Learning via Mechanosensitivity and Activity in Cytoskeletal Networks

Deb S. Banerjee [1,2] Martin J. Falk [1,3] Margaret L. Gardel [1,3,4,5] Aleksandra M. Walczak [6,1,3] Thierry Mora [6,1,3] and Suriyanarayanan Vaikuntanathan [1,2,*]

[1] *James Franck Institute, University of Chicago, Chicago, Illinois 60637, USA*
[2] *Department of Chemistry, University of Chicago, Chicago, Illinois 60637, USA*
[3] *Department of Physics, University of Chicago, Chicago, Illinois 60637, USA*
[4] *Department of Molecular Genetics and Cell Biology, University of Chicago, Chicago, Illinois 60637, USA*
[5] *Pritzker School for Molecular Engineering, University of Chicago, Chicago, Illinois 60637, USA*
[6] *Laboratoire de Physique de l'École Normale Supérieure, CNRS, Paris Sciences et Lettres University, Sorbonne Université, and Université Paris-Cité, Paris 75005, France*

(Received 2 June 2025; accepted 17 October 2025; published 5 January 2026)

In this work we show how a network inspired by a coarse-grained description of the cytoskeleton can learn—in a contrastive learning framework—from environmental perturbations if it is endowed with mechanosensitive proteins and motors. Our work is a proof of principle for how force-sensitive proteins and molecular motors can form the basis of a general strategy to learn in biological systems without neurons. Our work identifies a minimal biologically plausible learning mechanism and also explores its implications for commonly occurring phenomenology, such as adaptation and homeostasis.

DOI: 10.1103/s7tg-6q4q

I. INTRODUCTION

Single cells in multicellular organisms perform complex tasks and remain functional in changing environments by adapting and maintaining cellular or tissue-level homeostasis. Examples of such phenomenology include the adaptation of force generation in airway smooth muscle cells in response to mechanical perturbation [1] and adaptation of cellular connectivity and cell division orientation in response to tension in developing tissues [2,3]. Such tasks demand sensing and processing of information about their environment and altering chemical or mechanical processes inside the cell to optimize cellular functions. Recent studies in single cells and tissues suggest that retention of memories of cellular morphologies and tuning of cellular features based on local cell environment may help their biological functionality [4–7]. This process is reminiscent of the concept of *learning* wherein properties and behaviors are tuned such that multiple desired input-output relations can be obtained. While mechanisms that enable learning are well studied in the neuronal context, it is not immediately clear if such mechanisms can be supported more broadly in cell mechanics.

In this work, we focus on a simplistic form of learning that aims to achieve a desired mechanical response. Our main results show how biomolecularly plausible force-sensitivities—such as mechanosensitive proteins and force generator motors—allow systems to learn desired input-output force relations. While previous studies [8–11] established energy-based physical learning algorithms for learning desired responses in mechanical networks, a clear understanding of how such learning mechanisms may be realized in biological systems is not present. Our work is built on this new paradigm of *physical learning* [8–11] and adapts recent work [12] to show how learning may be possible with generic biomolecular ingredients present in a cell.

More specifically, we consider a simplified mechanical network caricature that could represent a coarse-grained version of the cellular cytoskeleton with essential regulatory actors—these mimic the various mechanosensitive proteins and motors that interact with the cytoskeleton—and try to understand whether learning may emerge as a result of cell-autonomous mechanochemical interactions between these few key elements. Our main finding is that with particular kinds of mechanosensitivities and rules for network remodelling due to stresses, our mechanical network caricature is able to learn (potentially multiple) input-output relations encoded in environmental signals. Building on and extending work in Ref. [12], we show that our cytoskeleton inspired mechanical networks can perform a version of *contrastive learning*, a framework used routinely in machine learning. Importantly, we show how this learning persists even in cases with biophysical ingredients such as turnover of the constituent elements of the network.

We also advance our analysis to explore if a cell-autonomous process can replace the external environment and if a self-organized physical learning mechanism may emerge. Finally, to elucidate the role of physical learning in cellular functionalities, we explore how a version of our learning mechanisms may be vital for achieving the homeostasis of the mechanical state of the cell, e.g., homeostasis of tension or strain in the cytoskeletal structure.

