ANNUAL REVIEWS

*Annual Review of Condensed Matter Physics*

# Learning Without Neurons in Physical Systems

Menachem Stern[1] and Arvind Murugan[2]

[1]Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, Pennsylvania; email: nachis@sas.upenn.edu

[2]Department of Physics, University of Chicago, Chicago, Illinois; email: amurugan@uchicago.edu

Annu. Rev. Condens. Matter Phys. 2023. 14:417–41

The *Annual Review of Condensed Matter Physics* is online at conmatphys.annualreviews.org

https://doi.org/10.1146/annurev-conmatphys-040821-113439

## Keywords

physical learning, learning theory, inverse design, metamaterials, machine learning, self-assembly, molecular computing

## Abstract

Learning is traditionally studied in biological or computational systems. The power of learning frameworks in solving hard inverse problems provides an appealing case for the development of physical learning in which physical systems adopt desirable properties on their own without computational design. It was recently realized that large classes of physical systems can physically learn through local learning rules, autonomously adapting their parameters in response to observed examples of use. We review recent work in the emerging field of physical learning, describing theoretical and experimental advances in areas ranging from molecular self-assembly to flow networks and mechanical materials. Physical learning machines provide multiple practical advantages over computer designed ones, in particular by not requiring an accurate model of the system, and their ability to autonomously adapt to changing needs over time. As theoretical constructs, physical learning machines afford a novel perspective on how physical constraints modify abstract learning theory.

# 1. INTRODUCTION

## 1.1. Physical Learning

When no off-the-shelf solutions are readily available, systems are often rationally designed to provide solutions to our everyday problems. Designing such a system is known as solving an "inverse problem" because one attempts to find a system with a desired response to a perturbation, in contrast to the "forward problem" of predicting responses of a given system to perturbations. To solve inverse problems, one must sift through possible systems by varying design parameters or design degrees of freedom (d.o.f). Solutions are typically sought using centralized, top-down approaches, where a designer can access and modify the elements of a system. Numerous computational algorithms have been developed for designing a wide range of physical systems ranging from materials to robots, including simulated annealing and genetic algorithms, that typically employ standard computers and may involve physical prototypes (**Figure 1*a***). Inverse problems are usually much more difficult to solve compared to forward prediction problems, in particular because solutions are typically nonunique.

Learning is an established bottom-up framework for solving inverse problems (1) with specific advantages. Learning has been explored in the biological context of the brain and in artificial neural networks. However, learning is a more broadly applicable framework in which a system is changed incrementally in a way that enables adoption of a desired behavior. These incremental changes are defined as a function of the system's response to external stimuli. A learning process consists of two major parts: (*a*) evaluating the system's output for a given input, and (*b*) modification of the system based on the output. The system is first evaluated by applying stimuli from a training set. Then, the response of the system to the stimuli drives the modification of learning d.o.f, such that subsequent application of stimuli result in improved responses. This process is usually conducted iteratively until the system achieves satisfactory performance.

Here, we explore the emerging area of physical learning in materials. Typically, no computers are involved in this approach. Instead, a material is physically subject to examples of the desired behavior. In response, elements of the material—the learning d.o.f—change as described by a system-dependent dynamical process called a learning rule (**Figure 1*a***), underpinning the bottom-up, distributed nature of learning. A physical learning system thus has the following ingredients:

1. Physical d.o.f, $\mathbf{s}$, that respond to external stimuli, $f$, by adopting a state or dynamic behavior, $\mathbf{s}(f)$, a subset of which defines the desired behavior (i.e., output).
2. Learning d.o.f, $w_i$, that can modify how the physical d.o.f, $\mathbf{s}(f;\{w_i\})$, respond to external stimuli, $f$.
3. A learning rule, $\mathrm{d}w_i = h(\mathbf{s}(f;\{w_i\}))\mathrm{d}t$, that modifies the learning d.o.f, $w_i$, based on the response of the physical d.o.f, $\mathbf{s}$, to the stimuli, $f$.

Here, $h$ is an arbitrary nonlinear function. The quality of learning is evaluated by whether the system learns the desired behavior, often quantified by a learning cost function $C(\mathbf{s}(f;\{w_i\}))$. Note that the dynamics of the physical system itself do not compute the cost function; instead, physical learning seeks to minimize this cost function through the learning rule and the choice of stimuli used for training.

Broadly, we can divide physical learning into categories based on the amount of user input during training: (*a*) Physical unsupervised learning (**Figure 1*b***), where a physical system self-adapts to instances of external stimuli with no supervisor intervention, typically by a Hebbian-like process (8), and thereby learns to represent some aspect of these stimuli, and (*b*) physical supervised learning (**Figure 1*c***), where an error signal is provided by a supervisor comparing the obtained and desired physical outcome, so systems can learn to exhibit specific responses for specific stimuli.

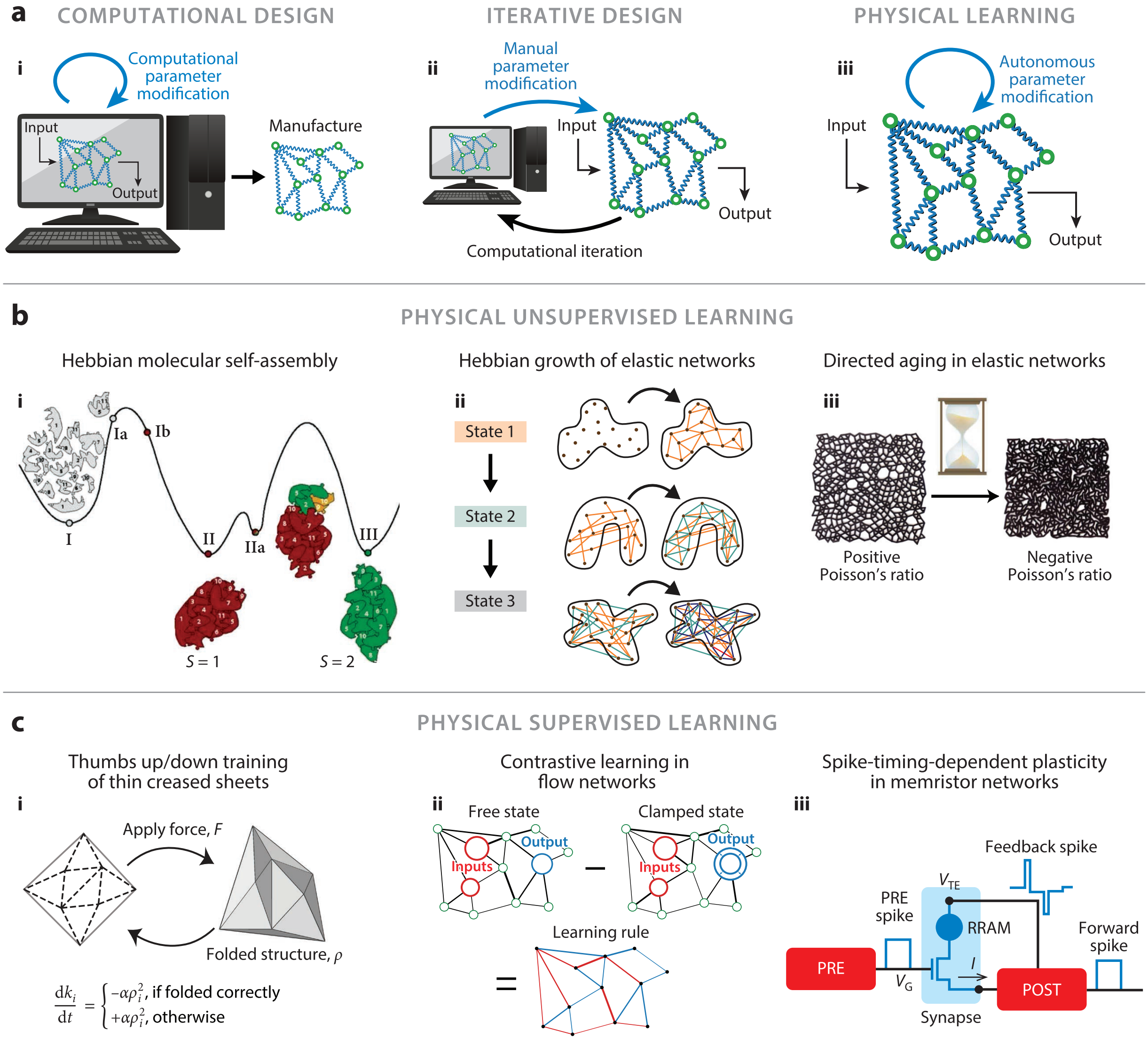

a
COMPUTATIONAL DESIGN
i
Computational parameter modification
Input
Output
Manufacture
ITERATIVE DESIGN
ii
Manual parameter modification
Input
Output
Computational iteration
PHYSICAL LEARNING
iii
Autonomous parameter modification
Input
Output
b
PHYSICAL UNSUPERVISED LEARNING
Hebbian molecular self-assembly
i
I
Ia
Ib
II
IIa
III
S = 1
S = 2
Hebbian growth of elastic networks
ii
State 1
State 2
State 3
Directed aging in elastic networks
iii
Positive Poisson's ratio
Negative Poisson's ratio
c
PHYSICAL SUPERVISED LEARNING
Thumbs up/down training of thin creased sheets
i
Apply force, F
Folded structure, ρ
$\frac{dk_i}{dt} = \begin{cases} -\alpha\rho_i^2, \text{ if folded correctly} \\ +\alpha\rho_i^2, \text{ otherwise} \end{cases}$
Contrastive learning in flow networks
ii
Free state
Clamped state
Output
Inputs
Output
Inputs
−
Learning rule
=
Spike-timing-dependent plasticity in memristor networks
iii
Feedback spike
$V_{TE}$
PRE spike
RRAM
Forward spike
PRE
$V_G$
$I$
POST
Synapse

**Figure 1**

Physical learning versus computer-aided design. (*a*) Materials are often (*i*) computationally designed for particular properties or responses, either entirely on a computer or through (*ii*) an iterative design-build-test process. In contrast, given external stimuli, in physical learning (*iii*), materials autonomously modify their parameters to adopt desired properties or functions. Such autonomous learning machines modify themselves based on their response to stimuli according to physical "learning rules," which can be classified by the level of supervision. (*b*) Physical unsupervised learning, e.g., (*i*) molecular self-assembly with Hebbian-learned interactions (2), (*ii*) Hebbian growth (3), and (*iii*) directed aging in elastic networks (4; subpanel provided by N. Pashine). (*c*) Physical supervised learning, e.g., (*i*) thumbs-up-or-down rules in creased sheets (5), (*ii*) contrastive learning in flow networks (6), and (*iii*) spike-timing-dependent plasticity in memristor neural nets (subpanel adapted from Reference 7; CC BY 4.0). Abbreviations: PRE, presynaptic neuron; POST, postsynaptic neuron; RRAM, resistive random access memory.

## 1.2. Why Learn Using a Physical System?

Physical learning blurs the boundary between structure and function; the physical elements responsible for material properties also perform the task of adapting those material properties. Such a merger is motivated by both practical and theoretical considerations.

At a practical level, physical learning can lead to a new generation of materials capable of autonomous learning that offer several advantages over computer design. (*a*) Although silicon-based computing and learning is incredibly powerful, a physical system that learns is more appropriate to perform tasks in which the inputs and outputs are physical influences from or on the environment (e.g., forces, currents, molecule production) rather than symbolic information. (*b*) In this new framework, physical systems can learn from real examples of the desired behavior, so that the desired behaviors themselves do not need to be modeled on a computer. (*c*) Similarly, a detailed model of the physical system is not needed because the response of the real physical system, imperfections included, determines physical learning. In contrast, computer-aided design is only as good as the model of the material and the model of the desired behavior and environment in which the material will be used. (*d*) Physical learning can allow systems to continually learn new behaviors in situ as requirements change. (*e*) Physical learning may be appropriate for applications with space, time, or energy constraints or where robustness due to the distributed nature of learning and information processing (rather than in one electronic processor) is important.

