Theta(-T) - i\epsilon\partial_{\Theta^*(-T)}C. \tag{8}$$

To be clear, here C is always the sample-specific cost function. We show in detail how to prove this result, in general, in the next section. It has a simple interpretation: To first order in ϵ, the evolution of Θ satisfies Hamilton's equations, with an effective Hamiltonian equal to the cost function.

(2) *Decay step*. The dissipation is switched on for a time interval T_D. The evolution of Θ is determined by $H_{\rm SL,decay}$. This will ensure the suitable update of the Θ field needed to implement SGD via the physical procedure explained above. Finally, we proceed with the next training sample.

The evolution of the learning field during the whole procedure is schematically depicted in Fig. 5(c). Imagine

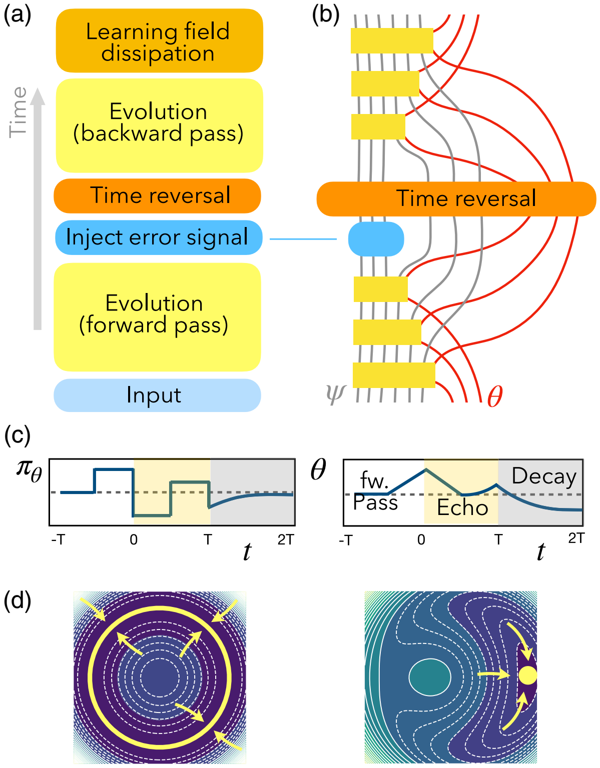

FIG. 5. Ingredients of a self-learning machine based on Hamiltonian echo backpropagation. (a) Sequence of events. (b) Use of “pseudodissipation via ancillary modes.” Here, the learning field Ψ is decomposed into a number of discrete modes. As time progresses, less and less modes participate in the physical interaction (yellow boxes), becoming ancillary modes. Both these ancillary modes and the learning field θ also take part in time reversal, however. (c) Time evolution of the learning field momentum (left) and the field itself (right) during one forward-backward pass, including the effects of time reversal (we show a single value of the field at one location). The momentum decays back to zero, but the field has been updated suitably. (d) Intrinsic dynamics of the learning field during the decay step, showing a continuously degenerate ground state (i.e., a manifold attractor). In the right panel, the effect of the training dynamics is to break this degeneracy and produce a minimum determined by the cost function.

that these two steps are alternated many times, with suitable small values for ϵ and T_D. Then, the average dynamics of Θ is approximately described by an effective Hamiltonian given by

$$H_{\theta,\mathrm{eff}} = \frac{1}{\epsilon + T_D}(\epsilon C + T_D H'_\theta), \tag{9}$$

where C here represents the sample-averaged cost function, and H'_θ depends on the scenario we are considering. For case I (time reversal of Θ outside the device), we simply have $H'_\theta = H_{\theta,\mathrm{decay}}$. For case II (time reversal inside), we have $H'_\theta = H_\theta + H_{\theta,\mathrm{decay}}$. Remember that, by our assumptions, in both cases, the phase space associated with H'_θ has a continuous degenerate stable manifold. Consider what happens in the low-temperature limit of the thermal dissipative reservoir that is used to act on Θ. When we set $\epsilon > 0$, we break the degeneracy of H'_θ [see Fig. 5(d)]. The dynamical process we described tends to equilibrate the temperatures of Θ and the reservoir, so Θ tends to minimize the cost function, constrained to the manifold of ground states of H'_θ. Once the training has converged, we can stop performing training steps (in terms of the effective Hamiltonian, this is equivalent to setting $\epsilon = 0$). Since we have converged to one of the many stable states, Θ will remain near the learned configuration for arbitrarily long times. Of course, this is important to ensure that, after training is completed, the SL machine does not quickly forget the result of the learning process. This line of reasoning does not depend on the particularities of H'_θ, as long as the phase space possesses some stable manifold.

This concludes our overview. The steps of the approach are illustrated in Fig. 6 in terms of a very simple mechanical Hamiltonian system, highlighting the generality of the procedure.

We now present a few additional considerations.

Consider what happens when the set of stable ground states of H'_θ is, in fact, not continuous but actually discrete. This may be the case if, for example, Θ describes a collection of degenerate parametric oscillators. In this situation, a gradient descent strategy would not succeed, as Θ cannot change continuously between fixed points.

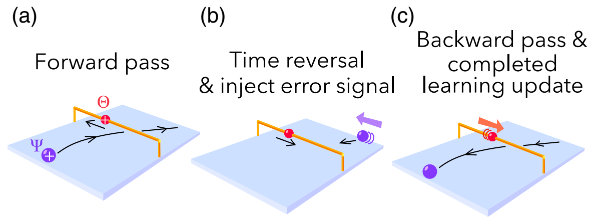
(a)
Forward pass
(b)
Time reversal & inject error signal
(c)
Backward pass & completed learning update
Θ
Ψ

FIG. 6. Hamiltonian echo backpropagation in a simple mechanical model. In panel (a), a charged ball Ψ is launched with a velocity representing the input to the machine. Its trajectory is deflected depending on the position of another charged ball Θ, which can freely move along a rail and which experiences a momentum kick due to this interaction. Eventually, as shown in panel (b), the velocities of both balls are reversed, and the ball Ψ is slightly adjusted according to its deviation from the desired outcome (“error signal”). (c) Afterwards, it retraces almost exactly its initial trajectory, delivering yet another momentum kick to the Θ particle. Finally, any remaining velocity of Θ is dissipated and converted into a finite displacement, representing the training update. Of course, the expressive power of this machine, containing only these 2 degrees of freedom, is very limited (i.e., it can only produce very rough approximations to most input-output relations). Nevertheless, it illustrates the generality of the procedure—e.g., an arbitrary (even unknown) static force field acting on the Ψ particle will not spoil the learning procedure.

Nevertheless, our procedure for training can still minimize the cost function. Indeed, consider that we start from a relatively high temperature in the thermal reservoir. The thermal fluctuations in the reservoir would result in random jumps between the stable points. As our training procedure advances, we decrease the temperature of the reservoir, inducing annealing. In the end, if the system converges to a thermal equilibrium, the configuration of Θ would be found in one of the minima of the cost function.

In practice, controllable dissipation may be a too stringent requirement. Actually, we do not need to have control over the dissipation on Θ if the dissipation timescale is large enough. In order to see this, we require that the damping constant is weak enough that its effect can be neglected during one single evaluation step: i.e., the damping timescale Γ^{-1} is much longer than the duration of the evaluation step T. The core of our procedure is to enact an effective force on θ that is proportional to the slope of the cost function. If the dynamics were purely conservative, that would not be sufficient, as Θ would oscillate around the stable manifold without ever converging anywhere. It is then clear that we need such a damping term. In consequence, we distinguish two timescales: the short timescale T of an evaluation step, in which all dynamics is approximately Hamiltonian and time reversible, and the long timescale of the total training time, in which dissipation ensures that, in the end, Θ converges to the stable manifold.

F. Invertible dynamics vs contractive maps

Before going any further, it is useful to introduce the concept of pseudodissipation via ancillary modes. The goal of supervised learning (both for classification and regression) is often to approximate a map from a set of high-dimensional input vectors to a lower-dimensional space. This is an inherently contractive process. Nevertheless, the fact that our theory of SL machines is based on reversible dynamics during the evaluation (forward pass) does not imply that it cannot be used for this purpose. It is always possible to embed contractive maps inside a higher-dimensional reversible map. In physical terms, we can introduce pseudodissipation via ancillary modes: By introducing additional ancillary dynamical variables (which will also be considered part of Ψ in our formalism), we can simulate the effect of dissipation [Fig. 5(b)]. We call it pseudodissipation via ancillary modes because, unlike for real dissipation, we can still time-reverse the whole system, including the ancillary modes.

Aside from its use to learn contractive maps, there is a further reason to include pseudodissipation via ancillary modes in SL machines. In order to be expressive, a SL machine must have a very complex nonlinear dynamics. However, it seems advisable to avoid the strong sensitivity on the input that is implied by chaotic dynamics. When adding dissipation to a conservative chaotic system, it is known that this reduces the sensitivity to initial conditions. In the case of driven systems, where asymptotic Lyapunov exponents can be defined, increasing dissipation leads to an overall shift of Lyapunov exponents to smaller values, thus stabilizing the system [38]. In the case of undriven systems, stronger dissipation enhances the rate for leaving transient chaotic dynamics and settling into fixed points [39]. In our numerical experiments (reported later in Sec. VII), we considered a photonic convolutional neural network with pseudodissipation via ancillary modes. We find that, consistent with these general observations, adding pseudodissipation via ancillary modes helps to reduce the sensitivity to initial conditions and improves the performance of the self-learning machine.

Nevertheless, we remark that pseudocontractive dynamics is not a requirement for HEB, as it can also train devices without such a property. Pseudocontractive dynamics is only a desirable property in some interesting applications of a self-learning machine. However, there are other applications in which it is actually beneficial to have a reversible input-output relation for the relevant degrees of freedom, and hence no pseudocontractive dynamics is required. For example, reversible residual networks have recently been found useful in image classification tasks [40–42].

To avoid confusion, we emphasize that this conceptual split of Ψ into the original evaluation field and additional ancillary degrees of freedom is irrelevant with respect to the following analysis of the echo step. Therefore, to simplify the notation, we just consider the full vector Ψ without specifying how or whether it is split.

V. ANALYSIS OF THE ECHO STEP

We now prove more formally that the crucial ingredient of our scheme, the echo step, indeed works as advertised above, deriving Eq. (8) in the most general setting.

In order to simplify the notation, we define $\Phi \coloneqq \binom{\Psi}{\Theta}$ as the joint configuration of all dynamical variables of the system. The dynamics of Φ is given by the following equation of motion:

$$i\dot{\Phi} = \nabla_{\Phi^*} H, \tag{10}$$

where $H[\Phi]$ is the Hamiltonian. Crucially, we assume H to be time-reversal invariant, which means $H[\Phi] = H[\Phi^*]$ in this notation. Let us remark here that time-reversal invariance is the only assumption in this section (we do not need to assume a degenerate ground state for the expressions derived in this section). The operator ∇_{Φ^*} involves the derivatives $\partial/\partial\Phi_i^*$ and is formally defined as $(\nabla_{\Phi^*} f)_i = \partial f/\partial\Phi_i^*$. We stress that, at the moment, we study the dynamics purely in the absence of coupling to the dissipative reservoir. In the notation of Eq. (1), $H[\Phi]$ corresponds to the first three terms of the SL Hamiltonian, $H = H_\psi + H_{\mathrm{int}} + H_\theta$, without the bath.

A. Expression for the gradient of the cost function in terms of an “advanced” backpropagating perturbation

In a SL machine, the output of an evaluation step is a function of the input $\Psi(-T)$ and the learning field, whose configuration at the initial time is given by $\Theta(-T)$. In the following, we want to obtain a formal expression for the gradient of C with respect to $\Theta(-T)$, which will later form the basis for physical backpropagation. Since Ψ and Θ are treated on an equal footing in Eq. (10), we set out to find the gradient of C with respect to the initial value of the joint state $\Phi(-T)$:

$$\begin{pmatrix} \nabla_{\Phi^*(-T)}C \\ \nabla_{\Phi(-T)}C \end{pmatrix} = D \begin{pmatrix} \nabla_{\Phi^*(0)}C \\ \nabla_{\Phi(0)}C \end{pmatrix}, \tag{11}$$

where the matrix D is defined as

$$D \coloneqq \begin{pmatrix} \nabla_{\Phi^*(-T)} \otimes \Phi^*(0) & \nabla_{\Phi^*(-T)} \otimes \Phi(0) \\ \nabla_{\Phi(-T)} \otimes \Phi^*(0) & \nabla_{\Phi(-T)} \otimes \Phi(0) \end{pmatrix} \tag{12}$$

and $[a \otimes b]_{ij} = a_i b_j$. Here, the operator D obviously maps changes in the output back to corresponding changes in the input, and we now relate it to the Green’s function of the problem.

Suppose that $\Phi(t)$ is the solution of the nonlinear Hamiltonian equations when the initial conditions are given by $\Phi(-T)$. Let us consider what happens if we perturb $\Phi(-T)$ by some small $\delta\Phi(-T)$. This results in a weak perturbation $\delta\Phi(t)$ traveling forward on top of the zeroth-order solution. One can write $\delta\Phi(t)$ as the solution to the linearized equations of motion:

$$L_\Phi(t) \begin{pmatrix} \delta\Phi(t) \\ \delta\Phi^*(t) \end{pmatrix} = \delta(t+T) \begin{pmatrix} -i\delta\Phi(-T) \\ i\delta\Phi^*(-T) \end{pmatrix}. \tag{13}$$

Here, we introduced the linear differential operator

$$\begin{aligned} L_\Phi(t) = {} & i \begin{pmatrix} +1 & \\ & -1 \end{pmatrix} \frac{d}{dt} \\ & - \begin{pmatrix} \nabla_\Phi \otimes \nabla_{\Phi^*} & \nabla_{\Phi^*} \otimes \nabla_{\Phi^*} \\ \nabla_\Phi \otimes \nabla_\Phi & \nabla_{\Phi^*} \otimes \nabla_\Phi \end{pmatrix} H[\Phi(t)] \end{aligned}$$

in compact notation, with $[\nabla_\Phi \otimes \nabla_{\Phi^*}]_{ij} = (\partial^2/\partial\Phi_i \partial\Phi_j^*)$ and so on. If the vector Φ has M components, the operator $L_\Phi(t)$ acts on a space of dimension $2M$. The inhomogeneous term on the right-hand side of the preceding equation is there to enforce the initial conditions, assuming that $\delta\Phi = 0$ for $t \to -\infty$.

The solution of Eq. (13) can be written in terms of the retarded Green’s function associated with $L_\Phi[t]$:

$$\begin{pmatrix} \delta\Phi(t) \\ \delta\Phi^*(t) \end{pmatrix} = \mathcal{G}_\Phi(t,-T) \begin{pmatrix} i\delta\Phi(-T) \\ -i\delta\Phi^*(-T) \end{pmatrix}. \tag{14}$$

The Green’s function $\mathcal{G}_\Phi(t,-T)$ is a linear operator such that

$$L_\Phi(t')\mathcal{G}_\Phi(t',t) = \delta(t'-t)\mathbb{I}, \tag{15}$$

where $\mathbb{I}$ is the identity matrix of size $2M$. Note that both L_Φ and $\mathcal{G}_\Phi(t,-T)$ depend on the zeroth-order solution $\Phi(t)$ of the nonlinear dynamical equations, around which we have linearized. We choose $\mathcal{G}_\Phi(t,-T)$ to be the *retarded* Green’s function, meaning that $\delta\Phi(t)$ originates at $t=-T$ and propagates in the forward direction of time. Causality dictates that $\mathcal{G}_\Phi(t',t) = 0$ for $t' < t$.