*Contact author: svaikunt@uchicago.edu

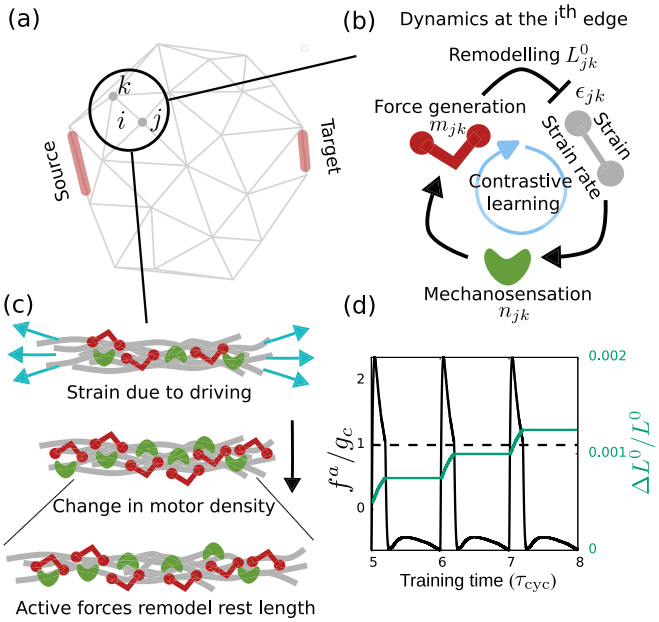

FIG. 1. Physical learning via structural remodeling in a model inspired by the cytoskeleton: (a) Our model is a disordered spring network of nodes connected by edges. The source and target edges are indicated in color. The learning mechanism involves supervised driving at the target edge. (b) Schematic representation of the molecular mechanism of learning showing the dynamic coupling between the network elasticity and agents mimicking molecular motors and mechanosensitive proteins that can bind/unbind from each edge. (c) Schematic showing how driving at the target edge creates local strain and enables an update of the learning degree of freedom (rest length) at any arbitrary edge of the network, via mechanosensation and active force generation. (d) Learning dynamics on one edge of the network. The change in the learning degree of freedom (rest length) at the edge in response to active forces generated by motor dynamics. The dashed line indicates where the active force reaches the threshold active force value $f^a = g_c$ above which rest length changes according to the learning rule.

II. MODEL

We consider a disordered network inspired by the actomyosin cytoskeleton [13–15] [Fig. 1(a)]. The edges of this disordered network can be conceived as cross-linked bundles of actin filaments that undergo structural remodeling based on local quantities such as active stress [16–20]. We shall consider the edges to be Hookean springs (i.e., linear elasticity) given by stiffness (k) and rest length (L^0). In addition, we consider the coupled dynamics of two molecular components: one that mimics an active stress-producing motor and one that mimics a mechanosensitive protein, both of which can bind/unbind from the edges. Although there are many types of molecular motors, such as myosin, kinesin, etc., and different mechanosensitive proteins such as the LIM-domain proteins like zyxin and paxillin, for simplicity, here we consider two generic molecular components to capture the functional role of mechanosensitive LIM-domain proteins [21] and molecular motors. We describe the state of this cytoskeletal network by the position of the nodes $\{\mathbf{r}\}$, the number density of the molecular motor (m) and mechanosensitive protein (n) on the edges [Fig. 1(b)].

The length of the edge between the nodes j and k is given by $L_{jk} = |\boldsymbol{r}_j - \boldsymbol{r}_k|$ and determined by the dynamics of these two node positions. The node position changes according to overdamped dynamics, and the equation for jth node is given by

$$\gamma \dot{\boldsymbol{r}}_j = \sum_k^{nn} -k_{jk}\big(|\boldsymbol{r}_j - \boldsymbol{r}_k| - L^0_{jk}\big)\hat{\boldsymbol{r}}_{jk} + f^a_{jk}\hat{\boldsymbol{r}}_{jk}, \tag{1}$$

where the sum is over the neighbors of node j set by the network connectivity and the unit vector $\hat{\boldsymbol{r}}_{jk}$ denotes the direction from the jth node to kth node. γ is the friction coefficient, k_{jk} and L^0_{jk} are the stiffness and rest length of the edge between the nodes j and k. The active force in the edge due to motor contractility is given by $f^a_{jk} = \xi m_{jk}$ where m_{jk} is the number density of the bound motors and ξ is a positive constant known as the *contractility* or activity parameter. The motor binding-unbinding kinetics is known to be associated with mechanosensitive proteins [22]. The motor dynamics at the edge depends on the number density of mechanosensitive proteins, n_{jk}, and is given by