From a theoretical perspective, studying learning in physical systems may shed new light on the fundamental requirements and limitations of learning in the presence of physical constraints. How can natural systems evolve the ability to learn? A theory of physical learning would expand our current modular conception of learning and memory in biological systems; we often look for control units (e.g., the brain) separable from the parts being controlled (e.g., the muscle). For example, even in single-celled organisms, we tend to look for a gene regulatory network (or protein interaction circuit) that learns and makes decisions that are then carried out downstream by, say, physical self-assembly or cytoskeletal processes (9). But the downstream "muscle" itself can potentially learn and make decisions, as known to be the case in organisms ranging from ciliates to fruit flies and beehives (10–13). Physical learning could reveal the broader scope of such nonmodular learning and information processing available for free in the muscle (14).

Finally, this approach can provide a framework to understand atypical disordered systems and how physical systems explore parameter space to arrive at such atypical points. Although we have made much progress in studying typical disordered systems through random ensembles, many examples of disordered systems in nature are highly atypical, e.g., due to evolution or other natural processes. Physical learning provides one framework for such atypical phenomena as the origin of structures and architectures (e.g., hierarchies) in networks (15) and the prevalent low dimensionality of physical responses (16, 17).

The framework outlined here also builds upon ideas previously explored in materials science, where materials are subject to some conditions (e.g., compression, temperature changes) with the goal of internal rearrangements that enhance desired properties. For example, metals have been hardened by thermomechanical protocols (18), which reshape grain size and distribution in a way that improves strength, which in turn is part of a long tradition of using annealing and hot stamping protocols (19). Similar ideas apply to processing polymers by which the same polymer mixes can be trained to rearrange molecules with different resultant mechanical properties by, e.g., different extrusion protocols (20).

The framework here goes beyond such processing techniques by expanding the space of behaviors training can be used for, including supervised learning for input–output responses and pattern recognition of spatial or temporal correlations in chemical and mechanical stimuli. The framework also significantly expands the range of learning mechanisms beyond molecular rearrangements to include growth and degradation, and active learning of molecular interactions. Many of the advances here are a conceptual synthesis between ideas in materials processing and ideas of learning in computer science. A closely related development is the idea of memory in materials (21), discussed in Section 5.1

## 1.3. Challenge and Opportunity: Local Learning Rules

Machine learning (ML) algorithms implemented on a computer typically compute a cost function, $C(\{w_i\})$, a global quantity that reflects performance on a task at hand, and strives to minimize it, e.g., through gradient descent,

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} \sim -\nabla_{w_i} C(\{w_j\}) \quad \text{(nonlocal gradient descent).} \qquad 1.$$

Gradient descent procedures are efficiently implemented on computers (especially graphics processing units or GPUs), e.g., in artificial neural networks through backpropagation (22). Note that such a process is highly nonlocal when $w_i$ are interpreted as parts of a physical system: The change $\mathrm{d}w_i$ in a given element $w_i$ depends in principle on the entire neural network, including, e.g., how a "distant" neuron responds to stimuli.

Such highly nonlocal changes cannot be realized in physical learning without effectively mimicking a computer algorithm with physical components. Instead, we seek to understand to what extent physical systems can learn by exploiting typically local natural processes without any explicit cost function,

$$\frac{\mathrm{d}w[x,t]}{\mathrm{d}t} \sim h(\mathbf{s}(f;\{w\})[x,t]) \quad \text{(local learning rule).} \qquad 2.$$

Here, the change in a learning d.o.f $w[x, t]$ located at a point $x$, $t$ in the physical system is restricted to change only based on the state $\mathbf{s}[x,t]$ of the system at the same (or close vicinity) spacetime point. (Here, $h$ is an arbitrary local function.) Note, however, that the local state $\mathbf{s}(f; \{w\})[x, t]$ will generically depend on a stimulus $f$ and learning d.o.f $\{w\}$ of the entire system at distant points because of collective dynamics of the physical system; e.g., the way an elastic material deforms at point $x$ due to a force $f$ can depend on force components $f(y)$ and on material properties at distant points $y$.

Thus, there is no explicit cost function or global optimization involved in local rules in Equation 2. However, if the local learning rule $h$ and stimuli $f$ are chosen correctly, the system can nevertheless minimize a cost function, as the collective physical dynamics during the response to stimuli can encode global information in the local state $\mathbf{s}(f,\{w\})[x,t]$.

Physical systems are in principle more constrained in their learning abilities than an in silico neural network (23). However, conceptually similar constraints also distinguish the brain from artificial neural networks; learning in the former is often constrained by locality [e.g., Hebb's rule (8) or spike-timing-dependent plasticity (STDP; 24)]. Nevertheless, biologically plausible learning rules, simulated on computers, have proven successful (25), suggesting that physical learning has potential despite locality constraints (26). Furthermore, locality provides benefits as well, because learning can be desynchronous and more robust, and it can scale better with system size because it does not rely on a central processor (27, 28).

## 1.4. Relationship to Machine Learning, Neuromorphic Computing, and Physical Computation

There have been many threads of work related to physical learning over the previous years. The field of neuromorphic computation (29, 30) is closely related to physical learning in that physical elements are modified so that the system adopts a desired computational ability. However, neuromorphic computers are often considered for a distinct objective—to compete with in silico ML in its domain of symbolic inputs and outputs. The physical learning machines explored here can deal with problems in which the inputs and outputs are physical (e.g., forces, molecular assemblies). Neuromorphic computing and in silico ML algorithms do not directly compete in this realm because they require translation of stimuli to electronic or other symbolic inputs. More saliently, these methods cannot directly display any physical output such as an elastic response or a self-assembled molecular structure in response to a stimulus.

The fields of physical computation (31) and molecular computation (32, 33) explore how to implement fixed computations with physical systems. The systems explored here build on these prior works by autonomously learning what computation (or more generally, physical behavior) needs to be carried out in the first place by physically experiencing examples of the desired behavior. Another closely related physical computation framework is the field of reservoir computing (34). Reservoir computing uses physical systems with such complex internal dynamics that no physical elements need to change during a learning process; instead, learning how to interface with a fixed physical system can effectively define inputs and outputs in a way that solves inverse problems. This interface can be learned using a computer as an output filter (35).

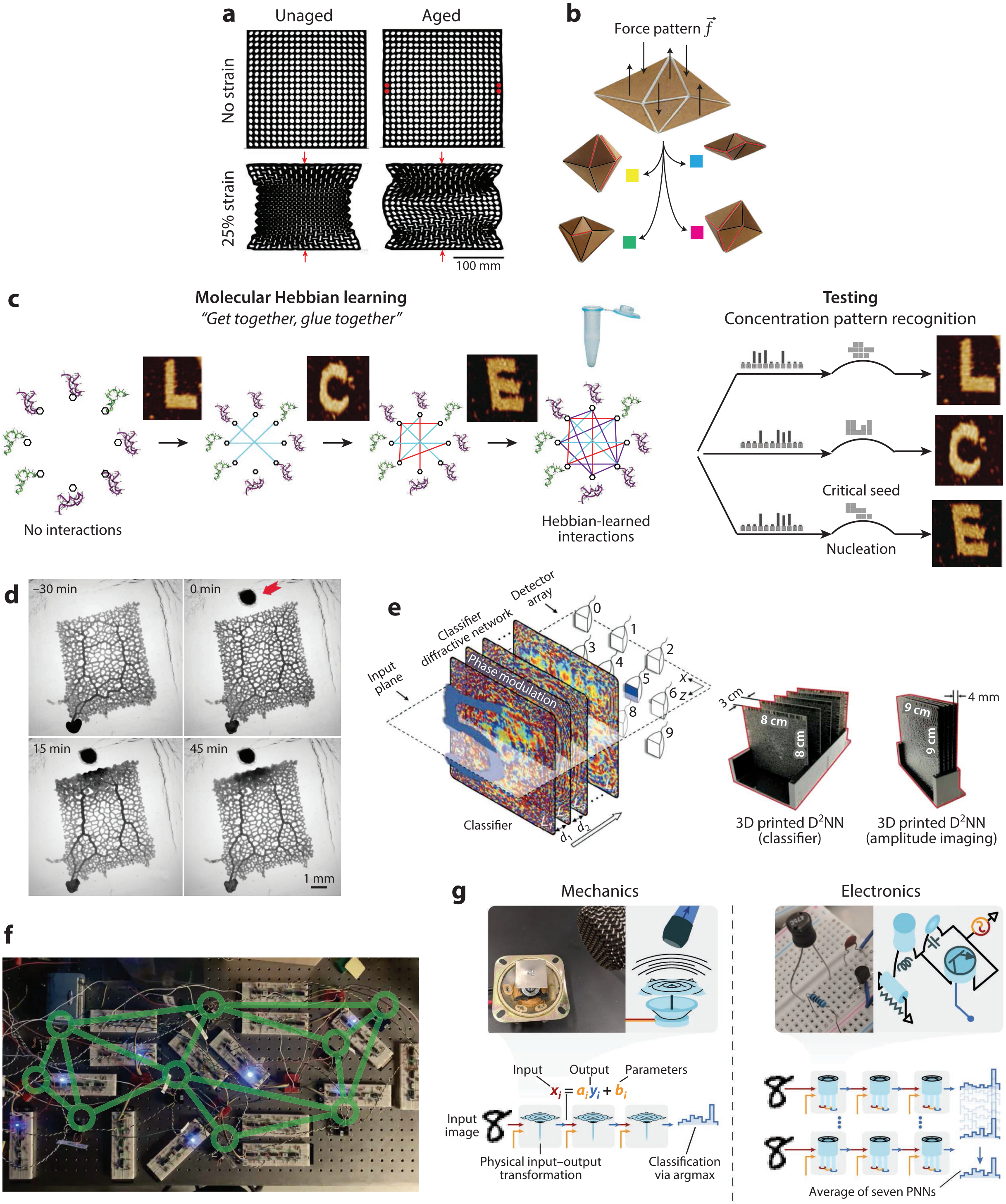
In this review, we strive to showcase recent experimental and theoretical advances in the emerging field of physical learning. The remainder of the review is organized as follows: In Section 2, we feature examples of physical learning machines, emphasizing ways of experimentally realizing physical learning. In Section 3, we discuss the physically realizable learning rules powering learning machines, subdivided into unsupervised and supervised learning rules. In Section 4, we describe practical aspects inherent to realizing learning machines in experiments. In Section 5, we highlight some theoretical learning concepts and how they manifest in the framework of physical learning. Finally, Section 6 focuses on how the physical properties of a system are modified as it undergoes physical learning.

## 2. EXAMPLES OF PHYSICAL LEARNING MACHINES

### 2.1. Elastic Materials: Networks and Sheets

Disordered elastic media and their abstractions, ranging from spring networks to creased sheets, have received much interest for their potential to mechanically respond in nonstandard, desired ways (36–38). Recent works have explored the feasibility of physically learned behaviors in these systems (3, 4, 39–41). Here, the learning d.o.f $w_i$ generally involve either bond lengths or bond stiffnesses in elastic networks and crease stiffnesses in creased sheets. For example, auxetic materials (negative Poisson's ratio; **Figure 1*b***, subpanel *iii*) were trained by imposing the desired global deformation (the stimulus $f$) on a material (**Figure 2*a***), which leads to a pattern of local strain across the material [the response $\mathbf{s}(f)$]. A learning process, typical of these works, involves bond stiffnesses $w_i$ that change according to the strain in that bond (i.e., locally) according to rules such as $\frac{\mathrm{d}w_i}{\mathrm{d}t} \sim -(\mathrm{strain}_i)^2$.

Physical learning can also train nonlinear features like bifurcated folding pathways in mechanical systems (**Figure 1*c***, subpanel *i*). Origami and Kirigami sheets with disordered crease patterns have rich folding topologies (48), starting from 2D flat sheets with creases and slits, and ending with various folded shapes in response to given force patterns. While the crease geometry is usually fixed by fabrication, the bending stiffness of creases can serve as learning d.o.f $w_i$ and may change in response to folding strain. Theory and experimentations have shown that folding pathways of such sheets can be trained for specific topologies with applications to dynamical control of folding (49), mitigation of misfolding pathways (50, 50a) and even classification of mechanical force patterns, analogous to neural networks (5; **Figure 2*b***).