We now connect the matrix D to the Green’s function. From the definition of D in Eq. (12), we see it measures the change at time $t=0$ when introducing a small change at time $t=-T$. The same relation is expressed by the Green’s function, as defined in Eq. (15), except for some constant factors. Merging both equations, we see that

$$\mathcal{G}_\Phi(0,-T) = -D^\dagger i\sigma_z, \tag{16}$$

where σ_z is the Pauli matrix. Therefore, we have $D = -i\sigma_z\mathcal{G}_\Phi(0,-T)^\dagger$, and we can now express the change of the cost function in terms of the Green’s function. However, we see presently that it is helpful in terms of physical interpretation to introduce the *advanced* Green’s function for this purpose, which also obeys Eq. (15) but has $\mathcal{G}^{\mathrm{adv}}(t',t) = 0$ for $t' > t$ and represents a signal going backward in time. It is related to the retarded Green’s function via

$$\mathcal{G}_\Phi^\dagger(t',t) = \mathcal{G}_\Phi^{\mathrm{adv}}(t,t'). \tag{17}$$

Combining Eqs. (11), (16), and (17), we finally obtain a formula relating the gradient of C that is required for the learning update to the advanced Green’s function and the (easily obtained) gradient of C with respect to the output fields:

$$\begin{pmatrix} \nabla_{\Phi^*(-T)}C \\ \nabla_{\Phi(-T)}C \end{pmatrix} = -i\sigma_z\mathcal{G}_\Phi^{\mathrm{adv}}(-T,0) \begin{pmatrix} \nabla_{\Phi^*(0)}C \\ \nabla_{\Phi(0)}C \end{pmatrix}. \tag{18}$$

This is still a formal expression at this point, but in the next section, we show how it can be implemented physically.

B. Time reversal of the perturbation: Physical backpropagation

How can we produce the signal of Eq. (18) in practice, in order to allow the training update to take place? Apparently, we need to inject the source term $\nabla_{\Phi^*(0)}C$ as a perturbation and follow its “backwards time evolution.”

Let us consider that, at $t = 0$, we have obtained the output Φ_{out}. Suppose that we have a way to implement an interaction Hamiltonian $C[\Phi]$ that is proportional to the cost function (written as a function of the output). Let us further suppose that we can control this interaction Hamiltonian, so we can switch it on during some short period of time and later switch it off again. In theory, this is always possible because a cost function is just a real function of the output. In practice, we see that, depending on the cost function, this corresponds to driving the system, or applying phase shifts, or, for more complicated cost functions, to realizing some nonlinear interactions. In any case, this operation is a function only of the output, not requiring any knowledge of or feedback on the internal degrees of freedom of the autonomous self-learning machine. As we discuss later, a practical implementation is possible for several standard cost functions with relatively simple setups. For the moment, we consider the general case in which C is an arbitrary cost function. When we realize such an interaction Hamiltonian C, according to Hamilton's equations, the evolution will be given by $i\dot{\Phi} = \nabla_{\Phi^*} C$. Let us suppose that we only realize such interactions during an infinitesimally small time interval ϵ. The result of such a brief Hamiltonian evolution is to update the fields as $\Phi(0) \mapsto \Phi(0) - i\epsilon\nabla_{\Phi^*} C$.

The advanced Green's function evolves a perturbation of the output backwards in time to obtain the corresponding perturbation of the input. In order to realize this physically, we need to induce the time-reversed evolution of the system. In a time-reversal-invariant system (such as the one considered here), this is possible by implementing a time-reversal operation and then letting the natural dynamics proceed from there forwards in time. In the present complex notation, we know that a time reversal is equivalent to phase conjugation, $\Phi \mapsto \Phi^*$, flipping the momenta. By means of phase conjugation, applied to the already-perturbed field, we produce a perturbed echo given by $\Phi^*(0) + i\epsilon\nabla_{\Phi} C \equiv \Phi_{\mathrm{echo}}(0) + \delta\Phi_{\mathrm{echo}}(0)$.

Importantly, the whole configuration of the system is time reversed, and, as a consequence, the evolution for $t \in [0, T]$ is, to leading order, given by the fully time-reversed nonlinear evolution: $\Phi_{\mathrm{echo}}(t) = \Phi^*(-t)$. The first-order correction $\delta\Phi_{\mathrm{echo}}(t)$ is the linear response of the system to the perturbation $i\epsilon\nabla_{\Phi} C$ in the initial conditions, evolving on top of $\Phi^*(-t)$.

Now, we come to the crucial step. In any time-reversal-invariant system, the advanced and retarded Green's functions of linear perturbations propagating on top of the original nonlinear dynamics and its time-reversed counterpart, respectively, are connected in the following way:

$$\mathcal{G}^{\mathrm{adv}}_{\Phi}(t', t) = \mathcal{T}\mathcal{G}^{\mathrm{ret}}_{\Phi_{\mathrm{echo}}}(-t', -t)\mathcal{T}, \tag{19}$$

where $\mathcal{T} = \sigma_x$ is the time-reversal operation, interchanging $\delta\Phi$ and $\delta\Phi^*$. Therefore,

$$\mathcal{G}^{\mathrm{adv}}_{\Phi}(-T, 0)\begin{pmatrix} \nabla_{\Phi^*(0)} C \\ \nabla_{\Phi(0)} C \end{pmatrix} = \mathcal{T}\mathcal{G}^{\mathrm{ret}}_{\Phi_{\mathrm{echo}}}(T, 0)\begin{pmatrix} \nabla_{\Phi(0)} C \\ \nabla_{\Phi^*(0)} C \end{pmatrix}. \tag{20}$$

It is now obvious that the left-hand-side, which (as we have already found above) formally yields the update needed for the gradient descent, can be *physically* implemented in terms of the right-hand-side version, with a suitable perturbation injected and propagated *forward* in time on top of the echo. Finally, as we can see from Eq. (20), at $t = T$, we have to perform a final time reversal (phase conjugation) again.

Overall, after the whole process is completed, we obtain a replica of the initial configuration at $t = -T$ but, importantly, with a small correction that is proportional to ϵ. The result of the whole physical process can be summarized as

$$\begin{pmatrix} \Phi(T) \\ \Phi^*(T) \end{pmatrix} = \begin{pmatrix} \Phi(-T) \\ \Phi^*(-T) \end{pmatrix} - \mathcal{T}\mathcal{G}^{\mathrm{ret}}_{\Phi_{\mathrm{echo}}}(T, 0)\begin{pmatrix} \epsilon\nabla_{\Phi(0)} C \\ \epsilon\nabla_{\Phi^*(0)} C \end{pmatrix}. \tag{21}$$

We can convince ourselves that this is the learning update that was needed by reformulating the deviation (second term on the right-hand side) with the help of Eq. (20) and, subsequently, Eq. (18). This produces

$$\Phi(T) - \Phi(-T) = -i\epsilon\nabla_{\Phi^*(-T)} C, \tag{22}$$

which is exactly the learning update that was needed, according to Eq. (8), for the learning parameters Θ. Incidentally, the evaluation degrees of freedom Ψ have also changed, but they will be discarded when the next input is injected into the learning machine.

VI. MAIN INGREDIENTS OF IMPLEMENTATIONS

The most important feature of our new method is its generality. It can be used to train any time-reversible Hamiltonian physical system. It is not our intention here to give a full description of all possible implementations of the SL machine. On this general level, we cannot even decide which ones are optimal, as the most suitable architecture is heavily dependent on the machine learning task of interest. Nevertheless, we can give some guidelines.

The main ingredients of any SL machine of the type presented here are as follows: a time-reversal-invariant Hamiltonian system with two sets of dynamical variables Ψ and Θ, a device that can time reverse Ψ and Θ, and a device or method to transform the output according to the cost function Hamiltonian (injecting the error signal). In this section, we discuss each of these ingredients on a general level, and we provide a summary of some of the most promising possibilities for their realization.

A. Nonlinear core

As emphasized above, HEB can work with an arbitrary, completely unrestricted Hamiltonian, as long as time-reversal invariance is maintained.

Any actual implementation will typically be based on local interactions between the degrees of freedom. Since for practical applications it is desirable to have designs that are scalable, a SL device, in practice, should be constructed by a combination of simple elements. For example, we can think of integrated photonics or superconducting circuits.

The details of particular experimental platforms will be discussed in Sec. VIII. Here, we briefly list some general guidelines. In order to attain a nonlinear input-output relation that is expressive and useful for machine learning, we require that (1) many degrees of freedom interact during the forward pass and (2) there are nonlinear interactions whose effect is strong during such a pass. A common question in this regard would be the effect of chaos. Chaotic dynamics is defined as a property of the long-time limit while here we are considering a finite evolution time for the forward pass. In practice, as we will see in the numerical examples, this is a matter of degree: It is useful to have sufficiently strong nonlinear effects, as long as they do not become too sensitive to small deviations in initial conditions.

A device combining linear local couplings and local nonlinearities is enough to produce extremely rich dynamics. For example, in optics, one can use a combination of beam splitters and nonlinear layers to reproduce the architecture of a feedforward neural network [15]. However, there is no reason *a priori* to restrict ourselves to such a layout: Nonlinear wave fields interacting with each other inside some extended medium do not possess any obvious layered structure, but they are still compatible with HEB, just like arbitrary lattices of coupled degrees of freedom.

One important requirement for HEB, is, of course, a time-reversal-invariant Hamiltonian. Practically, this means that dissipation in the course of the forward pass must be as low as possible, which already rules out some physical devices where a strong attenuation is unavoidable (e.g., when nonlinearities are based on absorption of waves). Nevertheless, even for promising platforms, such as coherent nonlinear optics, there is some remaining level of dissipation. In the limit of large-scale networks, even a small amount of attenuation could be problematic since its effect increases linearly with the propagation length. However, since we are considering classical devices, there is, in principle, no obstacle to employ amplification in order to compensate for the loss present in the device (we discuss the aspect of dissipation further when assessing concrete experimental platforms).

B. Time-reversal operation

The second key ingredient of HEB is the ability to perform a time-reversal operation after the forward pass. Such an operation does not represent purely Hamiltonian evolution, i.e., it does not preserve the Poisson brackets of coordinates and momenta. In practice, this means the operation necessarily requires dissipation. As opposed to the case of quantum mechanics, in classical systems, this requirement poses no fundamental obstacle. However, a physical and fast implementation is not necessarily straightforward in all classical systems. Fortunately, a well-known solution in the case of nonlinear optics [43] is phase conjugation (for a review of this extended field, see Ref. [44]). It is well established that phase conjugation is not particular to optics. In fact, it can be engineered in any wave field where one can realize three- or four-wave mixing. For example, it has been demonstrated with acoustic waves [45], matter waves [46], microwaves [47], and spin waves [48].

The fact that phase conjugation can be used to time reverse wave pulses has long been understood (see, e.g., Ref. [49] for the optical domain and Ref. [50] for atomic matter waves). The reversal of time evolution has been demonstrated in several nonlinear systems, including nonlinear optics [51,52], electromagnetic wave-chaotic systems [53], and water waves [54].

Moreover, in systems where phase conjugation is not straightforward to achieve physically, it is often possible to measure the output and inject a phase-conjugated replica. We note that this would replace one part of the HEB procedure by external processing and feedback, but the overall scheme could still be advantageous since we still benefit from physical backpropagation and a physical parameter update. This last possibility is know as digital phase conjugation, and it has been demonstrated with remarkable success in optical setups [55], using electro-optic components.

Although the concepts of implementations of phase conjugation are well established, it may be helpful to recall the principle here. From a mathematical point of view, for any time-reversible Hamiltonian, the evolution can be reversed by inverting the momenta: $\boldsymbol{\pi} \rightarrow -\boldsymbol{\pi}$. In our complex notation, this means performing a phase conjugation of the fields: $\Phi \rightarrow \Phi^*$. As we remarked above, this step cannot be realized with purely Hamiltonian evolution, not even by breaking time-reversal symmetry momentarily. A time-reversal operation does not preserve the Poisson brackets, so it is not a canonical transformation. However, one can circumvent this problem by using ancillary modes.

For simplicity, we now discuss a situation with discrete coupled modes of the wave field; the extension to extended waves is straightforward. Let us assume that we have a dynamical variable Φ that we want to phase conjugate. Additionally, we also have a second, ancillary dynamical variable Ξ. An important assumption is that the initial value of the ancilla is $\Xi_{\mathrm{in}} = 0$. The phase conjugation can then be realized in three steps. First, we couple Φ and Ξ in such a way that, after a fixed time, we have $\Xi = \Phi_{\mathrm{in}}^*$. This step can be realized with Hamiltonian dynamics. For example, a Hamiltonian of the "parametric interaction" form $H = \chi(\Phi\Xi + \Phi^*\Xi^*)$ would result in the desired outcome after an interaction time $t = \chi^{-1}\log(1+\sqrt{2})$. Second, we swap both modes, so $\Phi = \Phi_{\mathrm{in}}^*$. This step can also be the result of Hamiltonian evolution, now choosing a beam-splitter Hamiltonian, $H = \chi(\Phi\Xi^* + \Phi^*\Xi)$. Third, we erase the state of the ancilla (this is where dissipation enters), so $\Xi = 0$ again.

C. Implementation of the decay step

A third ingredient of our scheme is the ability to transform the physically backpropagated perturbation into an update of the learning degrees of freedom. More specifically, the goal of the decay step is to shift the learning degrees of freedom according to the momenta (which, in turn, are proportional to the gradient of the cost function) and, at the same time, dissipate (only) the momenta. Mathematically, we want to produce the following transformation:

$$\theta' = \theta - \eta\nabla_\theta C, \tag{23}$$

$$\pi_\theta' = 0, \tag{24}$$

where θ' and π_θ' are the learning field quadratures after the decay step, and η is the learning rate.

Importantly, this operation can be implemented by means of a sequence of three elementary transformations. Since all the most promising platforms we conceive are based on wave fields, we provide here an outline of its implementation for a wave-field mode representing a learning degree of freedom.

The transformation given by Eq. (23) can be approximated by a succession of three elementary steps. In the first step, we couple θ and π_θ by means of a phase shift. In the second step, Θ undergoes parametric amplification, a routine transformation in nonlinear optics and wave platforms, in general. Finally, there is a step of attenuation. The overall transformation as a result of these three steps is given by

$$\theta' = \sqrt{AG}\big(\cos(\varphi)\theta + \sin(\varphi)\pi_\theta\big), \tag{25}$$

$$\pi_\theta' = \sqrt{\frac{A}{G}}\big(\cos(\varphi)\pi_\theta - \sin(\varphi)\theta\big), \tag{26}$$

where φ is the phase-shift angle, G is the parametric gain, and A is the attenuation.

If we set $\sqrt{A/G} \ll 1$, we have $\pi_\theta \approx 0$. We need to compensate for the attenuation of θ by tuning the parametric amplification. By choosing $\cos(\varphi)\sqrt{AG} = 1$, we attain the desired transformation given by Eq. (23). Since $\pi_\theta = -\epsilon\nabla_\theta C$, the learning rate is then given by $\eta = \epsilon\tan(\varphi)$. Note that the learning rate can be controlled by tuning the phase shift φ (and tuning the parametric gain accordingly). Importantly, the learning rate can be controlled independently of the size ϵ of the injected error signal since we always have the free parameter φ available. This is useful since the values of ϵ should be neither too large (breakdown of the linearization approximation for the backpropagated error signal) nor too small (negative impact of extra noise).

This is not the only possible practical implementation of the decay step. Observe that the desired transformation can also be written in complex form as $\Theta' = (\exp^{-i\varphi}\Theta + \exp^{i\varphi}\Theta^*)/2\cos(\varphi)$ [a straightforward calculation shows that the learning rate is again $\eta = \tan(\varphi)\epsilon$]. It follows immediately that the decay step can be realized by a combination of linear unitary operations (i.e., beam splitters), phase conjugation, and phase-insensitive amplification.