$$\dot{m}_{jk} = k^0_b + k^1_b n_{jk} - k_u m_{jk}, \tag{2}$$

where k^0_b and k_u are the bare binding and unbinding rates and k^1_b is a constant factor that determines how these proteins promote motor binding. Motivated by the tension-dependent recruitment of mechanosensitive proteins [22–25], we consider force-dependent catch-bond-like dynamics for the number density of bound proteins, where a higher strain rate leads to a lower unbinding rate. The number density of bound proteins in the edge between nodes j and k is given by

$$\dot{n}_{jk} = k_{bn} - k^0_{un} e^{-\beta \dot{\epsilon}_{jk}} n_{jk}, \tag{3}$$

where k_{bn} and k^0_{un} are the bare binding and unbinding rate. Here $\dot{\epsilon}_{jk}$ is the strain rate and the strain ϵ_{jk} in the edge is defined as $\epsilon_{jk} = (L_{jk} - L^0_{jk})$ and β is the coefficient for strain-rate-dependent unbinding. We linearize and rescale the above equations to derive a set of dynamical equations with unitless parameters as detailed in the Appendix A. We present a summary of the parameters and variables in Table I.

We focus on a simple form of learning, namely, learning a desired strain response in mechanical networks [9,11]. Learning, in the current context, occurs when a system changes its internal interactions between its components based on the system's response (output) to an external stimulus (input), such that its response to subsequent stimuli *improves* its ability to reach a desired state [8]. In particular, we want the strain of a given magnitude in one part of the network (i.e., input at source) to create a desired amount of strain in a faraway part of the network (i.e., output at target). To build such an input-output relationship, the learning dynamics or training changes the interactions, that is, the rest length (L^0) and stiffness (k) of the edges based on the local active force (f^a). Thus, the rest length and stiffness of the edges become the learning degree of freedom (LDOF) governed by the learning dynamics or the *learning rule*. In the following, we show how this setup naturally allows our system to learn and remodel in response to environmental stimuli. In particular, adapting Ref. [12], we argue that the mechanosensitive proteins and remodeling by active forces allow our network to implement a

TABLE I. Variables and parameters.

Symbol	Definition	Symbol	Definition
i, j, k	Node indices	$jk, ij, \ldots$	Indices for edge between jth and kth node, etc.
		Network dynamics	
$\boldsymbol{r}_j$	Node position	$\hat{\boldsymbol{r}}_{jk}$	Unit vector toward jth to kth node
k_{jk}	Stiffness of the edge	L_{jk}	Instantaneous length of the edge
L^0_{jk}	Rest length of the edge	ϵ_{jk}	Strain on the edge ($= L_{jk} - L^0_{jk}$)
f^a_{jk}	Active force on the edge	ξ	Contractility or activity parameter
		Motor dynamics	
m_{jk}	Motor number density	m^0_{jk}	Steady-state motor number density
k^0_b	Bare motor binding rate	k^1_b	Mechanosensitive protein-dependent binding rate
k_u	Motor unbinding rate		
		Mechanosensitive protein dynamics	
n_{jk}	Protein number density	n^0_{jk}	Steady-state protein number density
k_{bn}	Protein binding rate	k^0_{un}	Bare protein unbinding rate
β	Coefficient of strain-rate-dependent unbinding		
		Memory kernel	
β_1	Prefactor (mechanosensitivity) ($=k^1_b \beta n_0$)	τ_k	Kernel timescale ($= k_u^{-1}$)
		Training protocol	
τ_f and τ_s	Fast and slow timescale of driving	τ_{cyc}	timescale of a driving cycle ($= \tau_f + \tau_s$)
f^e	Supervised driving force at target	λ	Amplitude of driving ($\lambda \in [0, \lambda_{\max}]$)
$g(x)$	Nonlinear activation function	α	Learning rate
ϵ^*_S	Strain value at source edge (input)	ϵ^*_T	Desired strain at target edge (output)
		Network turnover dynamics	
τ_{sev}	Edge severing timescale	τ_{con}	Timescale of reconnecting of severed edges

contrastive learning protocol—a common learning framework in the machine learning context.

III. FEEDBACK BETWEEN MECHANOSENSITIVE PROTEINS AND MOLECULAR MOTORS CAN PROVIDE A PATHWAY FOR LEARNING

Here, we show how the coupled dynamics of network elasticity (L^0, k), molecular motors (m), and mechanosensitive protein response (n) implement contrastive learning in our cytoskeletal system, thereby allowing the cytoskeleton to encode desired input-outputs maps in response to external mechanical signals.