(*Caption appears on following page*)

**Figure 2** (*Figure appears on preceding page*)

Examples of experimentally realized physical learning machines. (*a*) Elastic network, aged under stress, adapts nongeneric auxetic responses. Panel adapted from Reference 4 (CC BY 4.0). (*b*) Self-folding origami sheets trained to fold into desired shape in response to an input force pattern. (*c*) Pattern recognition through nucleation in a soup of molecules with Hebbian-inspired interactions, proposed theoretically (42), experimentally tested using DNA self-assembly (43). Panel adapted from References 42 and 44. (*d*) Pinned *Physarum polycephalum* network changes its structure to memorize locations of food sources. Panel adapted from Reference 45; published under PNAS license. (*e*) Optical diffractive neural network constructed to classify digits and fashion products. Panel adapted from Reference 46 with permission from AAAS. (*f*) Electric circuit with variable resistors performs supervised learning and is robust to damage to individual elements (28). (*g*) Mechanical, electronic, and optical networks trained to classify handwritten digits using physics-aware-training. Panel adapted from Reference 47 (CC BY 4.0). Abbreviation: PNN, physical neural network.

## 2.2. Molecular Systems and Active Matter

Molecular systems with specific interactions show behaviors that define the complexity of life, ranging from computation and neural network-like information processing (32, 51–53) to structural behaviors like self-assembly of complex and dynamic structures, targeted phase separation and active matter (54–56). While most work tends to assume fixed interactions, several works have explored ways of treating molecular interactions as learning d.o.f $w_i$ and thus learning complex molecular behaviors.

One broad learning mechanism involves creating new molecular species through ligation or polymerization (57, 58) or by preferential multiplication of molecules that bind input molecules tightly and discarding of those that do not bind (59, 60; **Figure 2*c***). These new species then mediate new interactions $w_i$ contingent on the history of the system (61, 62). One recent example (2, 63) explores learning interactions needed for self-assembly using a Hebbian-inspired "localized together, wired together" rule (**Figure 1*b***, subpanel *i*): $\mathrm{d}w_{ij}/\mathrm{d}t = s_i(x,t)s_j(x,t)$ where $s_i(x,t)$ is the concentration of species $i$ at $x$, $t$ and $w_{ij}$ is the interaction strength between species $i$ and $j$. That is, species $i, j$ whose concentrations are high in the same place and time start interacting more strongly. The collective nucleation dynamics in such a system has been shown experimentally (43) to be capable of pattern recognition by assembling different structures in response to different concentration patterns (63), with mathematical connections to Hopfield's associative memory (64).

Similar ideas of learning in multicomponent phase separation (65, 66), e.g., based on DNA nanostars (55), are likely to be realized in the coming years. All of these works suggest that inevitable physical processes (52) such as self-assembly, phase separation and nucleation can enable learning and perform pattern recognition, despite not being set up to mimic a neural network element by element (63) as in circuit approaches.

Another important type of molecular learning involves growth. Molecular systems like hydrogels and DNA nanotubes can grow elements based on their current geometry (67, 68) and can potentially continually learn multi-stability (3). Some molecular systems such as crystals with defects are capable of rudimentary evolution through repeated fracture and growth; this aspect raises the intriguing possibility of learning through evolution in nonbiological systems (69).

Learning in well-mixed molecular circuits (as opposed to the structural processes above) has a larger literature, ranging from bio-inspired associative learning (70) to recent progress towards molecular neural networks (53, 71). Molecular systems can also exploit their intrinsic stochasticity needed, e.g., for Boltzmann machines (72).

Active systems (e.g., self-propelled particles) can expend energy and perform actions on microscopic scales. Such systems are potentially capable of learning either by combining synthetic active matter with DNA-based elements (73) or through rearrangement of nematic or polar filaments as in natural systems such as the cytoskeleton (74, 75). Related ideas have been explored through simulations (76–78), though truly autonomous learning has yet to be demonstrated on the molecular scale. Autonomous learning can be realized more readily in macroscopic robotic swarms (79, 80) that demonstrate similar active matter phenomena. Similarly, tissue-level rearrangements through active adhesion and junction changes might also allow for training (81).

## 2.3. Flow Networks

In both biological (e.g., vascular) and engineered (e.g., microfluidic) networks, transport of materials is often achieved via fluid flow. Properties of pipes in a flow network, such as their radii, conductance, or capacitance, determine the network's ability to globally transport material from one point to another. While such optimization could be carried out on a computer, many natural systems appear to adjust individual elements based on local feedback. For example, flow might couple to the mechanical properties of the pipes, constricting or expanding them to control flow conductance and pipe capacitance locally. Such adaptation might allow the network to control the distribution of cargo. For example, in *Physarum polycephalum*, the thickness of tubes controls the shape of the organism and allows it to move, forage and memorize features of its surroundings (11, 45, 82; **Figure 2*d***).

Similarly, adaptive processes in other natural flow networks such as those in leaves and vasculature (83, 84) are thought to work through local rules, as there is no central controller. Theoretical work on such flow networks has explored the ability to learn different behaviors through local rules (**Figure 1*c***, subpanel *iii*), including ML-like classification of stimuli (6, 85).

## 2.4. Neuromorphic Computing Through Networks

A related but distinct thread of research is neuromorphic computation through network architectures, which typically use solid-state, electronic, or optical elements to construct networks, inspired by neuronal or computational learning (29, 30). Promising implementations include crossbar arrays (86), spintronic tunnel junctions (87) and phase change materials (PCMs) in photonic systems (**Figure 2*e***). In these systems, elements might adapt their resistance (e.g., memristors, potentiometers) as part of learning.

While past neuromorphic systems tried to implement nonlocal gradient descent (88) or biological learning rules (89) using complex elements, a new generation of neuromorphic systems implement simpler local learning rules based on contrastive learning (90, 91), allowing for regression and classification (27, 28; **Figure 2*f***). Neuromorphic computing seeks to compete in traditional ML domain on energy and robustness (92, 93). In contrast, the spirit of this review is to explore learning as a metaphor for how the response of atypical disordered physical systems to physical stimuli can change in functional ways; we do not seek to solve problems in the traditional ML domain with physical systems.

A recent new approach goes beyond engineered neural network-like architectures, taking the power of backpropagation to exploit the dynamics of complex physical systems. In this approach (47), physical parameters of a mechanical, electronic or optical system are updated by a backpropagation algorithm that observes the response of the system to physical stimuli. Such training effectively creates "deep physical networks," i.e., physical systems, capable of solving classification problems (**Figure 2*g***). Computer-aided backpropagation in Reference 47 is not a focus of this review; we focus on training that is potentially implementable through local rules in situ by physical dynamics, instead of computer-aided backpropagation.

On the other hand, there are similarities. The goal of the molecular and mechanical systems reviewed here and the approach of Reference 47 is to exploit the intrinsic dynamics of physical systems for learning and computation on physical stimuli, rather than force physical systems to mimic neural network architectures element by element.

### 2.5. Reservoir Computing: Training Computational Interfaces

Physical computation approaches often use a physical system as a computational resource, and opt to only train an interface with that system. In reservoir computing, a complex dynamical network takes an external input and performs a generic high dimensional computation. A filter is then trained using a computer on the output of the system, such that a desired result is obtained (34). Different types of physical systems were used in this way, including electronic, fluidic, mechanical, and biological systems, for a broad range of regression and classification problems.

## 3. PHYSICAL LEARNING RULES

In learning theory, learning problems are typically divided into conceptually distinct classes, unsupervised and supervised learning. A similar distinction is worth making in the materials context since the classes correspond to potentially different applications with different physical training protocols.

### 3.1. Unsupervised Learning

In unsupervised learning, the learning d.o.f are modified directly as a response to observed signals. A completely unsupervised physical learning rule is of the form,

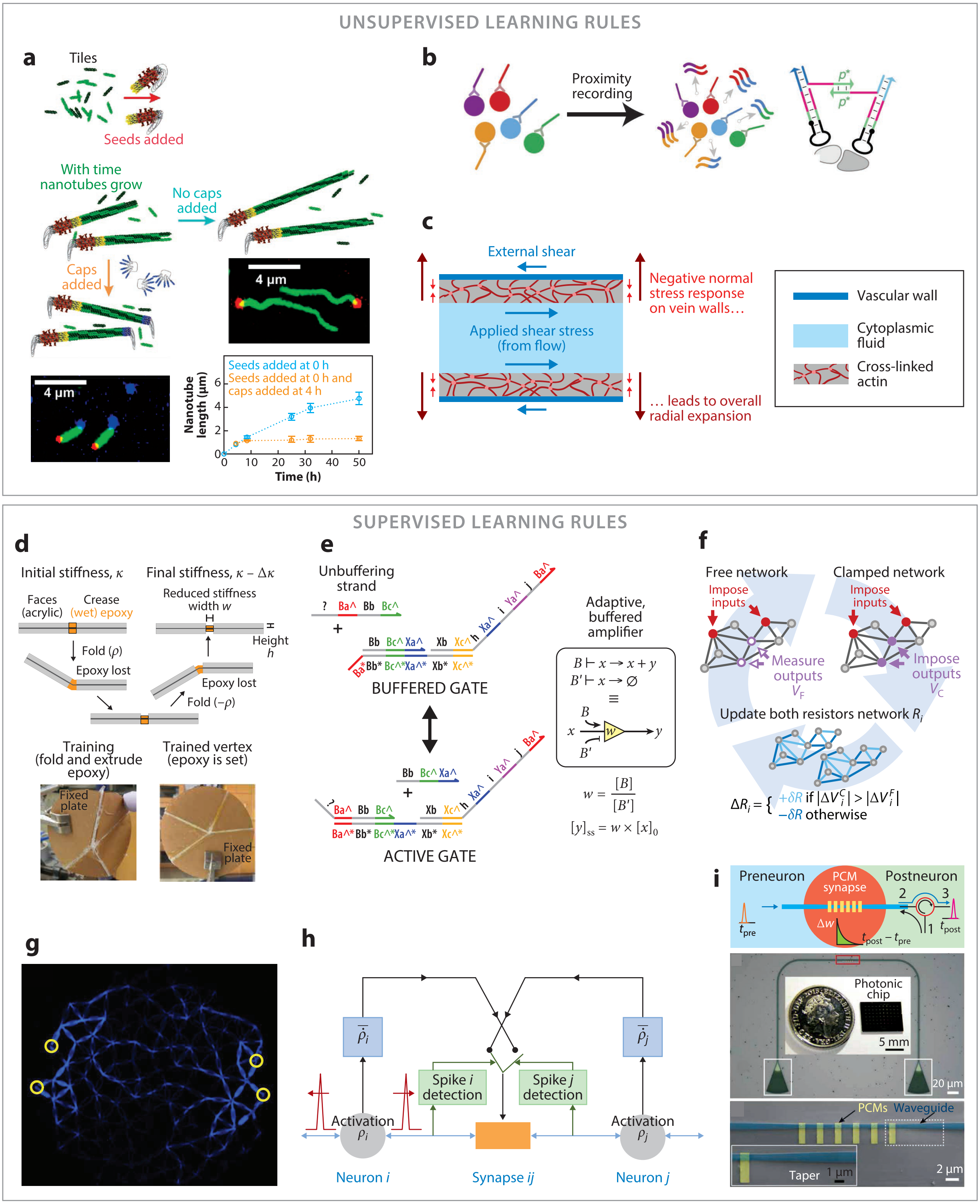
$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = h(\mathbf{s}(f;\{w_j\})) \quad \text{(unsupervised learning)}, \tag{3.}$$

where $\mathbf{s}(f;\{w_j\})$ is the configuration the system adopts in response to an external stimulus $f$, assuming material parameters $\{w_j\}$. $h$ is an arbitrary nonlinear function. Here, learning d.o.f $w_i$ adjust themselves based on the response $\mathbf{s}(f;\{w_j\})$. A paradigmatic example of such learning is directed aging in elastic systems. For example, elastic networks have been trained to be auxetic (4) by holding materials in the auxetic configuration as strained bonds soften according to a learning rule (Equation 3). In this case, aging reduces the energy of a system held at its desired state $\frac{\mathrm{d}w_i}{\mathrm{d}t} \sim -\frac{\partial}{\partial w_i}E(\tilde{\mathbf{s}}; w_i)$, where $E(\tilde{\mathbf{s}})$ is the energy of a desired state $\tilde{\mathbf{s}}$. Note that as the energy of elastic networks is a sum over individual bonds $i$, this is a local rule, where every bond $i$ changes according to the stress (energy) it carries. Similarly, at the molecular scale, stabilizing elements can grow between parts of a material that stabilize configurations the material is held in. For example, learning rules have been demonstrated experimentally in hydrogels using polymerases (67) and DNA nanotubes (68, 94; **Figure 3*a***) and theoretically modeled in Reference 3 to continually store multiple configurations in one mechanical system, without overriding previous ones, and retrieve configurations based on partial prompts.