Alternatively, observe that the transformation given by Eq. (23) could also be obtained as the evolution of a mechanical degree of freedom θ with ballistic propagation under the influence of friction. In this case, the initial momentum π_θ is translated to a shift in θ (proportional to the inverse of the mass) while the momentum is dissipated in the long-term limit as a result of the friction. Therefore, the decay step is straightforward to implement in the case of mechanical degrees of freedom (the problem with mechanical implementations is that there is no straightforward method to realize the time-reversal operation, unless by measurement and feedback).

D. Learning signal

The cost function is the fourth essential ingredient in the design of a SL machine. In our prescription for HEB, one crucial step is to inject the learning signal by perturbing the output fields according to $\Psi(0) - i\epsilon\nabla_{\Phi^*}C$, where C is a Hamiltonian that is proportional to the cost function. For arbitrary cost functions, this could always be achieved via some external processing and feedback, but as a general rule, of course, we want to avoid such steps. For simple cost functions, it could be possible to engineer a physical system whose evolution is actually described by the cost function Hamiltonian C.

In the particularly important case of the mean square error (MSE) cost function, a physical implementation is relatively easy. The MSE cost function is defined as the squared difference between the actual output and the target output:

$$C[\Psi(0)] = \int dx\, g(x)|\Psi(x,0) - \Psi_{\mathrm{target}}(x)|^2. \tag{27}$$

Here, $g(x)$ is a position-dependent weighting factor, which is zero in the areas that correspond to the ancillary degrees of freedom (which will be time reversed but not compared against any target). Expanding the square, we can write the corresponding Hamiltonian as $C[\Psi(0)] = g(x)|\Psi(0)|^2 - 2g(x)\mathrm{Re}\{\Psi(0)\Psi^*_{\mathrm{target}}\}$ (where we have dropped $|\Psi_{\mathrm{target}}|^2$, which is just an irrelevant constant term). In terms of a physical wave field, the first term $g|\Psi(0)|^2$ is just a shift in the frequency of the field. Its effect on $\Psi(0)$ is to induce a phase shift given by $e^{-i\epsilon g(x)}$. The only reason we do not completely neglect this phase shift is that $g(x)$ may be spatially inhomogeneous. The second term $-2g\mathrm{Re}\{\Psi(0)\Psi^*_{\mathrm{target}}\}$ is nothing but a drive proportional to the target. In other words, this term can be realized straightforwardly by injecting a weak copy of the target field distribution, adding it to the wave field Ψ, and realizing the corresponding phase shift set by $g(x)$.

Alternatively, the MSE cost function can also be realized using the intensity of the field. Now, instead of forcing $\Psi(0)$ to match a target field, we only try to match a target intensity distribution. In that case, the cost function Hamiltonian is given by $C[\Psi(0)] = g|\Psi(0)|^4 - 2g|\Psi(0)|^2 I_{\mathrm{target}}$. In that case, the experimental realization of $C[\Psi(0)]$ is reduced to a homogeneous Kerr nonlinearity combined with a phase mask proportional to the target intensity pattern.

Other cost functions may be realized with different interactions. Which one is optimal would depend on the problem at hand and how the output is encoded in the physical fields. In many cases, the choice of the cost function would not be a critical feature.

With regard to the amplitude ϵ of the injected perturbation (error signal), on the one hand, it has to be small enough so that the weak perturbation is approximately in the linear regime. On the other hand, in a noisy device, it is beneficial to have a larger amplitude, in order to overcome the effect of noise. Therefore, it is advisable to choose the amplitude of the perturbation as large as possible while remaining in the linearized regime. Once ϵ is chosen, the learning rate is not yet fixed: It can be controlled by changing the free parameters in the decay step. When the decay step and the error signal injection are performed outside the nonlinear core, there is, in principle, no restriction placed on those free parameters, and hence the learning rate can be tuned at will.

VII. ILLUSTRATION IN NUMERICAL EXPERIMENTS

In this section, we illustrate Hamiltonian echo backpropagation in two specific examples by means of numerical simulations (our simulations were performed using PYTHON and NUMPY). Both examples rely on nonlinear wave propagation, of the kind that might be implemented in photonic neural networks. In the first case, we consider a self-learning machine with a very simple setup, and we train it to learn a logical function. In the second example, we consider a more elaborate architecture inspired by convolutional neural networks. We train the self-learning machine to perform classification on the standard MNIST handwritten digits data set. We believe that a more sophisticated architecture or other choices of the activation function and the cost function, together with a careful tuning of the hyperparameters, could improve the results even further beyond the validation accuracy of 97.5% observed here. However, our primary goal here is to illustrate the concept, rather than constructing the optimal setup. In both scenarios, we simulated the full Hamiltonian echo backpropagation procedure, involving both the forward and the backward pass.

The setup considered in the first example consists of a nonlinear wave field. The dynamics of the evaluation field Ψ is given by the nonlinear Schrödinger equation coupled to the learning field,

$$i\dot{\Psi} = \frac{\beta}{2}\nabla^2\Psi + (\chi\theta + g|\Psi|^2)\Psi. \tag{28}$$

The learning field Θ permeates the nonlinear medium, and it interacts with the evaluation field through an interaction of the $\chi^{(2)}$ type. This could be implemented in a nonlinear optical setup or, alternatively, in a device based on matter waves. The evaluation field is initially prepared in a superposition of wave packets that encode the input (Fig. 7), with initial momentum $k_z > 0$. Thus, they propagate from left to right, accumulating the effect of the nonlinear self-interaction and the interaction with the learning field. The output intensity is measured at two points, $\mathbf{x}_0^{\mathrm{out}}, \mathbf{x}_1^{\mathrm{out}}$. The output is considered a logical 0 if $|\Psi(\mathbf{x}_0^{\mathrm{out}}, 0)| > |\Psi(\mathbf{x}_1^{\mathrm{out}}, 0)|$; otherwise, it is a logical 1 (for more details, see Appendix A). The goal is to learn the exclusive-or (XOR) function mapping from input to output. As can be seen in Fig. 7, this is achieved rapidly.

The setup that we consider for the second example is inspired by the photonic neural network introduced theoretically by Ong *et al.* [56]. We train this device to learn MNIST handwritten digits classification. It is composed of three convolutional layers followed by a dense, fully connected layer. The nonlinearity is provided by a $\chi^{(2)}$ optical nonlinearity, as described in Eq. (F1). The convolutional layers can be implemented by a combination of discrete Fourier transforms (DFTs) and learnable phase shifts. The proposal by Ong *et al.* did not address the problem of how to implement the dense layers; we introduce an implementation of the dense layers that is based on a sequence of DFTs, learnable phase shifts, and a linear interaction with ancillary degrees of freedom (see Appendix B). As a cost function, we choose the mean square error on the intensity.

In this setup, the learning parameters are all in the form of phase shifts. We assume that these phase shifts are produced by an interaction of the type $H_{\mathrm{int}} = \theta\Psi^*\Psi$ inside some optical cavities through which the evaluation wave field travels.

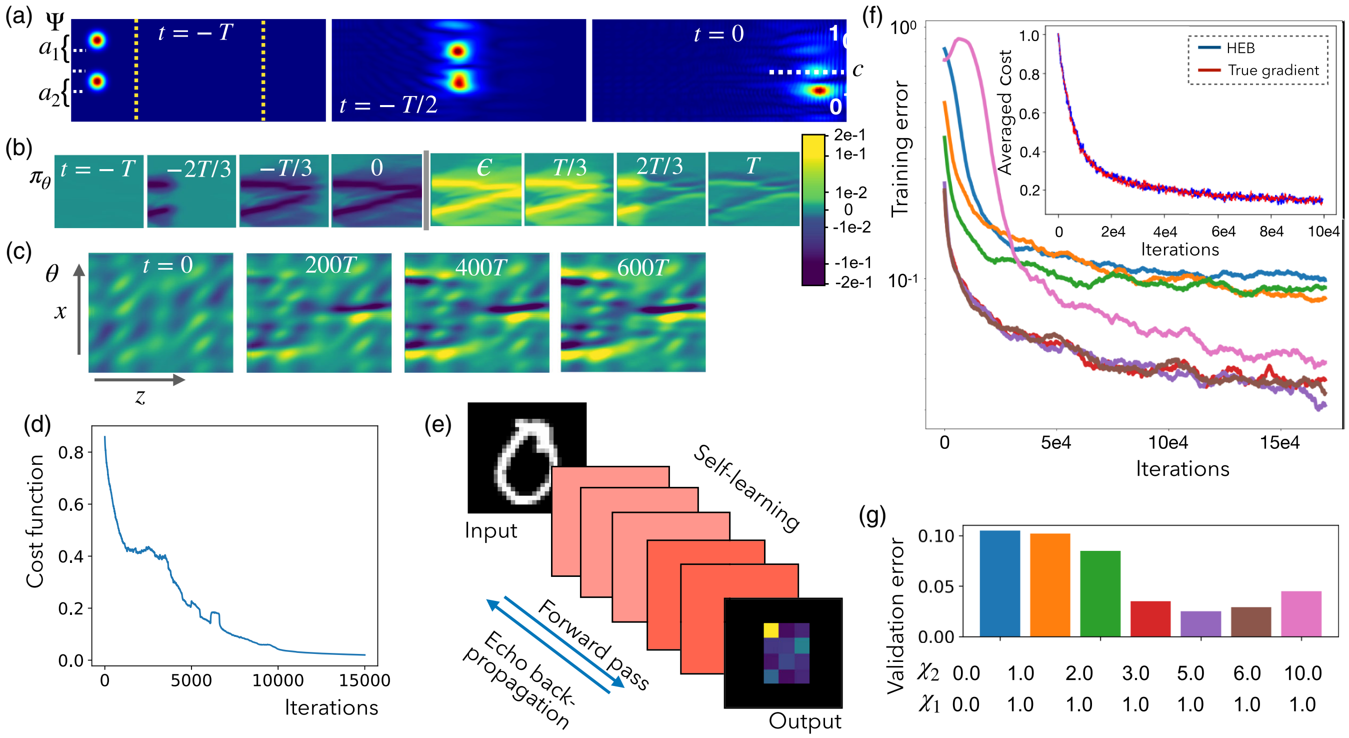

FIG. 7. Training a self-learning machine (simulations). (a) Time evolution of the “evaluation field” intensity $|\Psi|^2$ during a forward pass for a machine that has learned to implement XOR ($b = a_1 \oplus a_2$). The region in which the interaction with the learning field occurs is enclosed by yellow dashed lines. (b) Time evolution of the learning field momentum π_θ during a single training step. The pattern builds up from left to right as the Ψ pulse propagates. It is time reversed and then partially deleted as the Ψ pulse travels back. The finite remainder at time T is due to the injected error signal, and it produces the desired learning update. (c) Evolution of the learning field θ during many training steps. The training starts with a smooth random configuration, and there are 600 HEB steps (each for a different, randomly chosen training sample). (d) Cost function evolution during a particular training run for the model of panels (a)–(c). (e) Setup of a self-learning machine that learns to classify handwritten digits (MNIST). The outcome is a one-hot encoding of the detected digit; i.e., one of ten patches “lights up.” (f) Training error evolution of the MNIST setup for different strengths of the wave-field nonlinearity. The line colors represent a different value of the parameters χ_1, χ_2. Note that χ_1 and χ_2 are the nonlinear strengths at the convolutional and dense layers, respectively. The color that represents each value of χ_1, χ_2 can be seen from panel (g). In the inset, we display the cost function evolution when the SL machine is trained via HEB (blue line). We compare it with the gradient descent according to the true gradient (red line), with the same learning rate. The true gradient is obtained via autodifferentiation. In order to better compare the results, we average the cost function evolution over ten different trajectories. (g) Final validation set classification accuracy.

In our simulations, the best results were obtained by choosing values of the nonlinear coupling strength that are different in the convolutional and the dense layers. This can be explained by the fact that the intensity decays as the wave packets propagate, as a result of the interaction with the ancilla modes. Therefore, it is optimal to choose that the nonlinearity in the dense layer, χ_2, is stronger than the nonlinearity in the convolutional layers, χ_1. The results in Fig. 7 show the training progress for several values of χ_1 and χ_2. First, we show the linear case, when $\chi_1 = \chi_2 = 0$, and then we sweep across several values of χ_2, with fixed χ_1. One can see a sharp transition in the achievable accuracy when χ_2 is increased. When the nonlinearity surpasses some threshold ($\chi_2 \approx 3$), the accuracy jumps from around 90% to above 95% [Fig. 7(g)]. This result suggests that, at this point, the device gains a significantly better expressivity. Our observations are in qualitative agreement with the findings by Marcucci *et al.* [57], where the expressive power of information processing using nonlinear wave fields was analyzed in terms of the strength of the nonlinearity.

We now study the effects of noise and nonlinear effects in the gradient estimation for the case of the XOR scenario.

First, we study the consequence of nonlinear effects in the gradient estimation—i.e., what happens for larger values of ϵ. The backpropagating signal is proportional to the gradient to first order in ϵ, but for large values of ϵ, the second-order terms could be relevant. In Fig. 8(a), we show the relative error in the update (with respect to the ideal gradient descent) as a function of ϵ. The relative error is defined as $|\Delta\theta_j/\eta - \partial_{\theta_j} C|/|\partial_{\theta_j} C|$, where $\Delta\theta_j$ is the update in one training step.

From the previous paragraph, it would seem that it is optimal to use the smallest possible value of ϵ. However, in

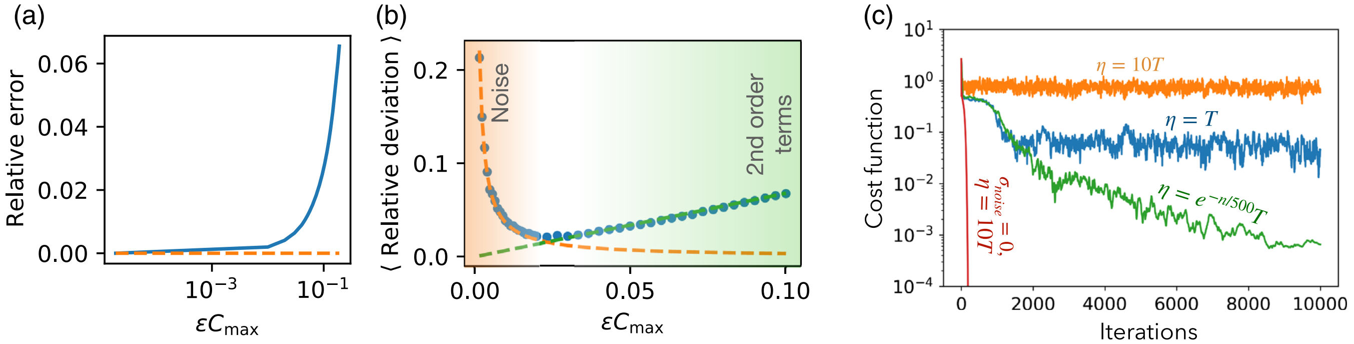

FIG. 8. Effects of noise and nonlinear effects in the gradient estimation. (a) Relative error in the approximation of $\partial_{\theta_j} C$, where θ_j is the value of the learning field in the middle of the evolution domain. We compare the update of θ_j obtained via HEB with an accurate estimation that was computed by means of the finite difference method. The result is presented as a function of the dimensionless quantity $\epsilon C_{\max}$, where $C_{\max}$ is the maximum possible value of the cost function. (b) Root-mean-square error of the update obtained by means of HEB in the presence of noise. On the one hand, for small values of ϵ, the noise overwhelms the term proportional to the gradient. (c) Analyzing the influence of a noisy time-reversal operation on training. Here, we consider a discretized version of the nonlinear Schrödinger equation, where the effect of noise is more prominent. We display the cost function during training for different values of the learning rate, at a fixed finite noise level whose standard deviation is given by $\sigma^2_{\text{noise}} = 0.05I$ (see Appendix A for details).

a real self-learning device, there may be noise (introduced during the time-reversal operation or other steps of the scheme). If ϵ is too small, noise will introduce significant errors into the gradient estimate provided by HEB. We analyze the situation in Fig. 8(b). The root-mean-square error is defined as $\langle |\Delta\theta_j/\eta - \partial_{\theta_j} C|^2 / |\partial_{\theta_j} C|^2 \rangle^{1/2}$, where the average is taken with respect to the noisy signal. On the other hand, for larger values of ϵ, the previously analyzed nonlinear effects dominate. The orange and green dashed curves are a fit to the expected behavior of the root-mean-square error in the limits of dominant noise (1ϵ) and dominant nonlinear-order effects (ϵ).