In standard contrastive learning formulations, there are two relevant states of the system: the "free" state, where input is supplied and the system naturally evolves under its own dynamics; and the "clamped" state, where input is supplied and outputs are additionally pushed to produce the *correct* or desired response. The clamped state therefore encodes the desired input-output map. After cycling between these two states, the learning degrees of freedom change in proportion to the differences in the system during the free and clamped states.

In the cytoskeletal system considered in this work, the standard contrastive update to achieve a desired input-output map by modifying bond rest lengths would be $\Delta L^0_{jk} \propto (\epsilon^{\text{free}}_{jk} - \epsilon^{\text{clamped}}_{jk})$ (see Appendix B for further details). Note that this standard update is nonlocal in time, as it requires the storage and comparison of information from the temporally separated free and clamped states. Thus, it naively requires memory storage of strain values, which is not trivially available in our cytoskeletal network model.

Fortunately, the strain rate dependence of the mechanosensitive protein kinetics [Eq. (3)] enables the local motor dynamics [Eq. (2)] to have a history of local strain, providing a pathway for biologically plausible contrastive learning [12]. Using the linearized form of Eqs. (2) and (3), we can show that the motor dynamics possesses a memory of the local strain in the form

$$\delta m = \int_{-\infty}^{t} \mathcal{K}(t - t')\delta\epsilon(t')dt', \tag{4}$$

where the memory kernel $\mathcal{K}$ is given by

$$\mathcal{K}(t - t') = \beta_1 \left(\delta(t - t') - \frac{1}{\tau_k} e^{-\frac{(t-t')}{\tau_k}} \right), \tag{5}$$

where δm and $\delta\epsilon$ are the variation of motor density and strain around a steady state given by m^0 and ϵ^0 (see Appendix A for details). Using the consistency between the integral form [Eq. (4)] and the motor dynamics described in Eq. (2) we can identify the kernel timescale τ_k and the coefficient β_1 to be k_u^{-1} and $k^1_b \beta n_0$, respectively (see Appendix C for details). It is evident from the memory kernel [Eq. (5)] and the form of the constant $\beta_1 = k^1_b n_0 \beta$ that the strain-rate-dependent

mechanosensitive protein dynamics ($\beta \neq 0$) and its coupling with the motor dynamics ($k_b^1 \neq 0$) are crucial for the local motor dynamics to have the memory of the local strain. We shall use the value of β_1 as a measure of mechanosensitivity later in this study.

These arguments, specialized for now to the linear limit, show how force-sensitive kinetics can provide the memory necessary for contrastive learning. However, we will we show that this learning mechanism works more broadly, even with nonlinear effects and biophysical effects such as turnover, and not just in the linear limit.

IV. DYNAMICS OF TRAINING AND LEARNING

To activate the strain memory described in the previous section, we apply a time-varying periodic external force at the target, shifting the system between the free and the clamped states. The system is taken from the free state to the clamped state very fast over a timescale τ_f and slowly brought back over a longer timescale τ_s (see Fig. 11 of Appendix A). Training involves driving the system through many such cycles, that is, the time duration of one cycle is $\tau_{\rm cyc} = \tau_f + \tau_s$.

The external drive and implicit memory combine to train the system by changing the learning degrees of freedom. Based on the structural remodeling dependent on the active force in the network edges, we consider the LDOF to be the rest lengths (L_{jk}^0) and stiffnesses (k_{jk}) of the edges. These LDOF will dynamically change, dependent on the active forces f_{jk}^a on the edges. First, we shall consider unalterable edge stiffness and only the rest length to be the LDOF, with the dynamics given by

$$\dot{L}_{jk}^0 = \alpha g\big(f_{jk}^a\big), \tag{6}$$

where α is the learning rate parameter and $g(x)$ is a nonlinear activation function. For simplicity, we consider a piece-wise continuous form adapted from Ref. [12] and given by $g(x) = x$ for all $|x| \geqslant g_c$ and 0 otherwise. The learning capability is not crucially dependent on this discontinuous functional form and any suitable smooth nonlinear function can be used to implement activation (see Fig. 12 of Appendix A). The parameter g_c represents a threshold active force above which the edges remodel their rest length. The active force being proportional to the motor density which possesses the implicit memory of strain, the above learning rule results in contrastive update of the LDOF (L_{jk}^0). While we introduce the learning mechanism considering linear elasticity of the network without turnover of edges and linear approximation in the mechanosensitive protein dynamics [Eq. (3)], we shall discuss how the learning mechanism is affected by nonlinearities and network turnover later in this work.