Unsupervised learning can create new molecular interactions (**Figure 3*b***). For example, proximity-based ligation (57, 58) can create interaction-mediating molecules for species $i, j$ by creating a ligated $i - j$ molecule; further, ligation is naturally localized in space and time by mass-action kinetics, resulting in a Hebbian-like learning rule:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} \sim s_i(x,t)s_j(x,t) \quad \text{(molecular Hebbian learning)}, \tag{4.}$$

where $w_{ij}$ is the interaction strength between molecular species $i$ and $j$ with concentrations $s_i(x,y)$, $s_j(x,t)$. Thus, spatial and temporal proximity during learning leads to changes in stronger or weaker binding interaction between different species of molecules. Equivalent mechanisms include activation of inactive particles, e.g., through strand displacement (57) or phosphorylation. Such processes were shown to allow (2, 63, 97) learning multiple self-assembly behaviors and recognize patterns through nucleation (see **Figure 4*d***).



*(Caption appears on following page)*

**Figure 3** (*Figure appears on preceding page*)

Local unsupervised and supervised learning rules in various physical substrates. (*a*) DNA nanotubes self-assemble to reinforce connections between "seeds" based on their spatial proximity. Panel adapted with permission from Reference 94; copyright 2017 American Chemical Society. (*b*) Creation of interaction-mediating molecules based on spatiotemporal proximity of other molecular species. Panel adapted from Reference 57 (CC BY 4.0). (*c*) *Physarum polycephalum* adapts the radius of vascular tubes according to the local flow shear stress. Panel provided by K. Alim and S. Marbach. (*d*) Epoxy-filled creases in sheets extrude epoxy in response to folding strain during the epoxy setting (i.e., training) period; consequently, final crease stiffnesses about the flat state depend on the history of folding. (*e*) Buffered DNA strand displacement circuit feeds back into the concentration of reactants to implement gradient descent in supervised learning. Panel adapted with permission from Reference 71; copyright 2016 American Chemical Society. (*f*) Two coupled electronic circuits, a free and a clamped network, that change the resistance of variable resistors in response to local voltage drops (27). (*g*) Local contrastive learning applied manually by pruning bonds in experimental photoelastic networks with visual stress measurements. Panel adapted with permission from Reference 95; copyright 2021 American Physical Society. (*h*) A spiking electronic circuit implementing a contrastive learning rule to modify the synaptic weights of a Hopfield-like network according to temporal spiking patterns. Panel adapted from Reference 91 (CC BY 4.0). (*i*) Optical transmissivity of elements in an optical neural network changes according to STDP, by feeding back optical signals into wave guides with phase change materials. Panel adapted from Reference 96 (CC BY 4.0). Abbreviations: PCM, phase change material; STDP, spike-timing-dependent plasticity.

Adaptation in flow networks has also been modeled (11, 84) with unsupervised rules that only depend on the flow through edges, proportional to a pressure drop $s_j - s_k$:

$$\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} \sim h(s_i - s_j) \quad \text{(flux-based rule)}, \qquad 5.$$

where $w_{ij}$ is the conductance of an edge connecting nodes $i$ and $j$. In *P. polycephalum*, the stimulus $f$ might correspond to food sources. Edges that exhibit low effective conductance are pruned, as observed in simulations and experiments (11; **Figure 3*c***).

## 3.2. Supervised Learning

In ML, supervised learning involves labeled training examples, e.g., labeled images of cats and dogs. In the context of physical learning, supervised learning corresponds to training a material to show desired specific responses for all input stimuli $f$. A typical supervised learning task is classification [e.g., the Iris data set (98)], where a system is trained to produce a specific response $s_A(f)$ for all stimuli $f \in F_A$ in class $F_A$ and a different response $s_B(f)$ for stimuli $f \in F_B$ in class $F_B$. Here, $F_A$ and $F_B$ are user-specified sets of stimuli and the challenge is two-fold: the physical learning process must identify correlations or features that separate the stimuli in $F_A$ and $F_B$ and, further, must result in a material that responds only to these features in a specified way (**Figure 4*a***).

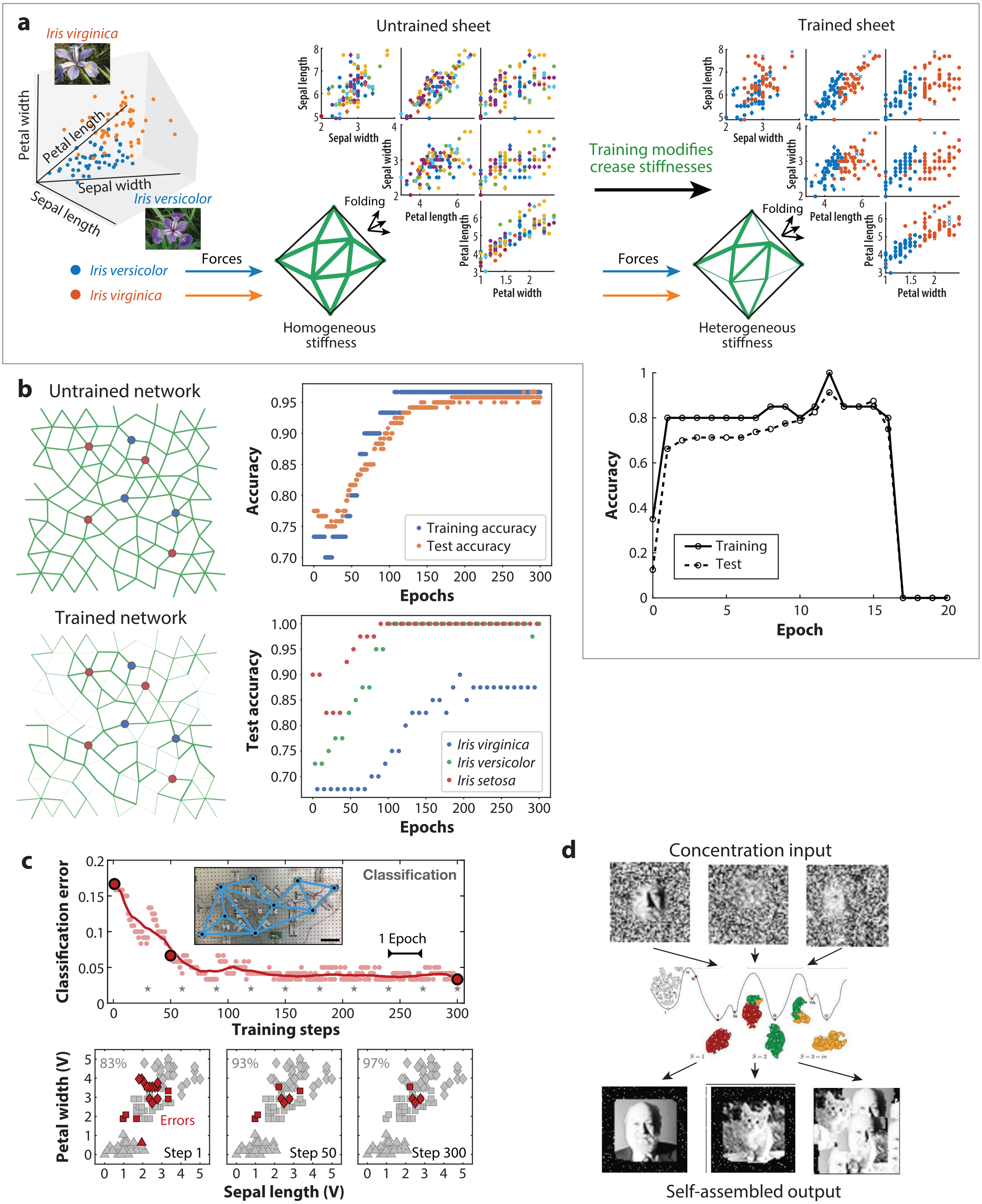
As in ML, one can evaluate learning performance by a cost function $C(\{w_i\})$ that is a measure of the number of stimuli $f$ for which the system (with given $\{w_i\}$) evokes an incorrect response. Then, the goal of physical supervised learning is for the material to naturally adapt $w_i$ such that the cost function $C$ is minimized.

To achieve such a goal in physical systems, one needs a "supervisor" who makes the learning process contingent in some way on the response $\mathbf{s}(f)$ to stimuli $f$ being "right" or "wrong" (as defined by the cost function).

The simplest form of such supervision is a "thumbs-up-thumbs-down" feedback in which the supervisor only decides the sign of the autonomous unsupervised rules described earlier:

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = \begin{cases} -h(\mathbf{s}(f;\{w_i\})) & \text{if } \mathbf{s}(f) \text{ is correct} \\ +h(\mathbf{s}(f;\{w_i\})) & \text{otherwise} \end{cases} \quad \text{(thumbs-up-thumbs-down rule)}. \qquad 6.$$

Such a rule was shown to be powerful enough to learn and classify subtle force correlations applied to disordered mechanical systems (5). In this theoretical study, spatial force patterns belonging to one of two sets $F_A$, $F_B$ are applied to a disordered creased sheet; a supervisor decides whether the sheet is folded into a desired structure $s_A$ or $s_B$ for stimuli in $F_A$ or $F_B$ respectively.



(*Caption appears on following page*)

**Figure 4** (*Figure appears on preceding page*)

Training physical learning machines for machine-learning-inspired tasks by mapping complex data sets to physical stimuli. (*a*) Data from the Iris flowers (98) data set is mapped to physical stimuli (here, force patterns) that are then used to train a creased thin sheet to show distinct responses for distinct classes of stimuli. Here, the sheet adopts a heterogeneous crease stiffness profile through a local learning rule and, eventually, correctly classifies the input stimuli through folding responses (5). (*b*,*c*) Similar classifications can be learned in flow and electric networks by mapping the Iris data to input pressures or voltages at some nodes and reading out the pressures or voltages at some other output nodes, in (*b*) simulations and (*c*) experiments (27). (*d*) 2,500-pixel images were mapped to concentration patterns of 2,500 molecular species (grayscale value of pixel $i$ = molecular concentration $a_i$). Molecular interactions were trained by a Hebbian rule, so the molecular system classifies different (potentially distorted) stimuli by showing different self-assembly behaviors in the nonequilibrium nucleation-dominated regime of self-assembly (63).

If the folded structure is correct for the stimuli, the supervisor lets creases soften according to their strain, say, by immersing the folded structure in one chemical environment. Otherwise, the supervisor immerses the folded structure in a different environment that *stiffens* creases based on strain. A simplified version of this local rule was realized by letting a liquid glue flow out of creases that significantly fold in a desired configuration in creased sheets (50a; see **Figure 3*d***).

Similar supervised rules can be implemented in molecular systems, e.g., by controlled strand displacement (71; **Figure 3*e***). Through such mechanisms, e.g., the unsupervised molecular Hebbian learning rule in Equation 4 can be modified so that interactions between colocalized molecules are either strengthened or weakened depending on the judgement of a supervisor.