We now analyze how to suppress the effects of noise. The operation of a SL machine is independent of the Hamiltonian, and therefore, one does not need to know the particularities of the device (including any possible deviations from intended fabrication) to make it work. This is true because the time-reversal operation produces an exact inversion of the Hamiltonian evolution, for any time-reversible Hamiltonian. One might then wonder how sensitive HEB becomes to the precision of the time reversal. In fact, since the nonlinear processes used to phase conjugate signals can also amplify noise, there are practical limits to the performance of a time-reversal mirror. Nevertheless, we show here that the effects of noise in the time reversal can be mitigated, although at the price of a longer training time. In Fig. 8(c), we compare the results using a perfect time reversal (red line) with the results obtained in the case of a noisy phase conjugation, for several values of the learning rate (orange and blue), as well as for a suitably chosen time-dependent learning rate ("learning rate schedule," an approach also known for ANNs, with n denoting the training step index). As could be expected, the performance of the SL machine deteriorates when the time reversal is noisy. However, one can observe that decreasing the learning rate improves the final accuracy because the overall update after several steps will approximately cancel out the random noise if the learning rate is small enough. Indeed, the backpropagating perturbation is a linear function of the error signal to first order in ϵ. Therefore, the update of Θ in each step will be the sum of the appropriate gradient update plus a random signal that is proportional to the noise in the time reversal. If each update step does not change the value of Θ significantly, the overall change after several steps will average out the unwanted noise. Note that this behavior only applies to temporally fluctuating noise. In the case of static systematic deviations from the correct time-reversal operation, an independent calibration of that operation would be required. Importantly, however, it would not rely on any knowledge of the internal dynamics of the SL machine.

We note that in Fig. 8(c), we studied a discretized version of the nonlinear wave-field device considered above. The domain of the Ψ and Θ fields is now a discrete lattice of size 20×30 (we do not require that the fields are approximately continuous). The reason for this choice is that the cost function converges much closer to the optimal value. In this way, we can appreciate the effect of noise on convergence more clearly.

VIII. IMPLEMENTATION CONSIDERATIONS AND POTENTIAL HARDWARE PLATFORMS

The elements for a SL machine that we have presented so far can be realized in many platforms. In fact, many of the ingredients needed for implementations of our proposal have already been demonstrated experimentally. In some cases, the physical platforms that we consider below were already studied in the specific context of building new ML hardware, but it was not known how to train them successfully with physical procedures, without the use of external computation and feedback. Hamiltonian echo backpropagation can now provide a means to do this. Our only aim here is to give a brief overview of the possibilities. It is clear that for each platform, substantial further research will be required to come up with an optimized design respecting the particular hardware constraints and to analyze the potential performance, as well as to take steps towards first experimental realizations.

A. General remarks on speed, energy efficiency, and scaling

In general, neuromorphic devices are designed to offer intrinsic parallelism, which means the time needed for one evaluation becomes independent of the size of the input vector. HEB builds on the same advantage, with all the physical degrees of freedom (e.g., wave fields) evolving in parallel.

In many big-data applications, an important bottleneck for the speed of training may be the RAM bandwidth. For large ANNs, storing or retrieving the updated weights at each step can be the most important contribution to the training time. This points to a considerable potential advantage of training via HEB: The learning field is updated at the same time that the evaluation field propagates in the device, so there is no overhead associated with memory retrieval.

Whether a self-learning machine can be more energy efficient than existing systems would depend of course on the particular physical system used to implement it. It has been claimed by other authors that some physical neuromorphic platforms could be significantly more efficient than electronic devices. For example, in photonic integrated circuits, the scaling of the power consumption can be more favorable [14]. We argue that, if we use one of those promising physical platforms, which compare favorably with digital devices, then training by means of HEB is a particularly efficient training method. For example, it is definitely more efficient than using a method based on parameter shifts and feedback. Indeed, applying the parameter shift method in a device with N parameters requires $O(N)$ forward passes, while we only require $O(1)$.

There is a further reason why we believe that self-learning machines trained via HEB may be more efficient than electronic devices in the case of large networks. In many of the most successful deep learning architectures, the same parameters are reused in many parts of the network. This can also be done in a physical learning machine of the type we envisage here, when the learning field interacts repeatedly with the evaluation field. This implies that the number of elementary operations scales faster than the number of parameters. For example, a convolutional filter of size M applied on an N-dimensional vector requires $O(MN)$ elementary operations. For L layers, the number of elementary operations would thus scale as $O(LMN)$. While for an electronic device the energy consumption is roughly proportional to the number of elementary operations, the scaling of the energy consumption is different for a time-reversible machine. During the forward pass, the energy is conserved (the dissipation is ideally nonexistent or at least very low). The main energy consumption comes only from the time-reversal operation and the decay step, as well as the preparation of the initial states and injection of the error signal. The overall unavoidable energy consumption for a time-reversible machine thus scales like $O(N+M)$, where N and M are the dimension of the input vector and of the vector of learning parameters, respectively. Therefore, in general, the scaling of the energy consumption as a function of the system size is more favorable in those cases where the number of elementary operations scales faster than the number of parameters.

B. Nonlinear optical platforms

Optical approaches to neuromorphic computing have recently attracted a lot of interest because they promise significant advantages in terms of speed, energy efficiency, or parallelism [15,16,36]. Importantly for us, those schemes involve a combination of coherent (linear and nonlinear) operations that are practically lossless, and hence they are amenable to our proposal. Optical SL machines would benefit from high parallelism and fast all-optical nonlinearities.

In the nonlinear-optics approach to neuromorphic computing, on-chip photonic circuits are one of the most promising technologies for ML hardware. As discussed in Sec. VI, a scalable implementation of a nonlinear core would be a combination of linear local couplings and local nonlinearities. A straightforward implementation of a self-learning machine would be to have both the evaluation field Ψ and the learning field Θ be sent as pulses into an arrangement of waveguides, beam splitters, and cavities (some of them nonlinear), and to implement time reversal outside this "nonlinear core," as described in Sec. VI. The most direct implementation of a feedforward neural network would include layers of linear transformations alternated with nonlinear layers implementing the nonlinear function. Along this line, there are several experimental demonstrations and theoretical proposals of deep neural networks based on programmable photonic circuits (e.g., Ref. [15]). The nonlinear activation functions could be provided, for example, by propagation inside a material with a strong Kerr effect, possibly enhanced via the use of resonators [58].

The time-reversal operation could be implemented in any of the approaches discussed for fast phase conjugation in the nonlinear-optics literature, specifically, by employing nonlinear optical interactions acting on the pulses propagating along waveguides or fibers or, alternatively, inside nonlinear cavities with multiple interacting modes.

We have already discussed the energy efficiency of HEB, in general. Specifically for optical neural networks, it has been suggested [15] that they can be orders of magnitude more efficient in terms of power than their electronic counterparts. An estimate of the power consumption would depend on the details of the particular implementation, but in any case, the matrix multiplications and nonlinear activation functions are realized in the time-reversible, ideally lossless manner. The dissipative steps (phase conjugation, decay step, injection of input) require the dissipation (and/or resupply) of an energy amount on the order of the energy contained in the optical pulses of the learning field and evaluation field. This follows from the way these fundamentally dissipative operations have to be implemented, as described before. We may estimate a typical energy on the order of 1 fJ per wave packet, where we assume the packet to be a 1-mW pulse of 1 ns duration, giving rise to sufficient nonlinearities in typical optical nonlinear resonators. The number of wave packets scales with the total number of learning parameters and size of the input vector, which gives an estimate for the total amount of energy dissipated per pass.

With respect to the footprint of a photonic implementation, the limiting factor would be the size of elementary building blocks, such as beam splitters and resonators. If nonlinear effects are to take place during propagation along waveguides (instead of inside resonant cavities), then we also have to find the minimum length to impart a strong nonlinear effect (e.g., a Kerr phase shift of π on the wave fields), which is a function of the material, the waveguide design (mode localization), and the realistically feasible pulse energy.

In nanophotonic circuits, the first neuromorphic devices with an area of about $(100\ \mu\mathrm{m})^2$ per building block (tunable beam splitter, in that case) have been demonstrated [15], but future engineering should be able to scale down to around $(10\ \mu\mathrm{m})^2$, as suggested by previous developments (e.g., Ref. [59]). Three-dimensional integration may make it possible to realize large networks that are even more compact [60].

We can also provide an estimate of the minimum feasible size of the nonlinear elements, based on the size of current all-optical switches. For example, a nonlinear microring resonator with a diameter of 70 μm was demonstrated to behave as an optical switch in the few-mW level or less, with low dissipation [58]. With respect to the footprint of the components implementing the decay step and the time-reversal operation, we observe that their area scales only linearly with the number of degrees of freedom. Hence, their footprint is not the dominant contribution in the large-scale (many-layer) limit. Additionally, as explained above, phase-conjugating mirrors can be implemented by means of three- or four-wave mixing processes, for example, inside a microring resonator [61]. Therefore, the size of the phase-conjugating mirror could be on the same order as a layer of nonlinearities. A similar argument applies to the decay step, which requires similar physical components.

With respect to the relevant timescales of a self-learning machine based on integrated photonics, we can also provide some rough estimates. The duration of a training step has four contributions: the forward pass, the time-reversal operation, the error injection, the backward pass, and the decay step. The timescale of the forward pass (which has the same duration as the backward pass) will be proportional to the number of layers. Each layer would have a depth of $O(1)$ beam splitters and nonlinear resonators. For beam splitters with a size of 100 μm, the latency would be on the order of 10 ps. In the case of the nonlinear microresonators, the relevant timescale is roughly given by the inverse of their maximum operating bandwidth. For example, nonlinear microring resonators can operate with 100-ps pulses [62].

As we argued above for the footprint, the duration of the phase conjugation, decay step, and error injection will not be the main contribution in the large-scale limit. Still, we can give some rough estimates. In the case of the time reversal, as one example, optical phase conjugation has been demonstrated experimentally by means of a silicon waveguide [63] having a latency of around 1 ns. However, it is reasonable to expect much faster phase-conjugating elements, operating in the 100-ps range, based on nonlinear microring resonators [62]. With respect to the decay step, the limiting factor would probably be the step of parametric amplification. Since parametric amplification can be achieved by an adequate pumping of a three- or four-wave mixing process, we believe that the decay-step duration could be engineered to be on the same order as that of the phase-conjugating mirror. Finally, the error injection corresponds to the injection of a signal and a phase-shift element. Hence, the timescale of the error injection will be comparable to that of a single linear layer, which was estimated to be on the order of 10 ps.

Like for any platform suitable for HEB, dissipation needs to be kept low; i.e., one needs to limit the attenuation of the optical fields. Substantial progress has been made in the area of integrated photonics in this regard, by careful materials selection, fabrication processes, and surface treatment as well as geometrical design. This has led to record achievements like the recently reported [64] ultra-low-loss cavity in SiN with an optical quality factor of 4×10^6, leading to a sub-MHz optical decay rate and relative losses of less than 10^{-4} on the 100-ps timescale. More generally speaking, we point out that Hamiltonian echo backpropagation in optical devices will profit from the current rapid push towards optical quantum computing since this involves the development of components with extremely low loss and strong nonlinearities, which are exactly the ingredients needed for our approach as well. Moreover, besides these ongoing improvements, attenuation could be compensated with coherent amplification (as we have mentioned before). In this way, one could place layers of amplifiers inside the integrated circuit, in such a way that the optical field is retained. Alternatively, the photonic device components could themselves be built as an amplifier (e.g., erbium-doped fibers) with a properly calibrated gain to compensate the loss. Similar to all the other aspects discussed above, further research will be needed, but the toolbox of nonlinear optics and integrated photonics is by now advanced and versatile enough for this challenge.

Another challenge in the case of optical implementations is the long-term storage of the learning degrees of freedom. In our approach, during each pass, the learning field, besides undergoing time reversals and the decay step, will have to be slightly amplified to counteract the small unavoidable losses. In this way, it is continuously kept alive throughout both the training and inference phases. However, eventually, one will want to store it somehow, in order to keep persistence for longer times (indefinitely, instead of seconds or minutes). This could be done by performing a homodyne measurement on the learning field pulses and storing the amplitude and phase on a digital computer. A more physical approach would involve holography. There are setups that have been proposed in the past [8] that make use of holograms as a means to store the learning degrees of freedom of a photonic learning machine. In this way, very high storage densities could be attained [65]. Photorefractive holograms, e.g., based on lithium niobate, can be persistent for very extended periods, up to years.

Architectures beyond layered feedforward networks are also realizable in integrated-photonics platforms. The basic ingredients of a recurrent neural network, i.e., a network with memory, can be implemented. One possibility would be to inject the inputs corresponding to a discrete time sequence as a train of wave packets, propagating along a waveguide. Assume that the waveguide is coupled to a ring resonator. If the pulse duration is much shorter than the ring's roundtrip travel time and the distance between pulses matches this time, then the subsequent pulses will overlap in the ring. When nonlinear interactions are present inside the ring, this is a way to obtain recurrent dynamics, where the output depends on the whole input sequence. Multiple such building blocks can be combined sequentially or in parallel, although we do not attempt to work out a full working device here. Training via HEB is possible even in this scenario, by time reversing the whole train of pulses emerging out of the device.

Finally, besides integrated photonics, another interesting platform for optical HEB is possible. For example, the architecture proposed by Skinner *et al.* [11] combines free propagation inside a crystal and a spatially homogeneous Kerr nonlinearity. Another alternative may be so-called optical Ising machines. Ising machines have received a lot of attention in recent years [66–68]. An Ising machine is an optical device that simulates the Ising Hamiltonian. One common implementation is based on a time-multiplexed train of pulses inside a fiber loop with a degenerate optical parametric oscillator (OPO). Because of the interaction with the OPO, the pulses have two degenerate stable states that can be treated as spinlike variables. A linear device in the fiber loop couples the pulses in a way that is reminiscent of an Ising Hamiltonian. In this way, an artificial lattice can be implemented with just a single fiber loop, e.g., demonstrating networks with 2000 degrees of freedom [67]. Although more research would be needed to work out a self-learning Ising machine, we can already see that such a setup has all the elements needed for our proposal: It has very complex nonlinear dynamics, it is possible to engineer a degenerate ground state for Θ, and it can be time reversed. As opposed to a standard photonic circuit, the number of nonlinear physical components can be strongly reduced since they are reused by the train of pulses.

C. Superconducting circuits

Beyond optical circuits, superconducting circuits are another possible platform. In a somewhat similar fashion to photonic circuits, a typical superconducting circuit can be described as a 2D arrangement of coupled superconducting oscillators coupled by waveguides [69]. In fact, we can understand such superconducting oscillators as cavities that operate in the microwave regime. However, unlike photonic circuits, these superconducting cavities are made by coupling superconducting inductors, capacitors, and Josephson junctions.