A more powerful supervised learning framework, requiring greater supervision but promising better results, is inspired by contrastive Hebbian learning (99). In contrastive learning, a system is trained by observing the contrast between its current responses to an input $\mathbf{s}(f)$ and a nudged "improved" response $\hat{\mathbf{s}}(f)$. This nudging takes the form of additional (weak) forces or constraints, applied by the supervisor on the system, such that it better represents the desired configuration (lowering the cost function). For physical systems that are defined by an energy function $E(\mathbf{s}; w_i)$, a simple contrastive learning rule takes the form,

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} \sim \frac{\partial}{\partial w_i}[E(\mathbf{s}(f)) - E(\hat{\mathbf{s}}(f))] \quad \text{(contrastive learning).} \qquad 7.$$

Although this approach is frequently used to train computational models (100), it was shown to give rise to local, physically realizable learning rules (25). Furthermore, equilibrium propagation (101), a version of contrastive learning, well approximates the gradient of a global cost function, suggesting that physical local learning is not fundamentally limited in performance compared to global computational ML. Contrastive learning and its derivative approaches were shown to successfully train (in silico) various physical models, including Hopfield nets (101), flow networks (90, 102; **Figure 4*b***), mechanical spring networks (6), cell assemblies and photonic neural networks (103). Early experimental systems implementing contrastive learning in linear resistor and elastic networks (**Figure 3*f*,*g***) demonstrated the success of this approach (27, 28, 95), e.g., in classifying the Iris data set (**Figure 4*c***). These experiments further hint at the potential scalability of physical contrastive learning to large networks and complex sets of tasks.

A plausible generalization to systems without an energy function (i.e., nonsymmetric dynamical systems) but still at steady state is given by spike-timing-dependent plasticity (STDP; 24, 104): $\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} \sim h(s_j)\dot{s}_i$, where $w_{ij}$ is the strength of a synapse (coupling) connecting neurons $j$ to $i$, and $h(\cdot)$ a nonlinear function. This learning rule was shown computationally to enable learning in nonsymmetric spiking recurrent neural networks (**Figure 3*h***) by continuously evaluating the activations of presynaptic and postsynaptic nodes (91). Such rules have been implemented in physical memristive substrates (105, 106); e.g., optical transmissivity of chalcogenide phase change materials (PCMs) changes as a function of the number of pulses applied to them, affording an approximate implementation of STDP in optical synapses (96; **Figure 3*i***). Going beyond training static responses in networks, contrastive learning can be applied to systems with time varying inputs based on a Lagrangian formulation of equilibrium propagation (26).

# 4. CHALLENGES IN IMPLEMENTING PHYSICAL LEARNING

## 4.1. Physical Implementation of Supervision

The principal difficulty in realizing physical learning machines is the requirement that the underlying physics allows for a useful local learning rule.

Many unsupervised rules explored here may be implemented by inevitable processes in natural systems. For example, directed aging in elastic systems (4, 39, 40) exploits inevitable aging processes in many materials. Similarly, growth-based rules in molecular systems naturally follow local geometry (3) while flow networks in many natural systems can expand or constrict according to local flow properties (11). Proximity-based ligation (57) allows for molecular interactions to increase based on spatial or temporal proximity, e.g., enabling learning of self-assembly behaviors by simply colocalizing particles in the desired arrangement during a training period (2, 97).

Supervised learning places greater demands on the material since it requires the same physical system to employ a learning rule with either sign, e.g., strengthening or weakening interactions, depending on context. For example, thumbs-up-thumbs-down supervised learning in creased sheets (5) required the same elastic material to be capable of stiffening (thumbs down) or softening (thumbs up) due to strain, based on a thumbs-up-or-down "signal" from the supervisor. Such sign switching might be achieved by the supervisor placing the system in different chemical environments (or at different temperatures; 4) based on whether the system shows the right or wrong response to a stimulus.

Greater supervision like contrastive rules requires more to realize its promise: in addition to the sign problem, contrastive rules require two copies of the system and the ability to change learning d.o.f $w_i$ based on the quantitative difference between the systems. While such duplication might be possible in some cases (27), a more broadly relevant implementation could apply the free and clamped conditions sequentially (101), if learning elements have a memory component. Another implementation might involve physical signals of multiple modalities to obtain the contrastive signal (107); e.g., in flow networks, tracer particles could be injected at the output and advected by the flow, carrying the error signal information to the learning d.o.f (85). Finally, approximations of idealized theoretical learning rules might suffice in real materials; e.g., correctly getting just the sign of change in $w_i$ based on change in energy might be sufficient for contrastive learning $w_i$ (27, 108).

## 4.2. Locality in Solid and Liquid-Like Systems

While physical learning rules need to be local, the nature of the locality requirement varies across systems. Solid-like systems, e.g., elastic networks and sheets and flow networks, typically have fixed neighborhood geometry. Each learning element $w_i$ has a fixed location in space and time $x, t$ and locality implies that $w_i$ can only change based on the state of the system $\mathbf{s}[x, t]$ in the vicinity.

Liquid-like systems, like molecular systems, have no fixed neighbors in space, since components typically diffuse freely like in molecular systems or rearrange in a more limited manner as in jammed packings (41). While training might seem difficult in such liquid-like systems without fixed neighbors, there are two ways forward. In molecular systems with many species of components (65, 66, 109), the learning d.o.f $w_{ij}$ are interactions $J_{ij}$ between species $i$ and $j$. The locality constraint dictates that $J_{ij}$ only change based on *transient* spatial or temporal coincidences of molecules $i, j$ as in Equation 4, e.g., as considered in proximity-based ligation and in Hebbian learning of self-assembly (2, 63). Note that the resulting learned interaction graph, encoded by $J_{ij}$, has no real-space interpretation and does not need to be embeddable in 3 dimensions.

A distinct challenge arises in liquid-like systems with only a few species or components, such as jammed sphere packings and actin or microtubule networks. Here, training often relies on the geometric arrangements, e.g., memories stored in cytoskeletal networks under shear (41, 75). The further extension of learning protocols to such liquid-like systems without fixed neighbors is important to broaden the scope of physical learning to all scales.

## 4.3. Exploiting Noise

Compared to computer algorithms, physical systems experience higher noise that might alter how they learn in response to external stimuli (110). Noise can affect both the physical response $\mathbf{s}(f)$ to stimuli $f$ and the consequent change in learning d.o.f $w_i$.

Such inevitable fluctuations in physical systems can be a blessing since several learning tasks require randomness. For example, molecular systems with finite numbers of molecules can naturally act as "chemical Boltzmann machines" needed for the unsupervised learning of probability distributions (72, p. 210). Similarly, Reference 63 shows how the intrinsically stochastic nature of nucleation of crystals can solve complex pattern recognition problems, much the way stochastic local search algorithms solve satisfiability problems on a computer. Noise during physical training might also be beneficial for generalization, as often found with stochastic gradient descent in in silico ML (112). In physical systems, noise improves the robustness of encoded memories in disordered jammed packings (111). Similar properties are likely inherited by other local learning rules in physical systems but this remains to be investigated. If noise levels are too high, potential mitigation strategies include modulating the learning rate and the magnitude of stimuli during training (**Figure 5*a***).

The distributed nature of physical learning also intrinsically makes it robust to other noise sources, like malfunctioning learning elements or damage. Similarly, unlike in silico ML, physical learning is generally asynchronous since different learning d.o.f $w_i$ are updated independently without a central clocked processor, which might be beneficial for discrete learning d.o.f  (28).

## 4.4. Material Constraints: Dynamic Range of Weights $w$, Overtraining

A significant constraint on learning is the dynamic range of the learning d.o.f $w_i$ achievable in real materials without other compromises. For example, interactions in molecular systems can vary over a finite range (relative to $k_B T$) before associations become irreversible. Similarly, in mechanical systems, the range of bond stiffnesses might be limited, e.g., to avoid fracture. Such constraints might be overcome to some extent by choice of materials, such as shape memory polymers (113, 114), hydrogels and poly-carbonates (115, 116). Existing works (2, 5, 102) have found that requirements for tasks studied so far are within experimentally available ranges. Another solution is through architecture; e.g., larger networks with more learning elements, each having a moderate dynamic range, might alleviate the need for larger dynamic ranges.

Some learning d.o.f $w_i$, such as stiffnesses, are constrained to be positive, an issue discussed before for the brain (117). Solutions proposed in ML, like shifting the task to positive values (118), or decomposing negative weights as the difference of two positive weights (119), might have physical analogs as well. For example, positive molecular binding energies can be effectively shifted to negative values by adjusting entropic costs of binding (2).

Another failure mode for learning physical systems is overtraining (5, 120). In materials that lose the variability of the learning d.o.f, overtraining can result in a network that can no longer adapt to new tasks (121).

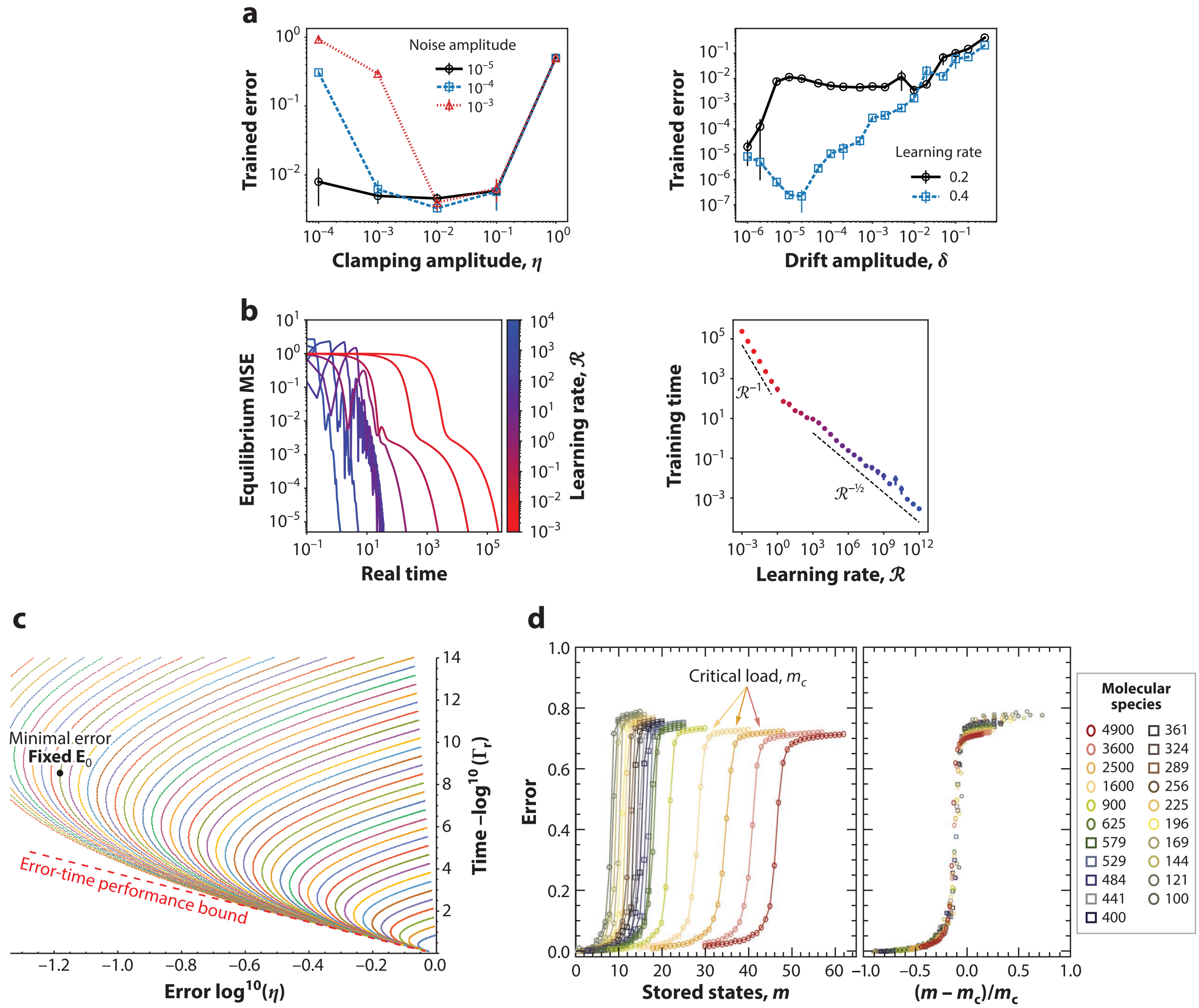


Figure 5

Constraints on physical learning. (*a*) Noise in either the evaluation of the physical or learning d.o.f can impede learning, yet it can be mitigated by controlling physical hyperparameters like the amplitude of clamping correction $\eta$, or the effective learning rate $\alpha$ (6). (*b*) Physical learning can succeed beyond physical equilibrium with minor effect as long as the learning rate does not exceed the physical dynamical rate $\mathcal{R} = 1$. Far beyond equilibrium, underdamped learning oscillations appear, yet the system might still be able to learn (102). (*c*) Speed-accuracy trade-off in molecular systems trained to recognize patterns in molecular concentrations (63); higher temperatures lead to higher accuracy and higher complexity of patterns recognized but come at a price of slower computation through nucleation and growth. (*d*) Learning capacity scales as the square root of the number of distinct molecular species for two-dimensional molecular self-assembly (2). Abbreviation: MSE, mean squared error.

## 5. PHYSICAL MANIFESTATIONS OF LEARNING CONCEPTS

### 5.1. Memory, Learning, and Generalization

A closely related concept to learning is memory. In the materials context, memory has a longer history of research, especially in disordered mechanical systems, polymer melts and glasses (21). Examples include colloidal systems "remembering" the amplitude of prior shear and glasses "remembering" a temperature at which they were aged.

What is the relationship between learning and memory? On one hand, memory is a manifest requirement for learning; the learning d.o.f $w_i$ must encode a memory of past stimuli $f$. On the other hand, successful learning places additional demands on the nature of memory. Learning requires that the memory have a specific physical retrieval mechanism, namely in response to stimuli for which a functional response is sought. Further, learning often allows physical systems to generalize and show the correct response to stimuli never seen before (2, 3, 5, 6, 53), much the way an in silico neural network can generalize to novel cat images in a test set not seen during training. Such generalization requires selective memory for informative features in the stimuli; complete memory of all features of the training stimuli (overfitting) would be counterproductive. In summary, learning requires selective and functional memory with a retrieval mechanism.

## 5.2. Learning Out of Equilibrium

Any learning process is associated with two distinct timescales: (1) a response timescale $\tau_{\text{response}}$ associated with the physical d.o.f responding to stimuli $f$, i.e., mapping inputs to outputs $f \to \mathbf{s}(f, \{w\})$, and (2) a learning timescale $\tau_{\text{learn}}$ associated with updating the learning d.o.f $w_i$ based on the response, i.e., $\frac{\mathrm{d}w_i}{\mathrm{d}t} \sim \frac{1}{\tau_{\text{learn}}} g(\mathbf{s}(f, \{w\}))$. In silico ML has a clear separation between these timescales as the first process is typically fully completed before weights are updated.

In contrast, physical and biological systems might not have a clean separation between $\tau_{\text{learn}}$ and $\tau_{\text{response}}$; learning might effectively occur out-of-equilibrium and not be separable from the physical dynamics of stimuli–response. However, learning can be successful despite the lack of separation of timescales, as investigated for neuronal networks (122–125) and tested recently for physical learning with contrastive rules (102). Even learning rates comparable with the physical response rate ($\tau_{\text{learn}} \sim \tau_{\text{response}}$) have little effect on the trained performance, though higher rates might lead to oscillations (**Figure 5*b***).

## 5.3. Out-of-Equilibrium Effects: Time-Reversal Symmetry, Steady States, and Nonreciprocal Interactions

Beyond the relative rate of learning and physical dynamics discussed above, there are two distinct notions of being out-of-equilibrium relevant for learning: (*a*) the nonequilibrium nature of the learning process and (*b*) the nonequilibrium nature of the underlying physical system (in the absence of learning).

In the learning process, successful learning is almost always out-of-equilibrium because it is irreversible; learning starts with a random set of learning d.o.f and ends with a distinct set of learning d.o.f $w_i$ from which it is likely impossible to reconstruct the initial $w_i$ exactly. As Landauer pointed out (126), such a many-to-one procedure in the space of weights $w_i$ can be interpreted as erasure and must consume free energy. Currently, it is unclear exactly how many-to-one this process is for different materials and tasks. At a more practical level, the learning process is bound to be dissipative in nature. For example, the molecular versions might involve polymerization, ligation, or strand displacement (57, 60, 71), whereas mechanical systems involve dissipative rearrangements in foams and other complex materials (4). The theoretical and practical constraints on learning by dissipation deserve further study.

The underlying physical system can be characterized as being at equilibrium or not without accounting for the learning process itself; many open questions remain on the distinction between the two cases. For example, equilibrium systems with time-reversal symmetry can typically avail themselves of more highly supervised protocols such as the contrastive Hebbian rule and echo backpropagation (99, 103). Some of the systems studied here—e.g., elastic and flow networks—are often studied at steady state and preserve time-reversal symmetry. Consequently, influences can propagate from output to input as easily as input to output, enabling contrastive learning

(Equation 7). In systems without time-reversal symmetry—e.g., molecular systems that are not at steady state or are powered by molecular motors—a perturbation of the output may not affect the input. Such systems can still be trained in an unsupervised way or through the thumbs-up-thumbs-down supervision described earlier.

However, relaxing the constraint of being at equilibrium could potentially help learning. For example, the capacity of pattern recognition tasks in molecular systems may be improved using nonequilibrium nucleation-dominated self-assembly (63) and show associated trade-offs among complexity of pattern recognition, accuracy, and speed of recognition (**Figure 5*c***). Similarly, nonreciprocal interactions (127) should increase the range of learnable behaviors; for example, dynamic phases in Kuramoto-like networks of coupled oscillators could potentially be learned (128). An open question is whether nonequilibrium behaviors are also more learnable; e.g., these systems may be more expressive (97) or show greater degeneracy of parameters for a given behavior.

### 5.4. Dynamic Architectures, Continual Learning, and Forgetting

Neural networks often have a fixed architecture (e.g., 2D convolutional network), with learning merely updating synaptic weights within that architecture. Although some solid-state physical systems might be similarly constrained, physical systems that learn through growth or molecular interactions have a freedom in architecture not usually present in in silico ML. For example, growing networks of nanotubes or grown gels (68, 94) or learned self-assembly (2) can change topology, geometry, and even dimensionality during training. It is possible that learning can construct networks with architectures compatible with performing desired tasks.

Furthermore, physical systems naturally "forget" learned experiences, e.g., through degradation. Such erasure and forgetting can allow physical systems to naturally learn new tasks or functions without exceeding any capacity set by the number of d.o.f (3, 129).

### 5.5. Expressivity, Capacity, and Hidden Nodes

A key property of any learning system is its expressivity, i.e., the complexity of input–output relationships that can be modeled with the available learning d.o.f (130). A system with higher expressivity might also be easier to train. In fact, the success of ML is often attributed to overparameterization (131). A related quantification is capacity (64, 132), which is, e.g., related to the largest number of distinct behaviors that can be learned simultaneously.

As found for neural networks, both expressivity and capacity increase with the number of d.o.f for physical learning (2; **Figure 5*d***). Work in specific systems like learned self-assembly (2, 63, 133) shows that frustrated interactions and dimensionality of these d.o.f can dramatically increase expressivity and capacity. However, we do not yet have broad principles for what controls the expressivity and capacity of physical systems.

Another way to increase expressivity and capacity is the introduction of hidden nodes that do not directly couple to input stimuli or the output behavior, as in restricted Boltzmann machines (134). In the materials context, similar ideas might apply, e.g., applying force patterns to only boundary and not the bulk (135, 136) or mapping inputs only to some molecules (53, 63), might allow learning of more complex behaviors.

## 6. PHYSICAL SIGNATURES OF PAST LEARNING

Physical learning leads to disorder as the learning d.o.f typically become heterogeneous during learning. However, though the usual approach to disordered systems averages over random ensembles (137), the learning process can lead to atypical instances of disorder. Here, we discuss some signatures of learning in physical systems.

We note that many of these learning signatures are analogous to signatures of evolution and evolvability (138). In both cases, functional systems are arrived at through an iterative procedure in parameter space $w_i$. Hence, learning (and evolved) systems may be highly atypical in the context of "random" ensembles typically studied in statistical physics; instead, these systems occupy functional regions accessible by the dynamical learning process. The requirement of accessibility results in features that may be incidental or irrelevant to the task being trained or selected for; such features would not be present in random systems or systems designed by a highly nonlocal algorithm in parameter space. Much like with evolution, these signatures can be used for examination if a natural disordered system is the product of physical learning.

## 6.1. Network Geometry and Topology

Starting from random initial conditions for the learning d.o.f, the learning process often significantly changes d.o.f according to the task. Learning can thus result in special systems with atypical properties not expected of typical members of random ensembles.

**6.1.1. Heterogeneities.** In solid-state systems such as elastic or flow networks, a learning process can leave profound signatures on the real space architecture. For example, like with synaptic weights in a neural network, learning inevitably leads to spatial heterogeneity in local elastic moduli or tube radii (**Figure 6*a***), up to complete pruning of unused edges (**Figure 6*b***). In liquid-state systems, the learning process can create atypical patterns of heterogeneous and promiscuous molecular interactions in chemical space; though a naive analysis might suggest a lack of function due to nonspecificity (139, 140), the learned interactions are actually structured so as to be functional as seen in learned and evolved systems (2, 141). Studying learning in relatively simple physical models may offer new insights on when and why such atypical heterogeneities develop, including potentially network motifs or hierarchical structures found in natural networks (15, 142, 143).

**6.1.2. Adaptability.** Systems alternately trained for multiple incompatible behaviors in an alternating sequence can discover rare but highly adaptive or mutable networks for each task (144–147). Such adaptive networks are able to switch from one behavior to another with far fewer changes in parameters $w_i$ than would be typical for generic designed networks that perform those tasks (148).

## 6.2. Network Dynamics and Energy Landscape

Typical disordered systems with frustrated interactions are characterized by glassy (strongly nonconvex) energy landscapes and associated slow isotropic dynamics. Learning often gives rise to atypical energy landscapes that control the proliferation of nonconvexity and specific dynamical responses through low-energy motions (soft modes).

**6.2.1. Not-so-glassy landscapes.** Generic frustrated many-body systems tend to have glassy landscapes with many randomly placed minima. However, learning can lead to exponentially fewer minima that are not randomly placed. For example, when molecular interactions are learned in a Hebbian way to self-assemble one of many structures, the number of resulting minima is exponentially fewer than naively expected (2). Furthermore, when the number of minima do proliferate near capacity, they are not random; rather, they correspond to chimeric assemblies of structures at other minima, analogous to spurious states in Hopfield associative memory. Similarly, jammed packings (41) can become ultrastable by aging in that state while disordered creased sheets (5, 49, 50, 149), upon training, show exponentially fewer branches at the flat state bifurcation point than expected of a random system.