In this scenario, our learning and evaluation fields are now microwave fields in the superconducting cavities. By using Josephson junctions, it is possible to engineer such superconducting cavities to induce strong nondissipative nonlinear interactions. Here, we consider that the circuit operates in the classical limit.

For the purpose of this paper, it suffices to understand that it is possible to make lattices of coupled nonlinear superconducting cavities. In the end, this platform is conceptually not very different from the optical platforms described above, except for the fact that the Ψ and Θ fields are now in the microwave spectrum, and the nonlinear elements are Josephson junctions. In Appendix C, we briefly describe some of the design possibilities for constructing a self-learning machine based on this platform, reviewing some aspects of superconducting circuits in a very compact manner. More importantly, in Appendix D, we consider a concrete setup (which can be understood without detailed knowledge of superconducting circuits) and show some numerical results on the simulation of a self-learning device based on an array of nonlinear cavities, which could be implemented by means of a superconducting circuit. This example also illustrates the case of a self-learning device in which the learning field is always localized entirely inside the nonlinear core.

The potential of superconducting circuits for fast and low-power classical computation has been recognized early on, which led to the development of so-called rapid single flux quantum logic in the 1990s. Although the significant advances in more conventional electronics have resigned this to the status of a niche application, it has more recently been considered as fast control electronics for quantum processors. In addition, classical superconducting circuits are now being considered for realizing forms of artificial spiking neuron hardware [22], e.g., based on superconducting nanowires [70], although these are designed to work in a dissipative regime. At the same time, the past 20 years have seen impressive advances in superconducting circuits operating in the quantum regime for quantum computation and simulation, building especially on superconducting microwave cavities that are made nonlinear by interaction with Josephson junction-based elements. Such applications have driven the significant minimization of losses, which is important for our purposes. Taken together, this means there now exists a very sophisticated toolbox for superconducting circuits that can be exploited for Hamiltonian echo backpropagation.

Regarding their footprint, microwave cavities range from cm scale (for 3D cavities and transmission line resonators) down to $(100\ \mu\mathrm{m})^2$ in a lumped-element LC-circuit implementation (the Josephson elements are much smaller and can be neglected here). Even though overall we do not expect the same integration density as for optics, such circuits offer the attractive possibility to be integrated on chip with quantum information processing devices, making use of the same technology. This would provide online self-learning on real devices with negligible latency and no signal distortion by propagation on cables to the outside.

Coherence times in qubits now reach several hundred microseconds, in cavities even up to 10 ms, while strong nonlinear effects at the single-photon level can be obtained during times of around 500 ns or less. We intend to perform operations in the classical regime, at much higher photon numbers n, where nonlinearities are enhanced, e.g., reducing the time needed for acquiring a Kerr-effect phase shift of π by a factor $1/n$. Thus, in a conservative estimate, dissipation per nonlinear pi-phase shift could be as low as $10^{-2}/n$, with n easily reaching the order of 10^6 or higher. This produces, to an extremely good approximation, the time-reversal-invariant Hamiltonian dynamics we are striving for, turning such systems into a very promising platform.

D. Other approaches

Approaches not based on optics or superconducting circuits are also possible, though we will only mention them briefly, and their detailed design would be a challenge for future research.

Clouds of cold atoms or molecules, possibly trapped in optical lattices, offer the possibility of storing information in the density distribution or the molecular orientation, making longer-term storage of the learning field Θ, in principle, less challenging than in the case of optical cavities. The interaction of an optical field Ψ with the atomic or molecular cloud would also provide strong optical nonlinear self-interactions, as has been demonstrated with Rydberg atom clouds. Alternatively, the fields themselves could be realized as macroscopically populated, classical matter waves. This directly leads to the nonlinear Schrödinger equation, which we showed to be useful for HEB in our numerical examples. Four-wave mixing and phase conjugation with matter waves have been demonstrated in Ref. [46].

Likewise, spin waves share all the required ingredients for a SL machine. We must remark that time-reversal symmetry must not be broken, which means that spin-spin interactions may be acceptable but not (for example) light-magnon interactions via the Faraday effect. In the case of spin waves, there are already experimental demonstrations of strong nonlinear interactions that give rise to sigmoid-like activation functions [71], and phase conjugation by nonlinear interactions has been demonstrated [48].

IX. CONCLUSIONS

In this work, we have introduced a general method, Hamiltonian echo backpropagation, to train SL machines based on Hamiltonian physical systems. For any physical device that meets some general assumptions, our method is guaranteed to realize gradient descent optimization of a cost function. The HEB procedure updates the learning parameters autonomously and *in situ*, leveraging all the advantages of a neuromorphic device. Whether the physical device could achieve a good performance on a particular task will depend on its size and its expressive power, but our method can always find a local minimum of the cost function via physical dynamics. Our numerical simulations illustrate how it can be applied both to simple physical systems and to other, more sophisticated architectures. In fact, we showed an example of its application in the case of a photonic convolutional neural network used for image recognition. Although there are many previous proposals and even experimental implementations of photonic learning machines, most of their potential advantages cannot be leveraged without a training method that is fully autonomous. We believe that when HEB is directly applied to such photonic devices, this will greatly enhance their capabilities.

Going beyond the realm of photonics, there are many other promising physical platforms for machine-learning hardware, as we have mentioned. We believe that our method would be especially helpful when there is no control over the internal dynamics of the device. Indeed, since none of the steps in HEB depends on the underlying Hamiltonian, one can use the physical device as a black box. In this way, HEB is insensitive to device variability brought about by fabrication imperfections, and it can also cope well with noise (as we have shown in one of our numerical examples), which is particularly important in low-power applications with a suppressed signal-to-noise ratio. Our method not only allows us to consider various physical platforms, it will also make it easier to explore new architectures. As opposed to other previous proposals, HEB does not rely on a sequential multilayered structure, which means that it can be used to train new architectures going beyond the structure of a standard ANN.

The code used for the numerical examples is available on Github [72].

ACKNOWLEDGMENTS

We thank Dirk Englund and Stefan Krastanov for insightful discussions on some aspects of experimental implementations in the domain of integrated photonics.

APPENDIX A: EXAMPLE: XOR FUNCTION

The first example of a SL machine that we show has a particularly simple architecture. The evaluation dynamical variable is a nonlinear wave field propagating in a 2D square lattice. We consider a tight-binding model with hard-wall boundary conditions in which every node of the lattice is coupled to its nearest neighbors. At each site of the lattice, there is a nonlinear attractive Kerr self-interaction. The learning dynamical variable is given by another field that does not propagate. We assume that the learning field and the evaluation field interact through a term in the Hamiltonian given by $\theta\Psi^*\Psi$, which is the simplest term that can produce local phase shifts.

During the time-reversible steps, the equations of motion for both fields are given by

$$i\dot{\Psi} = \frac{\beta}{2}\nabla^2\Psi + (\chi\theta + g|\Psi|^2)\Psi, \tag{A1}$$

$$i\dot{\Theta} = i\Omega\pi_\theta + \chi|\Psi|^2. \tag{A2}$$

The hard-wall boundary conditions are given by $\Psi(x=0,z) = \Psi(x=W,z) = \Psi(x,z=0) = \Psi(x,z=L) = 0$. During the decay step, the evolution of θ is given by $\dot{\theta} = \Omega\pi_\theta$, $\dot{\pi}_\theta = -\Gamma\pi_\theta$.

The goal of this example is to learn a simple logical function. In particular, we choose the XOR function, $b = a_1 \oplus a_2$, since it is well known that it cannot be learned by a single-layer perceptron. For each logical input, (a_0, a_1), we provide a physical input as a superposition of wave packets, $\Psi(\mathbf{x}, -T) = f^0(\mathbf{x}, a_0) + f^1(\mathbf{x}, a_1)$. Each wave packet, $f^s(\mathbf{x}, a_s)$, is a Gaussian wave packet centered around $((2s+1)W/8 + a_s W/2, l_{\text{in}}), s \in \{0, 1\}$ whose standard deviation is given by σ. In addition, each wave packet has an initial positive momentum in the propagation direction. In this way, each input boolean variable a_s is encoded in the position of the center of the corresponding wave packet.

We train such a self-learning machine to learn the XOR function using Hamiltonian echo backpropagation. The cost function is given by

$$C := \int dxdz\phi(\mathbf{x})|\Psi(\mathbf{x}, 0) - \Psi_{\text{tar}}(\mathbf{x}, b)|^2, \tag{A3}$$

where $\phi(\mathbf{x}, b) = (1/2\pi\sigma_\phi^2). \sum_{j=0}^{1} e^{-(\mathbf{x}-\mathbf{x}_j^{\text{out}})^2/(2\sigma_\phi^2)}$ is the sum of two Gaussian functions centered around $\mathbf{x}_0^{\text{out}} = (w, 0.8L)$ and $\mathbf{x}_1^{\text{out}} = (3w, 0.8L)$, respectively. The target field $\Psi_{\text{tar}}(\mathbf{x}, b)$ is also a Gaussian function centered around $\mathbf{x}_b^{\text{out}}$. Therefore, our cost function is a version of the MSE weighted by $\phi(\mathbf{x})$. In this way, the SL machine tries to fit $\Psi(\mathbf{x}, 0)$ to $\Psi_{\text{tar}}(\mathbf{x})$ inside the peaks of $\phi(\mathbf{x})$, where the output is measured. We finally evaluate the accuracy using the following prescription: If $|\Psi(\mathbf{x}_0^{\text{out}}, 0)| > |\Psi(\mathbf{x}_1^{\text{out}}, 0)|$, the logical output is 0; otherwise it is 1.

In order to solve numerically the equations of motion for the time-reversible steps, we use a split-step method. In particular, we use the symmetrized Fourier split-step method, which takes half a time step using the linear operator in momentum space, then takes a full time step with the nonlinear operator in position space, and then a second half time step, again with the linear operator in momentum space. In order to satisfy hard-wall boundary conditions, the discrete sine transform is used. We approximate the effect of the cost function perturbation by a single, small Euler step at $t = 0$ (this is a reasonable approximation when ϵ is small enough). For the decay step, we assume that we are in the regime in which Eq. (7) applies. Therefore, for each HEB step, we simulate a forward pass and a backward pass, and then we update the parameters according to Eq. (7).

We use a square lattice with 50×150 sites. The nonlinear strength is chosen as $g = 10^2 I^{-1} T^{-1}$, where $I := (2\sigma)^{-2} \int dxdz|\Psi|^2$ is a measure of the peak intensity of the input field. In this way, the accumulated phase shift due to the nonlinear self-interaction is much larger than 2π. The dispersion coefficient has to be large enough to prevent a self-focusing collapse. This implies that $(\beta/2)\nabla^2\Psi$ should be at least on the same order as $g|\Psi|^2\Psi$. In our numerical simulation, we set $\beta = 10^2[(2\sigma)^2/T]$. With this choice of parameters, one can check in the numerical simulations that the extent of the wave packets remains on the order of $(2\sigma)^2$ during the whole forward pass. We choose the initial momentum of the wave packets as $k_z = \frac{3}{4}(L/T\beta)$, so they propagate a distance $3L/4$ during the forward pass.

The parameter Ω determines the rate of change of θ. In order to have results that can be more easily interpreted, we work in the regime in which θ remains almost constant during a single forward pass. Hence, we choose $\Omega = 10^{-2}T$. Next, we must choose Γ so that

$\frac{\Omega}{\Gamma} \gg [\partial C/\partial\pi_\theta(-T)][\partial C/\partial\theta(-T)]^{-1}$, if we want to work in the regime in which Eq. (7) applies. Since the cost function depends on π_θ only indirectly, through the evolution of θ, one can estimate that $\partial C/\partial\pi_\theta(-T)$ is on the order of $\Omega T[\partial C/\partial\theta(-T)]$. Hence, we must make sure that $1/\Gamma \gg T$. In particular, we set $\Gamma = 10^{-2}T$. After one training step, the update in θ is of the order of $\epsilon(\Omega/\Gamma)\chi I$. Hence, the change in the phase shift in Ψ due to the interaction with θ is on the order of $\epsilon(\Omega/\Gamma)\chi^2 IT$. On the one hand, this update must be small enough so that the stochastic gradient descent works. On the other hand, it should not be so small that the training progress is too slow. We choose $\chi = 10^2(\Omega/\Gamma\epsilon IT)^{1/2}$. In our numerical simulations, we set $I = 1$, $T = 1$, $L = 3$, $\sigma = 0.1$.

APPENDIX B: EXAMPLE: IMAGE RECOGNITION

As an illustrative example of a SL machine, we consider a device based on a photonic neural network proposed by Ong *et al.* [56], thereby showing how an existing device design can profit from our self-learning procedure. It is composed of three convolutional layers followed by a dense fully connected layer. The nonlinearity is provided by a $\chi^{(2)}$ optical nonlinearity, as described in Eq. (F1). Both the convolutional layers and the fully connected layer can be implemented by a sequence of discrete Fourier transforms (DFTs) and learnable phase shifts. In Ong *et al.*, the activation function is a complex version of the ReLU function. Instead, we choose the activation function given by a $\chi^{(2)}$ nonlinear interaction, which, in principle, can be implemented in a physical setup. As a cost function, we choose the mean square error on the intensity.

The processing inside the photonic convolutional neural network that we consider is composed of a sequence of operations acting on a 2D square grid of $N \times N$ modes. In addition to the evaluation modes, there are learning and ancillary modes that take part in the sequential operations and also have to be time reversed.

Any convolution (with periodic boundary conditions) is diagonalized by the DFT. Therefore, the weight matrix in a convolutional layer can be performed by applying a DFT, then a diagonal matrix, and finally the inverse DFT. In the case of a unitary convolution, the diagonal matrix is of the form $D_{jj} = e^{i\theta_j}$ (i.e., a phase mask). All three operations are unitary operations that can be implemented with a lossless photonic circuit. The DFT matrix is of course fixed; all the learning parameters θ_j are contained in the phase mask.

The last two layers of the network are fully connected, and they should be able to express any arbitrary complex matrix. The implementation of these fully connected layers was not described in Ref. [56], so we provide a possible version here. We use a decomposition based on the singular value decomposition. In the singular value decomposition, any complex matrix can be cast as a product of the form USV, where U, V are unitary matrices and S is a diagonal matrix. The unitary matrices can always be expressed as a product of the form $U = D_0 \prod FD_i$, where F are DFT matrices and D_i are diagonal unitary matrices (phase masks) [73]. Finally, the singular value matrix S can be implemented by means of a linear interaction with an ancilla, provided that the singular values are bounded by $S_{ii} \leq 1$. This is achieved by coupling every relevant mode to an ancilla mode by means of a general SU(2) linear transformation, when the ancilla is initially zero. Any such transformation can be achieved by means of the product of a 50-50 beam splitter, a phase shift, and another 50-50 beam splitter. Since an overall scaling factor in the weight matrix is unimportant, it is sufficient that our implementation can produce any complex matrix with $|S_{ii}| \leq 1$.

All the trainable parameters are contained in the phase shifts, which, in this example, are realized by a local interaction Hamiltonian given by $\theta\Psi^*\Psi$. Just like in the first example, we assume that the machine operates in the limit in which the dissipation timescale is much longer than the duration of a forward pass.