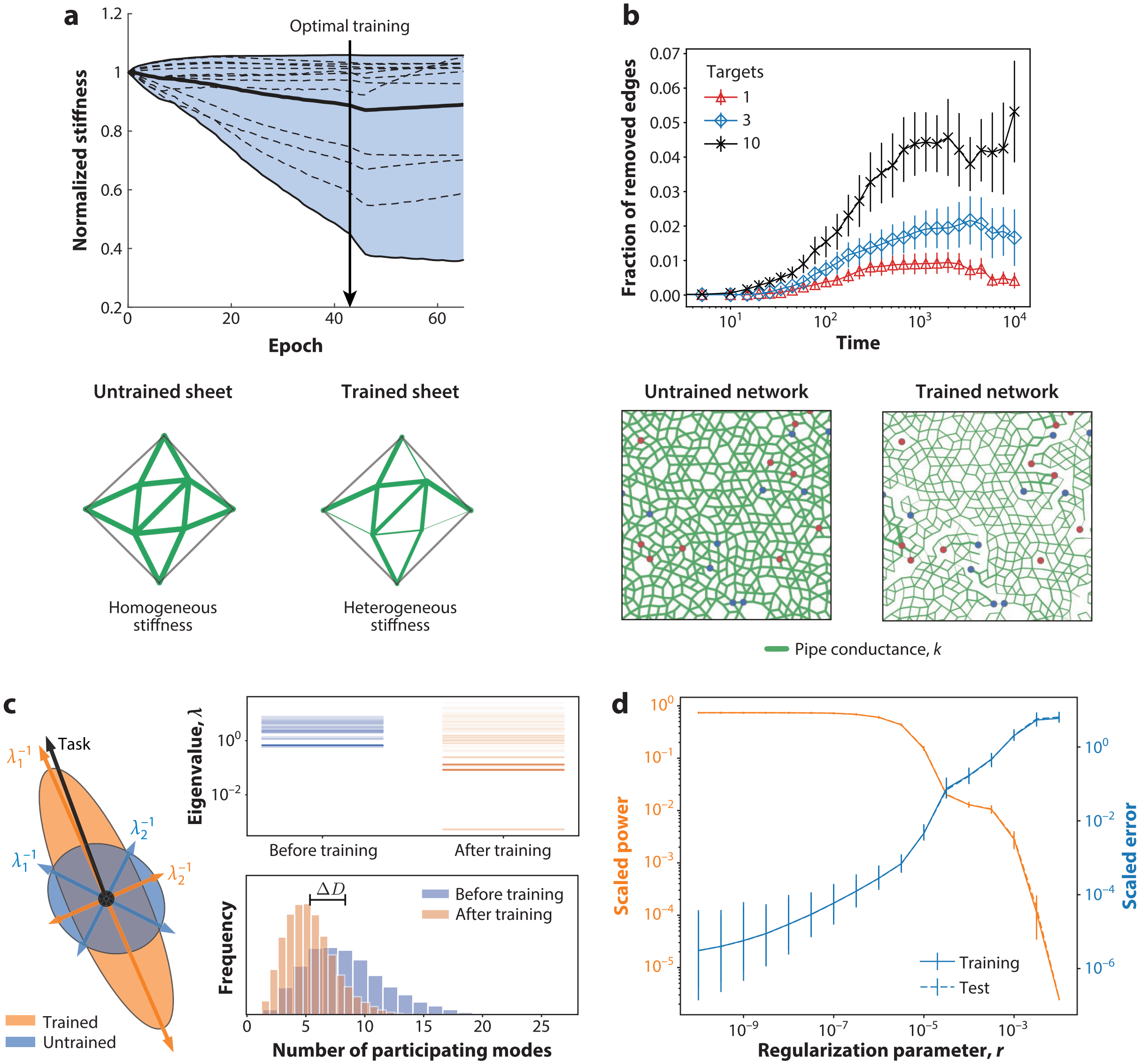


**Figure 6**

Learning creates physical signatures in the substrate beyond the desired functionality. (*a*) Learning induces spatial heterogeneity in the physical substrate, here a self-folding sheet with stiff creases (5). (*b*) System connectivity and topology may significantly change, as learning effectively prunes many unused or counterproductive edges in a network (6). (*c*) Physical learning creates soft modes in a system, so that most forces couple preferentially to these few modes, reducing the effective response dimension $D$ of the system. (*d*) Depending on the difficulty of the task, a learning system may trade off performance for energy. Here, a regression task performed by a flow network shows such a trade-off when a parameter $r$ that regulates power consumption is varied.

**6.2.2. Soft modes.** Trained systems often show soft modes even if training did not explicitly seek such softness (16, 17, 150, 151). Soft modes correspond to normal modes with low-energy eigenvalue in equilibrium systems, or more generally, a small Lyapunov exponent. Consequently, trained systems respond more strongly to random forces than a random system; responses to random forces are often low dimensional along these soft modes (152, 153; **Figure 6*c***). Furthermore, the energy required in flow and resistor networks to actuate desired behaviors is often reduced by supervised learning (**Figure 6*d***).

## 7. CONCLUSION

In this review, we discussed the general notions of autonomous physical learning machines, the physical constraints that affect them, and how some of them can be overcome. Although clearly inspired by neuroscience and ML, we attempted to convey that physical learning is separate and unique: Learning machines are physical, like biological learning networks, but they can learn to solve inverse problems and produce responses that have no analogy in biology or ML. Furthermore, treating learning as a physical phenomenon encourages research on fundamental questions on physically realizable learning models, how learning manifests as a collective behavior in systems with simple constituents, and how learning induces physical changes in these systems. We believe that such questions underlie a fundamental physical theory of learning and that their answers may be independent of the specific details of implementation of the learning machine. Physical learning, both theoretical and especially experimental, requires multidisciplinary approaches. We hope recent and upcoming research in this field will interest not only condensed matter physicists but also computer scientists and neuroscientists as well.

## DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

## ACKNOWLEDGMENTS

We thank Sam Dillavou, Constantine Evans, Martin Falk, Nidhi Pashine, Lulu Qian, Daniel Hexner, Shahar Yosha, Benjamin Scellier, Jason Rocks, Sid Nagel, Andrea Liu, Douglas Durian, Heinrich Jaeger, and Erik Winfree for insightful discussions and suggestions. M.S. would like to acknowledge funding from the National Science Foundation (NSF) via grant DMR-2005749 and from the National Institutes of Health Collaborative Research in Computational Neuroscience (CRCNS) via grant 1R01MH125544-01. This work was primarily supported by the University of Chicago Materials Research Science and Engineering Center, which is funded by the NSF under award number DMR-2011854.

## LITERATURE CITED

1. De Vito E, Rosasco L, Caponnetto A, De Giovannini U, Odone F. 2005. *J. Mach. Learn. Res.* 6:883–904
2. Murugan A, Zeravcic Z, Brenner MP, Leibler S. 2015. *PNAS* 112:54–59
3. Stern M, Pinson MB, Murugan A. 2020. *Phys. Rev. X* 10(3):031044
4. Pashine N, Hexner D, Liu AJ, Nagel SR. 2019. *Sci. Adv.* 5(12):eaax4215
5. Stern M, Arinze C, Perez L, Palmer SE, Murugan A. 2020. *PNAS* 117(26):14843–50
6. Stern M, Hexner D, Rocks JW, Liu AJ. 2021. *Phys. Rev. X* 11(2):021045
7. Pedretti G, Milo V, Ambrogio S, Carboni R, Bianchi S, et al. 2017. *Sci. Rep.* 7:5288
8. Hebb DO. 2005. *The Organization of Behavior: A Neuropsychological Theory*. London: Psychology
9. Micali G, Endres RG. 2016. *Curr. Opin. Microbiol.* 30:8–15
10. Bull MS, Prakash VN, Prakash M. 2021. arXiv:2107.02934
11. Marbach S, Ziethen N, Bastin L, Baeuerle F, Alim K. 2021. bioRxiv: 2021.12.29.474405
12. Ristroph L, Ristroph G, Morozova S, Bergou AJ, Chang S, et al. 2013. *J. R. Soc. Interface* 10(85):20130237
13. Peleg O, Peters JM, Salcedo MK, Mahadevan L. 2018. *Nat. Phys.* 14(12):1193–98
14. Braitenberg V. 1986. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press

15. Kanai R, Komura Y, Shipp S, Friston K. 2015. *Philos. Trans. R. Soc. B: Biol. Sci.* 370(1668):20140169
16. Tlusty T, Libchaber A, Eckmann JP. 2017. *Phys. Rev. X* 7(2):021037
17. Husain K, Murugan A. 2020. *Mol. Biol. Evol.* 37(10):2865–74
18. Wang Y, Chen M, Zhou F, Ma E. 2002. *Nature* 419(6910):912–15
19. Karbasian H, Tekkaya AE. 2010. *J. Mater. Proc. Technol.* 210(15):2103–18
20. Baird DG, Collias DI. 2014. *Polymer Processing: Principles and Design*. Hoboken, NJ: Wiley
21. Keim NC, Paulsen JD, Zeravcic Z, Sastry S, Nagel SR. 2019. *Rev. Mod. Phys.* 91(3):035002
22. Lillicrap TP, Santoro A, Marris L, Akerman CJ, Hinton G. 2020. *Nat. Rev. Neurosci.* 21(6):335–46
23. Nemenman I. 2005. *Neural Comput.* 17(9):2006–33
24. Caporale N, Dan Y. 2008. *Annu. Rev. Neurosci.* 31:25–46
25. Bengio Y, Lee DH, Bornschein J, Mesnard T, Lin Z. 2015. arXiv:1502.04156
26. Scellier B. 2021. *A deep learning theory for neural networks grounded in physics*. PhD Thesis, Univ. Montr., Montreal, Quebec, Can. arXiv:2103.09985
27. Dillavou S, Stern M, Liu AJ, Durian DJ. 2022. *Phys. Rev. Appl.* 18:014040
28. Wycoff JF, Dillavou S, Stern M, Liu AJ, Durian DJ. 2022. *J. Chem. Phys.* 156(14):144903
29. Burr GW, Shelby RM, Sebastian A, Kim S, Kim S, et al. 2017. *Adv. Phys.: X* 2:89–124
30. Marković D, Mizrahi A, Querlioz D, Grollier J. 2020. *Nat. Rev. Phys.* 2(9):499–510
31. Piccinini G. 2015. *Physical Computation: A Mechanistic Account*. Oxford, UK: Oxford Univ. Press
32. Adleman LM. 1994. *Science* 266(5187):1021–24
33. Soloveichik D, Cook M, Winfree E, Bruck J. 2008. *Nat. Comput.* 7(4):615–33
34. Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, et al. 2019. *Neural Netw.* 115:100–23
35. Jaeger H, Haas H. 2004. *Science* 304(5667):78–80
36. Silverberg JL, Evans AA, McLeod L, Hayward RC, Hull T, et al. 2014. *Science* 345(6197):647–50
37. Bertoldi K, Vitelli V, Christensen J, Van Hecke M. 2017. *Nat. Rev. Mater.* 2(11):17066
38. Rocks JW, Pashine N, Bischofberger I, Goodrich CP, Liu AJ, Nagel SR. 2017. *PNAS* 114(10):2520–25
39. Hexner D, Pashine N, Liu AJ, Nagel SR. 2020. *Phys. Rev. Res.* 2(4):043231
40. Hexner D, Liu AJ, Nagel SR. 2020. *PNAS* 117(50):31690–95
41. Hagh VF, Nagel SR, Liu AJ, Manning ML, Corwin EI. 2022. *PNAS* 119(19):e2117622119
42. Zhong W, Schwab DJ, Murugan A. 2017. *J. Stat. Phys.* 167(3):806–26
43. Evans CG, O'Brien J, Winfree E, Murugan A. 2022. arxiv:2207.06399
44. Wei B, Dai M, Yin P. 2012. *Nature* 485:623–26
45. Kramar M, Alim K. 2021. *PNAS* 118(10):e2007815118
46. Lin X, Rivenson Y, Yardimci NT, Veli M, Luo Y, et al. 2018. *Science* 361(6406):1004–8
47. Wright LG, Onodera T, Stein MM, Wang T, Schachter DT, et al. 2022. *Nature* 601(7894):549–55
48. Dudte LH, Vouga E, Tachi T, Mahadevan L. 2016. *Nat. Mater.* 15(5):583–88
49. Stern M, Jayaram V, Murugan A. 2018. *Nat. Commun.* 9:4303
50. Stern M, Pinson MB, Murugan A. 2017. *Phys. Rev. X* 7(4):041070

50a. Arinze C, Stern M, Nagel SR, Murugan A. 2022. arXiv:2206.08886

51. Bray D. 1995. *Nature* 376(6538):307–12
52. Winfree E. 1998. *Algorithmic self-assembly of DNA*. PhD Thesis, Calif. Inst. Technol., Pasadena, CA
53. Cherry KM, Qian L. 2018. *Nature* 559(7714):370–76
54. Zeravcic Z, Manoharan VN, Brenner MP. 2017. *Rev. Mod. Phys.* 89(3):031001
55. Biffi S, Cerbino R, Bomboi F, Paraboschi EM, Asselta R, et al. 2013. *PNAS* 110(39):15633–37
56. Needleman D, Dogic Z. 2017. *Nat. Rev. Mater.* 2(9):17048
57. Schaus TE, Woo S, Xuan F, Chen X, Yin P. 2017. *Nat. Commun.* 8:696
58. Fredriksson S, Gullberg M, Jarvius J, Olsson C, Pietras K, et al. 2002. *Nat. Biotechnol.* 20(5):473–77
59. Lee JH, Lee SH, Baek C, Chun H, Ryu Jh, et al. 2017. *Biosystems* 158:1–9
60. Baek C, Lee SW, Lee BJ, Kwak DH, Zhang BT. 2019. *Molecules* 24(7):1409
61. Baum EB. 1995. *Science* 268(5210):583–85
62. Mills AP, Yurke B, Platzman PM. 1999. *Biosystems* 52:175–80
63. Zhong W, Schwab DJ, Murugan A. 2017. *J. Stat. Phys.* 167(3):806–26
64. Hopfield JJ. 1982. *PNAS* 79(8):2554–58

65. Jacobs WM. 2021. *Phys. Rev. Lett.* 126(25):258101
66. Shrinivas K, Brenner MP. 2021. *PNAS* 118(45):e2108551118
67. Lee JB, Peng S, Yang D, Roh YH, Funabashi H, et al. 2012. *Nat. Nanotechnol.* 7(12):816–20
68. Mohammed AM, Sulc P, Zenk J, Schulman R. 2017. *Nat. Nanotechnol.* 12(4):312–16
69. Schulman R, Yurke B, Winfree E. 2012. *PNAS* 109(17):6405–10
70. Gandhi N, Ashkenasy G, Tannenbaum E. 2007. *J. Theor. Biol.* 249:58–66
71. Lakin MR, Stefanovic D. 2016. *ACS Synthet. Biol.* 5(8):885–97
72. Poole W, Ortiz-Muñoz A, Behera A, Jones NS, Ouldridge TE, et al. 2017. In *DNA Computing and Molecular Programming. DNA 23. Lecture Notes in Computer Science*, Vol. 10467, ed. R Brijder, L Qian, pp. 210–31. Cham, Switz.: Springer
73. Tayar AM, Hagan MF, Dogic Z. 2021. *PNAS* 118(30):e2102873118
74. Saha S, Nagy TL, Weiner OD. 2018. *Philos. Trans. R. Soc. B Biol. Sci.* 373(1747):20170145
75. Majumdar S, Foucard LC, Levine AJ, Gardel ML. 2018. *Soft Matter* 14(11):2052–58
76. Li S, Batra R, Brown D, Chang HD, Ranganathan N, et al. 2019. *Nature* 567(7748):361–65
77. Cichos F, Gustavsson K, Mehlig B, Volpe G. 2020. *Nat. Mach. Intel.* 2(2):94–103
78. Falk MJ, Alizadehyazdi V, Jaeger H, Murugan A. 2021. *Phys. Rev. Res.* 3(3):033291
79. Rubenstein M, Cornejo A, Nagpal R. 2014. *Science* 345(6198):795–99
80. Werfel J, Petersen K, Nagpal R. 2014. *Science* 343(6172):754–58
81. Banerjee S, Gardel ML, Schwarz US. 2020. *Annu. Rev. Condens. Matter Phys.* 11:421–39
82. Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, et al. 2010. *Science* 327(5964):439–42
83. Katifori E, Szöllősi GJ, Magnasco MO. 2010. *Phys. Rev. Lett.* 104(4):048704
84. Ronellenfitsch H, Katifori E. 2016. *Phys. Rev. Lett.* 117(13):138301
85. Anisetti VR, Scellier B, Schwarz JM. 2022. arXiv:2203.12098
86. Kim KH, Gaba S, Wheeler D, Cruz-Albrecht JM, Hussain T, et al. 2012. *Nano Lett.* 12:389–95
87. Grollier J, Querlioz D, Camsari K, Everschor-Sitte K, Fukami S, Stiles MD. 2020. *Nat. Electron.* 3(7):360–70
88. Rosenthal E, Greshnikov S, Soudry D, Kvatinsky S. 2016. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1394–97. Piscataway, NJ: IEEE
89. Serrano-Gotarredona T, Masquelier T, Prodromakis T, Indiveri G, Linares-Barranco B. 2013. *Front. Neurosci.* 7:2
90. Kendall J, Pantone R, Manickavasagam K, Bengio Y, Scellier B. 2020. arXiv:2006.01981
91. Martin E, Ernoult M, Laydevant J, Li S, Querlioz D, et al. 2021. *iScience* 24(3):102222
92. Sui X, Wu Q, Liu J, Chen Q, Gu G. 2020. *IEEE Access* 8:70773–83
93. Rajendran B, Sebastian A, Schmuker M, Srinivasa N, Eleftheriou E. 2019. *IEEE Signal Proc. Mag.* 36(6):97–110
94. Agrawal DK, Jiang R, Reinhart S, Mohammed AM, Jorgenson TD, Schulman R. 2017. *ACS Nano* 11(10):9770–79
95. Pashine N. 2021. *Phys. Rev. Mater.* 5(6):065607
96. Cheng Z, Ríos C, Pernice WH, Wright CD, Bhaskaran H. 2017. *Sci. Adv.* 3(9):e1700160
97. Bisker G, England JL. 2018. *PNAS* 115(45):E10531–38
98. Fisher RA. 1936. *Ann. Eugen.* 7(2):179–88
99. Movellan JR. 1991. In *Connectionist Models: Proceedings of the 1990 Summer School*, ed. DS Touretzky, JL Elman, TJ Sejnowski, GE Hinton, pp. 10–17. San Mateo, CA: Morgan Kaufmann Publ.
100. Xie S, Seung HS. 2003. *Neural Comput.* 15(2):441–54
101. Scellier B, Bengio Y. 2017. *Front. Comput. Neurosci.* 11:24
102. Stern M, Dillavou S, Miskin MZ, Durian DJ, Liu AJ. 2022. *Phys. Rev. Res.* 4(2):L022037
103. Lopez-Pastor V, Marquardt F. 2021. arXiv:2103.04992
104. Scellier B, Goyal A, Binas J, Mesnard T, Bengio Y. 2018. arXiv:1808.04873
105. Kim S, Du C, Sheridan P, Ma W, Choi S, Lu WD. 2015. *Nano Lett.* 15(3):2203–11
106. Serb A, Bill J, Khiat A, Berdan R, Legenstein R, Prodromakis T. 2016. *Nat. Commun.* 7:12611
107. Anisetti VR, Kandala A, Scellier B, Schwarz J. 2022. arXiv:2208.08862
108. Bernstein J, Wang YX, Azizzadenesheli K, Anandkumar A. 2018. *Proceed. Mach. Learn. Res.* 80:560–69

109. Murugan A, Zou J, Brenner MP. 2015. *Nat. Commun.* 6:6203
110. Ventriglia F, Di Maio V. 2002. *Biosystems* 67(1–3):287–94
111. Keim NC, Nagel SR. 2011. *Phys. Rev. Lett.* 107:010603
112. Bottou L. 2003. In *Advanced Lectures on Machine Learning*, pp. 146–68. Berlin: Springer
113. McKnight G, Henry C. 2005. In *Smart Structures and Materials 2005: Active Materials: Behavior and Mechanics*, Vol. 5761, ed. WA Armstrong, pp. 119–26. Bellingham, WA: SPIE
114. Lagoudas DC. 2008. *Shape Memory Alloys: Modeling and Engineering Applications*. New York: Springer
115. Henke M, Gerlach G. 2014. *Microsyst. Technol.* 20(4–5):599–606
116. Zhou Y, Duque CM, Santangelo CD, Hayward RC. 2019. *Adv. Funct. Mater.* 29(48):1905273
117. Sorscher B, Mel G, Ganguli S, Ocko S. 2019. *Adv. Neural Inf. Process. Syst.* 32:10003–13
118. Hu M, Strachan JP, Li Z, Grafals EM, Davila N, et al. 2016. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, Texas, June 5–9*, Hewlett Packard Labs, HPE-2016-23. **https://www.labs.hpe.com/techreports/2016/HPE-2016-23.pdf**
119. Wang Z, Li C, Song W, Rao M, Belkin D, et al. 2019. *Nat. Electron.* 2(3):115–24
120. Amari Si, Murata N, Müller KR, Finke M, Yang H. 1995. *Adv. Neural Inf. Process. Syst.* 8:145–51
121. Bhaumik H, Hexner D. 2022. *Phys. Rev. Res.* 4(4):L042044
122. Zucker RS, Regehr WG. 2002. *Annu. Rev. Physiol.* 64:355–405
123. Marom S. 2010. *Prog. Neurobiol.* 90:16–28
124. Ernoult M, Grollier J, Querlioz D, Bengio Y, Scellier B. 2019. *Adv. Neural Inform. Proc. Syst.* 32:7081–91
125. Bartunov S, Santoro A, Richards BA, Marris L, Hinton GE, Lillicrap T. 2018. arXiv:1807.04587
126. Landauer R. 1961. *IBM J. Res. Dev.* 5(3):183–91
127. Fruchart M, Hanai R, Littlewood PB, Vitelli V. 2021. *Nature* 592(7854):363–69
128. Seliger P, Young SC, Tsimring LS. 2002. *Phys. Rev. E* 65(4):041906
129. French RM. 1999. *Trends Cogn. Sci.* 3(4):128–35
130. Bahri Y, Kadmon J, Pennington J, Schoenholz SS, Sohl-Dickstein J, Ganguli S. 2020. *Annu. Rev. Condens. Matter Phys.* 11:501–28
131. Rocks JW, Mehta P. 2022. *Phys. Rev. Res.* 4:013201
132. Hertz J, Krogh A, Palmer RG. 2018. *Introduction to the Theory of Neural Computation*. Boca Raton, FL: CRC
133. Fink TM, Ball RC. 2001. *Phys. Rev. Lett.* 87(19):198103
134. Hinton GE, Osindero S, Teh YW. 2006. *Neural Comput.* 18(7):1527–54
135. Coulais C, Teomy E, De Reus K, Shokef Y, Van Hecke M. 2016. *Nature* 535(7613):529–32
136. Pinson MB, Witten TA. 2016. *J. Phys. Condens. Matter* 28(49):495102
137. Mézard M, Parisi G, Virasoro MA. 1987. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, Vol. 9. Singapore: World Sci.
138. Wagner GP, Altenberg L. 1996. *Evolution* 50(3):967–76
139. Johnson ME, Hummer G. 2011. *PNAS* 108(2):603–8
140. Huntley MH, Murugan A, Brenner MP. 2016. *PNAS* 113(21):5841–46
141. Su CJ, Murugan A, Linton JM, Yeluri A, Bois J, et al. 2022. *Cell Syst.* 13(5):408–25.e12
142. Alon U. 2007. *Nat. Rev. Genet.* 8(6):450–61
143. Modes CD, Magnasco MO, Katifori E. 2016. *Phys. Rev. X* 6(3):031009
144. Kashtan N, Alon U. 2005. *PNAS* 102(39):13773–78
145. Hemery M, Rivoire O. 2015. *Phys. Rev. E* 91(4):042704
146. Bloom JD, Labthavikul ST, Otey CR, Arnold FH. 2006. *PNAS* 103(15):5869–74
147. Murugan A, Jaeger HM. 2019. *MRS Bull.* 44(2):96–105
148. Falk MJ, Wu J, Matthews A, Sachdeva V, Pashine N, et al. 2022. arXiv:2211.02270
149. Pinson MB, Stern M, Carruthers Ferrero A, Witten TA, Chen E, Murugan A. 2017. *Nat. Commun.* 8:15477
150. Yan L, Ravasio R, Brito C, Wyart M. 2018. *Biophys. J.* 114(12):2787–98
151. Recanatesi S, Farrell M, Lajoie G, Deneve S, Rigotti M, Shea-Brown E. 2021. *Nat. Commun.* 12:1417
152. Kaneko K, Tsuda I. 2003. *Chaos: Interdiscip. J. Nonlinear Sci.* 13(3):926–36
153. Transtrum MK, Machta BB, Brown KS, Daniels BC, Myers CR, Sethna JP. 2015. *J. Chem. Phys.* 143:07B201_1