For the numerical implementation, we solved the evolution of the evaluation and learning fields in a sequential manner. There are five kinds of operations inside each forward pass: (1) DFTs, (2) the interaction of Ψ and Θ, (3) the interaction of Ψ and Ψ_{anc} (the ancilla in the dense layers), (4) the nonlinear activation function, and (5) the cost function perturbation. Each of these steps represents a different layer in the photonic device, and therefore they can be solved separately. The 2D DFT was performed by means of a fast Fourier transform algorithm. Step (2) corresponds to the interaction described by the Hamiltonian $H = \chi_l\theta\Psi^*\Psi$. The equations of motion for this step can easily be solved analytically. Therefore, we can calculate the overall effect of step (2) by means of a simple formula. In the same way, step (3) is described by the Hamiltonian $H = J(\Psi^*\Psi_{\mathrm{anc}} + \Psi\Psi^*_{\mathrm{anc}})$, which can also be solved analytically. Step (4) is integrable too, with the solution given by Eq. (F1). Step (5) is a small perturbation that can be well approximated by a single Runge-Kutta step. For the backward pass, the same operations are performed but starting with the phase conjugation of the output of the forward pass. Of course, one has to time reverse not only the evaluation field but also all the ancillary and learning variables.

The photonic convolutional neural network that we consider has three convolutional layers and two fully connected layers. The convolutional layers are somewhat unconventional: They only have one filter, but it is the same size as the output. As we already described, this is accomplished by an alternation of a DFT, a phase mask, and another DFT. The first DFT acts on the full input image with N^2 modes, while the phase mask and the second DFT act only on the innermost M^2 modes. The rest of the modes are considered to be ancillary modes. In particular, the first layer has $(N, M) = (31, 31)$, the second has $(N, M) = (31, 21)$, and the third has $(N, M) = (21, 17)$. The sizes of the input and output of the fully connected layers are (17,9) and (9,4), respectively. The input of the network is a rescaled version of the corresponding image from MNIST. The intensity of the field at each input mode goes from 0 to $I_{\max}$ (which has units of the square root of an action). In the numerical simulation, we set $I_{\max} = 1$. In addition, we set all the elementary steps (activation functions, interaction with θ, etc.) to have duration $T_s = 1$. At the convolutional and dense layers, we have two different values of the nonlinear strength, which are given by χ_1 and χ_2, respectively. From Eq. (F1), one can deduce that both χ_1 and χ_2 have to be chosen so that $\chi_i|\Psi|T_s \approx 1$ for the dynamics to be strongly nonlinear. The nonlinear strength is chosen as described in Fig. 7. The coupling to the ancilla is given by $J = 10^1 T_s^{-1}$ since we want to ensure that the mixing of the ancilla and the relevant modes is on the order of 1.

With regard to the decay step, it is assumed that we are in the regime in which Eq. (7) applies. Therefore, for each HEB step, we simulate a forward pass and a backward pass, and then we update the parameters according to Eq. (7). The coefficient of the cost function perturbation is taken to be $\epsilon = 10^{-2}$, and we set $\Omega = 10^{-1} T^{-1}$ so that the update of θ during a single forward pass can be neglected. We use a damping coefficient given by $\Gamma = 10^{-1} T^{-1}$ so that the learning rate is given by $\eta = 10^{-2} T$. Finally, by a similar argument to the learning of the XOR function, we set $\chi_l = 10^1 (\Omega/\Gamma\epsilon I_{\max} T)^{1/2}$ (see Appendix A).

APPENDIX C: IMPLEMENTATION IN SUPERCONDUCTING MICROWAVE CIRCUITS

We briefly describe some of the design possibilities for learning machines based on superconducting circuits. We also point out that understanding of the technical details that we briefly touch upon here is not needed to appreciate the simulations presented in the next appendix.

We first describe the physically simplest implementation, although this is not the one we simulate in the next appendix. This implementation would consist of a Josephson junction array, which is essentially a lattice of connected capacitors, inductors, and Josephson junctions. There is considerable design freedom since one can choose the connectivity via the electronic wiring topology and the properties of the individual elements by parameters such as the junction area, etc. Such an array implements a discrete nonlinear metamaterial. A classical Hamiltonian is obtained [69] by introducing node fluxes ϕ_n via the time integrals of voltages across the circuit elements and node charges q_n via the time integrals of currents. These form canonically conjugate variables, with q_n taking the role of momenta, such that the equations of motion for the circuit follow as $\dot{\phi}_n = \partial H/\partial q_n$ and $\dot{q}_n = -\partial H/\partial \phi_n$. The node fluxes are proportional to the superconducting condensate phase at each location in the circuit, and their difference across a Josephson junction indicates the phase difference that determines the energy. As an example, in a simple 1D array, one can produce the Hamiltonian of a discrete sine-Gordon model, $H = \sum_n (2C)^{-1} q_n^2 - E_J \cos(\varphi_{n+1} - \varphi_n)$, with $\varphi_n = (2e/\hbar)\phi_n$. The capacitance C and the Josephson energy E_J characterize the circuit elements. Despite the appearance of Planck's quantum $\hbar$ in this formula, one can operate in the classical regime by ensuring $E_J \gg e^2/C$.

The type of setup we consider from now on, and in the next appendix, however, is closer in spirit to photonic circuits built from coupled optical resonators. Indeed, it is possible to build superconducting setups that employ arrays of coupled microwave modes (produced using transmission line resonators, 3D cavities, or lumped-element LC circuits), which are modeled as LC oscillators in the Hamiltonian and which can be made nonlinear by the insertion of Josephson junctions. Expanding the cos term in the field amplitude yields a description of coupled modes analogous to nonlinear quantum optics (with Kerr terms, etc.). Since the Hamiltonian is time-reversal invariant in the absence of external magnetic fields, this platform fulfills all the requirements for HEB, provided a suitable mechanism for the echo and decay steps is implemented, following our prescriptions in the main text in Sec. VI.

In a simple circuit of this type, the main conceptual difference in contrast to nonlinear optical platforms is the absence of a natural propagation direction, which makes the circuit a fully recurrent device in the language of neural networks (no layers, no feedforward processing). This could be circumvented by employing transmission lines as waveguides, coupled to nonlinear microwave cavities, resulting in a setup whose mathematical description is completely equivalent to that presented earlier for the nonlinear integrated photonics platform, only operating in a different frequency regime (and with different physical mechanisms for tunability, as well as considerably stronger nonlinearities). Nonlinear wave dynamics, and applications like amplifiers, have been experimentally demonstrated in that type of setup.

However, it would also be interesting to explore physical learning machines beyond the standard feedforward architecture. In fact, the idea of using nonlinear physical dynamics as an analog implementation of a recurrent neural network was already considered by Hughes *et al.* [74]. Moreover, Tureci *et al.* [75] already studied the use of a superconducting circuit as an analog implementation of a reservoir computer with recurrent architecture. We take this opportunity to demonstrate the generality of our approach and deliberately consider a recurrent circuit (see next appendix).

We argue that there are promising applications for a learning device like the one we describe in the following appendix. For example, a similar learning device was proposed in a recent paper as a tool for efficient quantum state measurement [75]. In their proposal, which is based on the ideas of reservoir computing, the physical degrees of freedom are not trained. Instead, they require external postprocessing that has to be trained. Our scheme could potentially allow us to perform training *in situ* in a more efficient way.

APPENDIX D: NUMERICAL EXAMPLE: CAVITY ARRAY

In the main text, we have described two numerical examples, involving propagation of nonlinear wave fields that could be obtained either in a bulk medium or in integrated photonic circuits. Here, we present numerical simulations for a third platform, namely, cavity arrays. This is inspired by the superconducting microwave circuits (see previous appendix and discussion in main text), where it could be implemented.

In contrast to the examples considered in the main text, we note that this will require a mechanism to time revert the fields inside all the cavities, and it thus denies the treatment of the self-learning machine as an arbitrary “black box,” which would be possible if time reversal and decay took place outside the device, as in the optical case. Nevertheless, given the great amount of engineering control available nowadays in these platforms, it is not a major constraint. Imagine a setup of coupled microwave cavities that have been made nonlinear by insertion of Josephson junctions and that can be driven parametrically by externally applied microwave drives (a standard scheme in manipulating such cavities, employed routinely for quantum information processing). This makes it possible to implement the crucial steps of our protocol: time reversal, decay of the learning field, and injection of the error signal (as well as refreshing the input after one learning sample has been processed).

Consider the simplest kind of scenario, where an array of such nonlinear superconducting microwave cavities is arranged in a square lattice, with each cavity identical in the simplest case (though static disorder is not a problem, unless it affects the time reversal or the decay step). In a coupled-mode description, we have the Hamiltonian

$$H = -\sum_{m,n} J_{mn} a_m^* a_n + g \sum_n (a_n^* a_n)^2.$$

We distinguish between Ψ cavities and Θ cavities, but the only difference consists in the operations we apply to them: At the beginning, the field amplitude inside the Ψ cavities is quenched to zero, and then the training input is stored inside them. Afterwards, the evolution according to H takes place. After a time interval T, time reversal is applied to all cavities, using the kind of parametric setup discussed elsewhere in this paper. The error signal is injected, and after a further time T, the final decay step takes place.

We have numerically investigated this example, and again we trained it for the XOR training case, where only two input modes and two output modes are needed. The setup is explained in Fig. 9(a).

The setup consists of N^2 cavities that encode Θ and four additional cavities that encode Ψ (two input modes, two output modes). The cavities corresponding to Θ are placed in an $N \times N$ square array. The cavities corresponding to Ψ are placed in the corners. We set $N = 4$. The Hamiltonian is given by a linear term that couples the neighboring modes and a homogeneous local Kerr nonlinearity. To be precise, the linear term corresponds to a square tight-binding lattice. We work in a regime in which the ratio between the coupling strength of the tight-binding Hamiltonian and gI is very large: $J/(gI) \approx 10$, where I is the average intensity of the Φ field. We empirically observe that in this regime, the self-learning machine can train successfully. On the other hand, much larger nonlinearities often seem to lead to convergence issues.

We would not want the output value to be very sensitive to the specific timing of the readout of the output. For this reason, we also set $K/J \approx 10$. In this way, the fast oscillations of the modes of the lattice are averaged out in the output.

Because of the Kerr nonlinearity, the modes of the lattice will interact with each other via four-wave mixing processes. We ensure that the interaction time is long enough that different modes of the lattice can completely mix with each other. In other words, we make sure that $gIT \gg 2\pi$.

We simulate the dynamics by means of the Fourier split-step method. In Fig. 9(b), we can observe that there is a finite range in the values of the input intensity for which the training is optimal (notice the logarithmic scale). In Fig. 9(c), we study the effect of fabrication imperfections. In particular, we consider that the Hamiltonian of the device deviates from the ideal Hamiltonian by a random term that describes the fabrication imperfections. The random term in the Hamiltonian corresponds to random frequency shifts of each cavity mode in the lattice. The frequency shifts are drawn from a Gaussian distribution centered around 0 and with deviation given by σ_Δ. Crucially, HEB does not need knowledge of these random imperfections in order to realize backpropagation. We observe that the self-learning machine can minimize the cost function even for relatively large values of σ_Δ, on the order of $0.3\pi T$. For very large values of σ_Δ, even though it still performs gradient descent, the device seems to converge to poor local minima. Figures 9(d)–9(f) show the evolution of the output values during training, the evolution of the output during a single forward pass, and the effect of noise in the input.

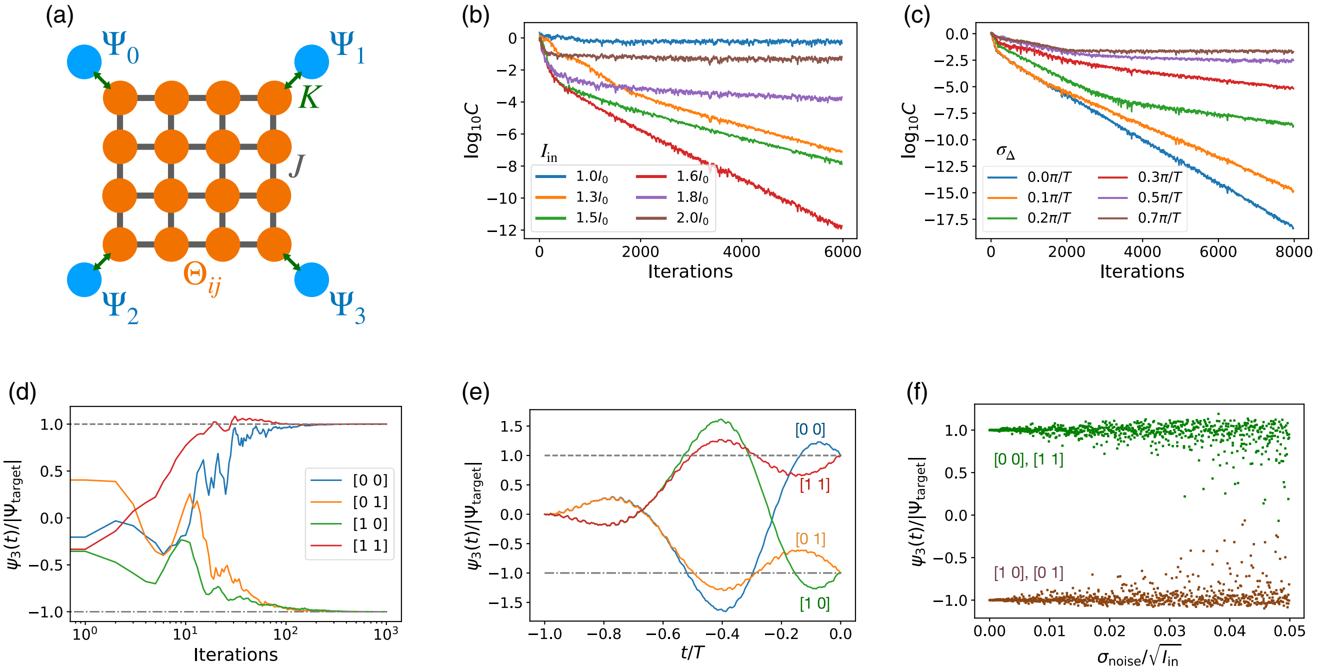

FIG. 9. Self-learning machine based on an array of nonlinear cavities (simulations). (a) Schematics of the setup. The cavities that encode the learning degrees of freedom are placed on a 2D square lattice. Each nonlinear Θ_{ij} cavity is coupled to its nearest neighbors, with a coupling strength given by J. The cavities that represent the evaluation degrees of freedom, Ψ_i, are coupled to the Θ_{ij} cavities in the corners of the lattice, and the coupling strength is given by K. All cavities have an identical Kerr nonlinearity whose strength is given by g. The input is encoded in the initial values of $(\psi_0(-T), \psi_1(-T))$, while the output is given by $\psi_3(0)$. (b) Training error evolution for different values of the intensity of the input. We choose a reference level of intensity I_0 that fulfills $gI_0T = 3$. (c) Robustness of the SL machine to fabrication imperfections. In particular, we consider that each cavity has a random frequency shift whose distribution is given by $\mathcal{N}(0, \sigma_\Delta)$. Note that, since the random shifts are on the order of $0.1\pi/T - 1\pi/T$, their effect on the output value is very significant. The HEB method can successfully implement gradient descent in the imperfect machine (since it is Hamiltonian independent). However, for larger values of σ_Δ, the training process seems to converge to worse performance. Nevertheless, this result shows robustness to significant fabrication imperfections. (d) Evolution of the output as the training progresses (with an input intensity given by $I_{\text{in}} = 1.6I_0$). Each curve represents each different input. (e) Time evolution of the field amplitude in the output cavity during a single forward pass, once the SL machine has been trained. (f) Outputs obtained when the input is noisy. We generate noisy inputs by adding Gaussian noise to the ideal input. After feeding each input to the SL machine, a noisy output is obtained. Here, we plot the resulting outputs vs the standard deviation of the Gaussian noise. We can observe that the output remains highly concentrated around the target value when σ_{noise} is in the range of 0%–3% of the input amplitude.

APPENDIX E: INGREDIENTS OF THE HAMILTONIAN: LINEAR AND NONLINEAR PARTS

The dynamics of Ψ and Θ is given by the Hamiltonian H_{SL}. If one wants to use a SL machine for any useful task, the dynamics of Ψ must be complex and nonlinear. The expressive power of ANNs can only be matched if the equations of motion for Ψ incorporate nonlinear interactions, as well as interactions with the learning parameters encoded in Θ. However, the input-output relation must be deterministic and robust to small amounts of noise. These considerations should guide the choice of Hamiltonians.

As we discuss later in this section, there is a considerable amount of theoretical and experimental work about the time reversal of wave fields, especially in nonlinear optics. For this reason, here and henceforth, we concentrate on approaches based on nonlinear wave fields. Here, $\Psi(\mathbf{x})$ and $\Theta(\mathbf{x})$ are complex fields defined on a continuous space or in a discrete lattice. Although we make extensive use of a notation that is reminiscent of optical fields, we must remember that, actually, the setting is completely general (the real and imaginary parts of the field are the two conjugate coordinates of a completely general Hamiltonian degree of freedom). However, the fact that we have now assumed $\Psi(\mathbf{x})$ and $\Theta(\mathbf{x})$ to be fields allows us to make use of the notion of locality.

In a SL machine based on wave fields, H_Ψ would be composed of two parts: a free-field Hamiltonian H_0 and a second term H_{NL} containing local self-interactions. We can define a free-field Hamiltonian as any quadratic function of Ψ, Ψ^* and its time and space derivatives. More restrictively, we can ask that H_0 is invariant with respect to a global phase shift of Ψ, so the total intensity $\int d^D x|\Psi|^2$ is a conserved charge. In that case, H_0 can only contain second-order terms that are bilinear in Ψ, Ψ^*.

To produce nonlinear dynamics, we introduce the self-interaction term H_{NL}. Often, this will be a local term. Typical examples of self-interaction Hamiltonians are the familiar Kerr nonlinearity $g|\Psi|^4$ and a Josephson nonlinearity $\chi \cos\psi$, which can become relevant for superconducting circuit quantum electrodynamics.

The free-field Hamiltonian, together with a self-interaction, is enough to obtain very complicated dynamics. However, if we want to train the SL machine, we still need to introduce some dependence on the learning parameters through the interaction Hamiltonian H_{int}. In analogy with ANNs, we may ask that the interaction produces dynamics linear in Ψ, although this is not a requirement. Moreover, H_{int} will also be local in most realistic settings. Therefore, the most reasonable choices for H_{int} are $g\Theta^*\Theta\Psi^*\Psi$ (in nonlinear optics, this is called cross-phase modulation) or $g(\Theta + \Theta^*)\Psi^*\Psi$ (the standard $\chi^{(2)}$ interaction in optics or, equivalently, an optomechanical interaction), or, in general, some interaction of the form $f(\Theta, \Theta^*)\Psi^*\Psi$. Alternatively, parametric interactions of the type $f(\Theta, \Theta^*)\Psi^*\Psi^* + \mathrm{c.c.}$ could also be considered, leading to a different kind of dynamics, although we will not discuss them here.

With respect to the strength of the nonlinear interactions, it should be large enough to attain a complex and highly expressive dynamics. However, some strongly nonlinear physical systems may exhibit chaotic behavior. A system with strong sensitivity to the initial conditions is undesirable because any small perturbation can be strongly amplified, making it impossible, in practice, to realize the time-reversed replica during the echo step. Ideally, in order to avoid strong sensitivity to the initial conditions while maximizing the expressivity of the device, the Lyapunov exponents should scale in such a way that the device operates near the onset of chaos but below it. Similarly, a recent paper showed numerical evidence that some neuromorphic devices achieve optimal performance in the edge of chaos [76]. Other recent related works have studied how transient chaos can improve the expressive power of neural networks [77–79].

The Hamiltonian terms we described could be homogeneous across space, like in the case of an optical field propagating inside a nonlinear crystal, or they could be implemented using discrete building blocks, as would be the case for a photonic or microwave circuit with nonlinear elements. It can be argued that the evolution of Ψ under the joint action of $H_0 + H_{\mathrm{int}} + H_{\mathrm{NL}}$ is very similar to the evaluation of an ANN (as argued in other works before; see, e.g., Refs. [8,11,74]). On the one hand, the effect of H_{NL} is to induce a nonlinear activation function. On the other hand, the part $H_0 + H_{\mathrm{int}}$ is quadratic in Ψ and can be thought of as implementing the trainable matrix of weights.

It is known that local controllable shifts, together with fixed unitary operations, are enough to realize any unitary transformation [80]. Of course, it needs to be noted that standard ANNs can have arbitrary weight matrices between layers, which seem to go beyond the unitary matrices available in the present setting. However, any arbitrary matrix can be embedded in a larger unitary matrix (with proper rescaling). For this reason, the structure of the linear part of the ANN can always be replicated in a SL machine. Alternatively, we may view a SL machine with unitary evolution as realizing a particular case of an invertible neural network.

APPENDIX F: PSEUDODISSIPATION VIA ANCILLARY MODES

As mentioned in an earlier section, pseudodissipation via ancillary modes can be a useful feature in a SL machine. We can imagine that in our SL machine, there are dynamical variables that are coupled to Ψ until some particular time, after which they evolve separately. Instead of explicitly switching off the couplings as a function of time, we might also have wave fields propagating out of the region where interactions are present. This can easily be arranged, e.g., in photonic circuits. The ancillary dynamical variables carry away part of the information, reconciling the contractive mapping with the overall time-reversal symmetry of the SL machine. We start with a high-dimensional input, but as time passes, the dimension of the interacting part of the system is increasingly reduced. In the end, the dimension of the output may be much smaller than the original input dimension. Importantly, in the echo step of HEB, the output and the ancilla must both be time reversed.

Pseudodissipation via ancillary modes can play an important role for the activation functions implemented via nonlinearities. As we already mentioned, the nonlinear evolution induced by the self-interaction Hamiltonian can be understood as an activation function. It is desirable to choose H_{NL} in such a way that the induced activation function is not oscillatory and has a bounded derivative. Ideally, a monotonic function would be the best option, reproducing standard ANN activation functions. In most previous approaches to physical learning with nonlinear optical devices, the nonlinearity is provided by a homogeneous Kerr self-interaction. The effect of a Kerr nonlinearity is to induce a phase shift proportional to the intensity of the field. Qualitatively, we can think of a Kerr medium as a deep neural network where the activation function is of the form $f(\Psi) = e^{ig|\Psi|^2}\Psi$. This is an oscillatory function, which is an undesirable property for an activation function in an ANN. What is worse, the derivative of the Kerr phase shift grows without bounds for increasing intensity. If we want to mimic the behavior of a sigmoid or ReLU activation function, we must incorporate ancillary degrees of freedom because all such standard activation functions are contractive. One possible choice is to use a $\chi^{(2)}$ nonlinearity and to couple an ancillary mode to the evaluation field: $H_{\mathrm{SI}} = \chi^{(2)}(\Psi_{\mathrm{anc}}^*\Psi^2 + \Psi_{\mathrm{anc}}\Psi^{*2})$.

When the ancillary field is initially $\Psi_{\mathrm{anc}} = 0$, the absolute value of $\Psi(t)$ as a function of the absolute value of the input is qualitatively similar to a sigmoid function (although it is not a monotonic function):

$$\Psi(t) = \Psi(0)\mathrm{sech}\big(2^{-1/2}\chi t|\Psi(0)|\big). \tag{F1}$$

Aside from $\chi^{(2)}$ nonlinearities, it is certainly also possible to produce an adequate activation function by a judicious combination of ancillary modes and a Kerr self-interaction Hamiltonian, or other kinds of self-interaction. Sigmoid-like thresholders based on the optical Kerr effect have previously been described [81], as well as bistable optical switches [82]. It is known that the Kerr interaction in combination with linear operations can be used to construct the Fredkin gate, and therefore it is universal for classical reversible computation [83].

Nevertheless, we should emphasize that nonoscillatory activation functions are not a requirement for HEB but a way to improve the expressivity of the device for those architectures that are more closely inspired on standard artificial neural networks. Just like some nonlinearities such as ReLU work better in standard ANNs, in a SL device, some nonlinear interactions have better results for some tasks. However, in general, a SL device does not necessarily require a nonoscillatory activation function; in fact, for many architectures of a SL device, the concept of an activation function does not apply (for example, consider a nonlinear core where the nonlinearity is homogeneously spread across the whole device). Finding the best architecture for each task will involve some future experimentation, but in this appendix, we offer some insights about how to choose the best nonlinear systems.

APPENDIX G: STABILIZATION OF THE LEARNING FIELD

The primary method we envisage for HEB concerns wave fields being transmitted through a nonlinear core and the subsequent time-reversal operation and decay step outside this core. In that case, as explained in the main text, we are insensitive to the details of the Hamiltonian of the nonlinear core.

However, one can also envisage situations where the learning field is not constantly being time reversed (during evaluation or training). When that is the case, one needs to make sure the learning field lives in a degenerate manifold, such that it is only the learning dynamics that will shift this learning field.

Once training via HEB has been completed and the cost function has been minimized, the learning parameters should remain stable for long times. The problem then is how to choose a physical system with degrees of freedom that can remain stable in a continuum of states. Mechanical degrees of freedom would be *a priori* a good selection. The position of freely moving nanoparticles or the orientation of a (possibly levitated) rotor (e.g., a molecule or nanorod) are examples of systems that could store a degree of freedom for indefinitely long times. The problem is that the time reversal of mechanical systems has no straightforward solution. The possible exceptions to this statement are optomechanical systems, where the coupling to the light field can be employed to time reverse mechanical motion, but then the mechanical degree of freedom oscillates around a single equilibrium position, making it unsuitable for our purposes, absent any additional ingredients. By contrast, in the case of wave fields, the experimental techniques for phase conjugation are better known, but storing information for longer times is more problematic.

The solution is to engineer a suitable physical system that has the desired characteristics. Modern systems in integrated nonlinear optics and in superconducting microwave circuits allow for the design of a wide range of nonlinear interactions between localized modes of the radiation field. In the following, we provide two examples of how this flexibility might be exploited to design a degenerate ground-state manifold for a learning field.

In the first example, we consider a cavity that has two modes $\alpha_{1,}\alpha_2$ with the same frequency (like counterpropagating modes in a microresonator). Inside the cavity, there is a nonlinear material that induces a repulsive Kerr-type interaction. Both modes have the same lifetime. In order to have stable states with nonzero amplitude, we couple both modes to a parametric drive. Let us define the quadratures of the field modes as $q_{1,2} = \mathrm{Re}\{\alpha_{1,2}\}/\sqrt{2}$, $p_{1,2} = \mathrm{Im}\{\alpha_{1,2}\}/\sqrt{2}$ and introduce the abbreviations $Q^2 \coloneqq q_1^2 + q_2^2$, $P^2 \coloneqq p_1^2 + p_2^2$. In terms of the quadratures, the full Hamiltonian reads

$$H = -\chi\beta Q^2 + gQ^4 + (\chi\beta + 2gQ^2)P^2 + gP^4. \tag{G1}$$

This Hamiltonian is written in the frame rotating at the mode frequency, and we assume the parametric drive amplitude β is purely real-valued. It is not hard to see that this Hamiltonian is qualitatively similar to that of a point particle moving inside a "Mexican hat" potential. Using this analogy, it is clear that any state with $P = 0$ and $Q^2 = \chi\beta/(2g)$ is a ground state. We could consider the angle in the (q_1, q_2) plane as our learning parameter θ. In order to couple to this parameter, the simplest option we found is to suppose that Ψ interacts differently with α_1 and α_2. For example, let us consider that there is a nonlinear element that couples Ψ only to α_1, through a cross-phase modulation. The interaction Hamiltonian can then be written in terms of the learning parameter as $H_{\mathrm{int}} = g\cos^2\theta\Psi^*\Psi$.

Of course, storing a learning parameter in the amplitudes of parametrically driven Kerr oscillators is not the only possibility. As a further example, we briefly present another setup in which θ is the relative phase between two modes. Consider two modes α_1, α_2 that are also parametrically driven. The two modes α_1, α_2 have different frequencies given by ω_1, ω_2, respectively. This physical system can be modeled by a Hamiltonian given by $\chi^{(2)}(\beta\alpha_1^*\alpha_2 + \beta^*\alpha_1\alpha_2^*) + \sum_{j=1}^2 \omega_j\alpha_j^*\alpha_j$. It is clear that this Hamiltonian is invariant with respect to any global phase shift $\alpha_j \to e^{i\varphi}\alpha_j$, $j = 1, 2$. Let us introduce the notation $\alpha_j \equiv |\alpha_j|e^{i\varphi_j}$, where φ_1, φ_2 are the phases of α_1, α_2, respectively. We define the learning parameter as $\theta \coloneqq \varphi_1 + \varphi_2$. Because of the symmetry of the Hamiltonian, θ is a free parameter in the ground state. The challenge is now to find a way to couple Ψ to the learning parameters. One possible way is to consider that Ψ has two components (field modes) Ψ_1, Ψ_2. Adequately choosing the frequency of the modes Ψ_1, Ψ_2, the product $g\alpha_1\alpha_2\Psi_1^*\Psi_2 + \mathrm{c.c.}$ can be made resonant. Then, θ parametrizes a linear interaction between the two components of Ψ. This requires frequency stabilization of the difference frequency of Ψ_1 and Ψ_2 vs the sum frequency of α_1 and α_2. The advantage, though, is that now the symmetry of the Hamiltonian is more robust.

As we illustrated above, in order to engineer a physical system that can store a continuous learning parameter, we need to realize a Hamiltonian with some continuous symmetry. The problem is that in any real implementation, we may expect that any continuous symmetry will be broken at some energy scale. If that is the case, most configurations of the learning field after training will have a finite lifetime. We assume that, after training, the learning field ends in one of the degenerate ground states of H'_θ, which is an ideal, exactly degenerate Hamiltonian. If the true Hamiltonian of the system is instead given by $H'_\theta + \delta H'$, the learning field will slide down in the energy landscape of $\delta H'$. Eventually, it will end up very far from the configuration that had been obtained as a result of training, unless the training is periodically repeated to stabilize the "memory" of the system.

This potential problem could be avoided if the learning parameters are stored in a discrete set of states. Consider that now H'_θ possesses a discrete set of degenerate ground states. In such a case, a small perturbation in H'_θ does not affect the long-term stability of θ, as long as it is much smaller than the energy barrier between the degenerate states. Stochastic gradient descent (SGD) is a strategy used to optimize continuous degrees of freedom. Although HEB is originally motivated by the search for a physical realization of SGD, we already argued that it can still be useful for training discrete parameters. In that case, HEB works in an analogous fashion to thermal annealing. In this picture, one can think of θ as a spin system that can be approximately described by a thermal distribution. The probability of each state is then given by $e^{-\beta H_{\mathrm{eff}}[\Theta]}$, where $H_{\mathrm{eff}}[\Theta]$ is the effective Hamiltonian of Eq. (9). If the temperature is slowly reduced, the system will undergo annealing, and Θ would end in one of the local minima of $H_{\mathrm{eff}}[\Theta]$. Spinlike degrees of freedom can be realized, for example, with optical parametric oscillators [66].

[1] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[2] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, *A Survey of Neuromorphic Computing and Neural Networks in Hardware*, arXiv:1705.06963.

[3] D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, *Physics for Neuromorphic Computing*, Nat. Rev. Phys. **2**, 499 (2020).

[4] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. P. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, *Photonics for Artificial Intelligence and Neuromorphic Computing*, Nat. Photonics **15**, 102 (2021).

[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 (London) **601**, 549 (2022).

[6] B. Scellier and Y. Bengio, *Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation*, Frontiers Comput. Neurosci **11**, 24 (2017).

[7] E. Martin, M. Ernoult, J. Laydevant, S. Li, D. Querlioz, T. Petrisor, and J. Grollier, *EqSpike: Spike-Driven Equilibrium Propagation for Neuromorphic Implementations*, Iscience **24**, 102222 (2021).

[8] K. Wagner and D. Psaltis, *Multilayer Optical Learning Networks*, Appl. Opt. **26**, 5061 (1987).

[9] H.-Y. S. Li, Y. Qiao, and D. Psaltis, *Optical Network for Real-Time Face Recognition*, Appl. Opt. **32**, 5026 (1993).

[10] K. Wagner and T. M. Slagle, *Optical Competitive Learning with VLSI/Liquid-Crystal Winner-Take-All Modulators*, Appl. Opt. **32**, 1408 (1993).

[11] S. R. Skinner, E. C. Behrman, A. A. Cruz-Cabrera, and J. E. Steck, *Neural Network Implementation Using Self-Lensing Media*, Appl. Opt. **34**, 4129 (1995).

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*, Nature (London) **323**, 533 (1986).

[13] D. Psaltis, D. Brady, and K. Wagner, *Adaptive Optical Networks Using Photorefractive Crystals*, Appl. Opt. **27**, 1752 (1988).

[14] G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljacic, C. Denz, D. A. B. Miller, and D. Psaltis, *Inference in Artificial Intelligence with Deep Optics and Photonics*, Nature (London) **588**, 39 (2020).

[15] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, *Deep Learning with Coherent Nanophotonic Circuits*, Nat. Photonics **11**, 441 (2017).

[16] X. Guo, T. D. Barrett, Z. M. Wang, and A. I. Lvovsky, *Backpropagation through Nonlinear Units for All-Optical Training of Neural Networks*, Photonics Research **9**, B71 (2021).

[17] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. P. Pernice, *All-Optical Spiking Neurosynaptic Networks with Self-Learning Capabilities*, Nature (London) **569**, 208 (2019).

[18] 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).

[19] S. Hu, Y. Liu, Z. Liu, T. Chen, J. Wang, Q. Yu, L. Deng, Y. Yin, and S. Hosaka, *Associative Memory Realized by a Reconfigurable Memristive Hopfield Neural Network*, Nat. Commun. **6**, 7522 (2015).

[20] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, *Memristor-Based Multilayer Neural Networks with Online Gradient Descent Training*, IEEE Trans. Neural Netw. Learn. Syst. **26**, 2408 (2015).

[21] Q. Xia and J. J. Yang, *Memristive Crossbar Arrays for Brain-Inspired Computing*, Nat. Mater. **18**, 309 (2019).

[22] M. Schneider, E. Toomey, G. Rowlands, J. Shainline, P. Tschirhart, and K. Segall, *Supermind: A Survey of the Potential of Superconducting Electronics for Neuromorphic Computing*, Supercond. Sci. Technol. **35**, 053001 (2022).

[23] A. Sengupta and K. Roy, *Spin-Transfer Torque Magnetic Neuron for Low Power Neuromorphic Computing*, in *2015 International Joint Conference on Neural Networks (IJCNN)* (IEEE, Killarney, Ireland, 2015), pp. 1–7.

[24] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima *et al.*, *Neuromorphic Computing with Nanoscale Spintronic Oscillators*, Nature (London) **547**, 428 (2017).

[25] J. Grollier, D. Querlioz, K. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles, *Neuromorphic Spintronics*, Nat. Electron. **3**, 360 (2020).

[26] A. F. Vincent, J. Larroque, N. Locatelli, N. B. Romdhane, O. Bichler, C. Gamrat, W. S. Zhao, J.-O. Klein, S. Galdin-Retailleau, and D. Querlioz, *Spin-Transfer Torque Magnetic Memory as a Stochastic Memristive Synapse for Neuromorphic Systems*, IEEE Trans. Biomed. Circuits Syst. **9**, 166 (2015).

[27] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, *Nanoscale Memristor Device as Synapse in Neuromorphic Systems*, Nano Lett. **10**, 1297 (2010).

[28] J. M. Shainline, S. M. Buckley, R. P. Mirin, and S. W. Nam, *Superconducting Optoelectronic Circuits for Neuromorphic Computing*, Phys. Rev. Appl. **7**, 034013 (2017).

[29] K. Friston, *The Free-Energy Principle: A Unified Brain Theory?*, Nat. Rev. Neurosci. **11**, 127 (2010).

[30] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, *A Solution to the Learning Dilemma for Recurrent Networks of Spiking Neurons*, Nat. Commun. **11**, 3625 (2020).

[31] J. Sacramento, R. Ponte Costa, Y. Bengio, and W. Senn, *Dendritic Cortical Microcircuits Approximate the Backpropagation Algorithm*, Adv. Neural Inf. Process. Syst. **31**, 8735 (2018), https://arxiv.org/abs/1810.11393.

[32] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, *Backpropagation and the Brain*, Nat. Rev. Neurosci. **21**, 335 (2020).

[33] G. E. Hinton and J. McClelland, *Learning Representations by Recirculation*, in *Neural Information Processing Systems 0* (1987), https://proceedings.neurips.cc//paper_files/paper/1987.

[34] M. Hermans, M. Burm, T. V. Vaerenbergh, J. Dambre, and P. Bienstman, *Trainable Hardware for Dynamical Computing Using Error Backpropagation through Physical Media*, Nat. Commun. **6**, 6729 (2015).

[35] M. Hermans, M. Soriano, J. Dambre, P. Bienstman, and I. Fischer, *Photonic Delay Systems as Machine Learning Implementations*, J. Mach. Learn. Research **16**, 2081 (2015).

[36] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, *Training of Photonic Neural Networks through in situ Backpropagation and Gradient Measurement*, Optica **5**, 864 (2018).

[37] *Green's Functions and Boundary Value Problems: Stakgold/Green's Functions*, edited by I. Stakgold and M. Holst (John Wiley & Sons, Inc., Hoboken, NJ, 2011).

[38] L. Hand and J. Finch, *Analytical Mechanics* (Cambridge University Press, Cambridge, England, 1998), pp. 458–459.

[39] A. E. Motter, M. Gruiz, G. Károlyi, and T. Tél, *Doubly Transient Chaos: Generic Form of Chaos in Autonomous Dissipative Systems*, Phys. Rev. Lett. **111**, 194101 (2013).

[40] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, *The Reversible Residual Network: Backpropagation without Storing Activations*, Adv. Neural Inf. Process. Syst. **30**, 2211 (2017), https://arxiv.org/abs/1707.04585.

[41] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, *Reversible Architectures for Arbitrarily Deep Residual Neural Networks*, Proc. AAAI Conf. Art. Intell. **32**, 2811 (2018), https://arxiv.org/abs/1709.03698.

[42] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen, *Invertible Residual Networks*, *Proceedings of the 36th International Conference on Machine Learning* 97, 573 (2019).

[43] R. Boyd, *Contemporary Nonlinear Optics* (Academic Press, New York, 2012).

[44] G. He, *Optical Phase Conjugation: Principles, Techniques, and Applications*, Prog. Quantum Electron. **26**, 131 (2002).

[45] A. P. Brysev, L. Krutyanskii, and V. L. Preobrazhenskii, *Wave Phase Conjugation of Ultrasonic Beams*, Phys. Usp. **41**, 793 (1998).

[46] L. Deng, E. W. Hagley, J. Wen, M. Trippenbach, Y. Band, P. S. Julienne, J. Simsarian, K. Helmerson, S. Rolston, and W. D. Phillips, *Four-Wave Mixing with Matter Waves*, Nature (London) **398**, 218 (1999).

[47] Y. Chang, H. R. Fetterman, I. L. Newberg, and S. K. Panaretos, *Microwave Phase Conjugation Using Antenna Arrays*, IEEE Trans. Microwave Theory Tech. **46**, 1910 (1998).

[48] A. Serga, B. Hillebrands, S. Demokritov, A. Slavin, P. Wierzbicki, V. Vasyuchka, O. Dzyapko, and A. Chumak, *Parametric Generation of Forward and Phase-Conjugated Spin-Wave Bullets in Magnetic Films*, Phys. Rev. Lett. **94**, 167202 (2005).

[49] D. A. B. Miller, *Time Reversal of Optical Pulses by Four-Wave Mixing*, Opt. Lett. **5**, 300 (1980).

[50] J. Martin, B. Georgeot, and D. L. Shepelyansky, *Time Reversal of Bose-Einstein Condensates*, Phys. Rev. Lett. **101**, 074102 (2008).

[51] Y. Zheng, H. Ren, W. Wan, and X. Chen, *Time-Reversed Wave Mixing in Nonlinear Optics*, Sci. Rep. **3**, 3245 (2013).

[52] I. Sackey, F. Da Ros, M. Jazayerifar, T. Richter, C. Meuer, M. Nölle, L. Molle, C. Peucheret, K. Petermann, and C. Schubert, *Kerr Nonlinearity Mitigation in 5 Å 28-GBd PDM 16-QAM Signal Transmission over a Dispersion-Uncompensated Link with Backward-Pumped Distributed Raman Amplification*, Opt. Exp. **22**, 27381 (2014).

[53] M. Frazier, B. Taddese, T. Antonsen, and S. M. Anlage, *Nonlinear Time Reversal in a Wave Chaotic System*, Phys. Rev. Lett. **110**, 063902 (2013).

[54] A. Przadka, S. Feat, P. Petitjeans, V. Pagneux, A. Maurel, and M. Fink, *Time Reversal of Water Waves*, Phys. Rev. Lett. **109**, 064501 (2012).

[55] I. N. Papadopoulos, S. Farahi, C. Moser, and D. Psaltis, *Focusing and Scanning Light through a Multimode Optical Fiber Using Digital Phase Conjugation*, Opt. Express **20**, 10583 (2012).

[56] J. R. Ong, C. C. Ooi, T. Y. L. Ang, S. T. Lim, and C. E. Png, *Photonic Convolutional Neural Networks Using Integrated Diffractive Optics*, IEEE J. Sel. Top. Quantum Electron. **26**, 1 (2020).

[57] 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).

[58] W. Yoshiki and T. Tanabe, *All-Optical Switching Using Kerr Effect in a Silica Toroid Microcavity*, Opt. Express **22**, 24332 (2014).

[59] J. Sun, E. Timurdogan, A. Yaacobi, E. S. Hosseini, and M. R. Watts, *Large-Scale Nanophotonic Phased Array*, Nature (London) **493**, 195 (2013).

[60] M. C. Rechtsman, J. M. Zeuner, Y. Plotnik, Y. Lumer, D. Podolsky, F. Dreisow, S. Nolte, M. Segev, and A. Szameit, *Photonic Floquet Topological Insulators*, Nature (London) **496**, 196 (2013).

[61] N. Sarapat, T. Frank, and P. Yupapin, *Conjugate Mirror Design and Simulation Using a Nonlinear Coupling Microring Circuit*, J. Nonlinear Opt. Phys. Mater. **22**, 1350024 (2013).

[62] A. C. Turner, M. A. Foster, A. L. Gaeta, and M. Lipson, *Ultra-Low Power Parametric Frequency Conversion in a Silicon Microring Resonator*, Opt. Express **16**, 4881 (2008).

[63] S. Ayotte, H. Rong, S. Xu, O. Cohen, and M. J. Paniccia, *Multichannel Dispersion Compensation Using a Silicon Waveguide-Based Optical Phase Conjugator*, Opt. Lett. **32**, 2393 (2007).

[64] M. W. Puckett, K. Liu, N. Chauhan, Q. Zhao, N. Jin, H. Cheng, J. Wu, R. O. Behunin, P. T. Rakich, K. D. Nelson *et al.*, *422 Million Intrinsic Quality Factor Planar Integrated All-Waveguide Resonator with Sub-MHz Linewidth*, Nat. Commun. **12**, 934 (2021).

[65] D. Psaltis, D. Brady, X.-G. Gu, and S. Lin, *Holography in Artificial Neural Networks*, Nature (London) **343**, 325 (1990).

[66] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K.-i. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, *A Coherent Ising Machine for 2000-Node Optimization Problems*, Science **354**, 603 (2016).

[67] Y. Yamamoto, K. Aihara, T. Leleu, K.-i. Kawarabayashi, S. Kako, M. Fejer, K. Inoue, and H. Takesue, *Coherent Ising Machines—Optical Neural Networks Operating at the Quantum Limit*, npj Quantum Inf. **3**, 49 (2017).

[68] D. Pierangeli, G. Marcucci, and C. Conti, *Large-Scale Photonic Ising Machine by Spatial Light Modulation*, Phys. Rev. Lett. **122**, 213902 (2019).

[69] M. Devoret, *Quantum Fluctuations in Electrical Circuits*, in *Les Houches Lecture Notes of the 1995 School on Quantum Fluctuations*, edited by S. Reynaud, E. Giacobino, and J. Zinn-Justin (Elsevier, New York, 1997).

[70] E. Toomey, K. Segall, M. Castellani, M. Colangelo, N. Lynch, and K. K. Berggren, *Superconducting Nanowire Spiking Element for Neural Networks*, Nano Lett. **20**, 8059 (2020).

[71] Q. Wang, A. Hamadeh, R. Verba, V. Lomakin, M. Mohseni, B. Hillebrands, A. V. Chumak, and P. Pirro, *A Nonlinear Magnonic Nano-Ring Resonator*, npj Comput. Mater. **6**, 192 (2020).

[72] See https://github.com/victorlopezpastor/self_learning_machines (2022).

[73] V. L. Pastor, J. Lundeen, and F. Marquardt, *Arbitrary Optical Wave Evolution with Fourier Transforms and Phase Masks*, Opt. Express **29**, 38441 (2021).

[74] T. W. Hughes, I. A. D. Williamson, M. Minkov, and S. Fan, *Wave Physics as an Analog Recurrent Neural Network*, Sci. Adv. **5**, eaay6946 (2019).

[75] G. Angelatos, S. A. Khan, and H. E. Türeci, *Reservoir Computing Approach to Quantum State Measurement*, Phys. Rev. X **11**, 041062 (2021).

[76] J. Hochstetter, R. Zhu, A. Loeffler, A. Diaz-Alvarez, T. Nakayama, and Z. Kuncic, *Avalanches and Edge-of-Chaos Learning in Neuromorphic Nanowire Networks*, Nat. Commun. **12**, 4008 (2021).

[77] C. Keup, T. Kühn, D. Dahmen, and M. Helias, *Transient Chaotic Dimensionality Expansion by Recurrent Networks*, Phys. Rev. X **11**, 021064 (2021).

[78] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, *Exponential Expressivity in Deep Neural Networks through Transient Chaos*, Adv. Neural Inf. Process. Syst. **29**, 3368 (2016), https://arxiv.org/abs/1606.05340.

[79] M. Farrell, S. Recanatesi, T. Moore, G. Lajoie, and E. Shea-Brown, *Gradient-Based Learning Drives Robust Representations in Recurrent Neural Networks by Balancing Compression and Expansion*, Nat. Mach. Intell. **4**, 564 (2022).

[80] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, *Experimental Realization of Any Discrete Unitary Operator*, Phys. Rev. Lett. **73**, 58 (1994).

[81] N. J. Doran and D. Wood, *Nonlinear-Optical Loop Mirror*, Opt. Lett. **13**, 56 (1988).

[82] M. Soljacic, M. Ibanescu, S. G. Johnson, Y. Fink, and J. D. Joannopoulos, *Optimal Bistable Switching in Nonlinear Photonic Crystals*, Phys. Rev. E **66**, 055601 (2002).

[83] M.-S. Kang, J. Heo, S.-G. Choi, S. Moon, and S.-W. Han, *Optical Fredkin Gate Assisted by Quantum Dot within Optical Cavity under Vacuum Noise and Sideband Leakage*, Sci. Rep. **10**, 5123 (2020).