V. LEARNING STRAIN RESPONSE IN A MODEL CYTOSKELETAL NETWORK

Using the dynamics described above, we explore whether the network can learn a specific desired mechanical response. In particular, we consider two edges—one source edge and one target edge [Fig. 1(a)]—and train the network to learn to contract to a specific strain value ϵ_T^* at the target edge when we apply an extension of ϵ_S^* at the source edge. While the chosen edges are peripheral, learning capability does not depend on this choice [11]. To train the network to produce the desired strain ϵ_T^* at the target edge, we first apply the strain ϵ_S^* at the source edge which takes the system into the *free state*. Then, on the target edge, we apply an external supervised force $\boldsymbol{f}^e$ which is proportional to the difference between the strain value at the target edge in the free state ϵ_T and the desired strain value at the target edge ϵ_T^*, and is given by

$$\boldsymbol{f}^e = -\lambda(t)\nabla\big(\tfrac{1}{2}|\epsilon_T - \epsilon_T^*|^2\big), \tag{7}$$

where $\lambda(t)$ controls the temporal dynamics of forcing. It goes from $\lambda = 0$ where the system is in the free state to $\lambda = \lambda_{\rm max}$ where the system is in the *clamped state* over the fast timescale τ_f and returns to $\lambda = 0$ over the slow timescale τ_s (see Fig. 11 of Appendix A). This asymmetry in the timescale of driving (i.e., $\tau_f < \tau_s$) is a hyperparameter which in previous work was shown to be necessary for successful learning [12]. Due to this driving, the force balance equation on the target nodes (nodes connected by the target edge) becomes

$$\gamma\dot{\boldsymbol{r}}_j = \sum_k^{nn} -k_{jk}\big(|\boldsymbol{r}_j - \boldsymbol{r}_k| - L_{jk}^0\big)\hat{\boldsymbol{r}}_{jk} + \xi m_{jk}\hat{\boldsymbol{r}}_{jk} + \boldsymbol{f}^e. \tag{8}$$

Driving at the target edge changes the density of the mechanosensitive proteins and motors and subsequently the strain at other edges in the network [Eq. (2), Eq. (3), and Fig. 1(c)]. As indicated previously, we expect that due to the implicit memory [Eq. (4)], the motor dynamics at any edge between nodes j and k estimates the derivative of the local strain, which in turn has the information of the difference in local strain in the clamped and free states $\sim\epsilon_{jk}^{\rm clamped} - \epsilon_{jk}^{\rm free}$ because of the temporal driving at the target edge which periodically takes the system from the free to clamped state and back (see Appendix B for details). Thus, this information gets fed back in the local active force which is proportional to the bound motor density and the active force-dependent remodeling of rest length facilitates contrastive learning [Fig. 1(d)]. The training error, defined as $|(\epsilon_T - \epsilon_T^*)|/\epsilon_T^*$ shows the effectiveness of the contrastive learning mechanism as it reaches close to zero as the training progresses [Fig. 2(a)]. The spatial distribution of increase and decrease in the rest length values (ΔL^0) over the network shows the trained network with altered interaction that led to this learned mechanical response [Fig. 2(b)]. The trained network shows a distinctly different strain value which is close to the desired value ϵ_T^* at the target edge compared to the un-trained network [Fig. 2(c)]. The learning dynamics remodels the rest lengths of various edges to achieve the desired strain at the target edge. This leads to a trained network that is appropriately pre-strained (i.e., with nonzero edge strain, $\epsilon_{jk} = L_{jk} - L_{jk}^0$, in equilibrium without any source strain) to achieve the desired mechanical response (see Fig. 13 of Appendix A). We repeat the learning problem in randomly chosen target and source edges on the periphery of the network to find the average training error to reduce over time indicating the learning capability to be independent of the particular choice of source and target edges (see Fig. 14 of Appendix A). We do not find any significant effect of the network size on learning capability (see Fig. 14 of Appendix A). The contractility parameter, ξ ($f_{jk}^a = \xi m_{jk}$) gives the extent of activity in the network. We find the learn-

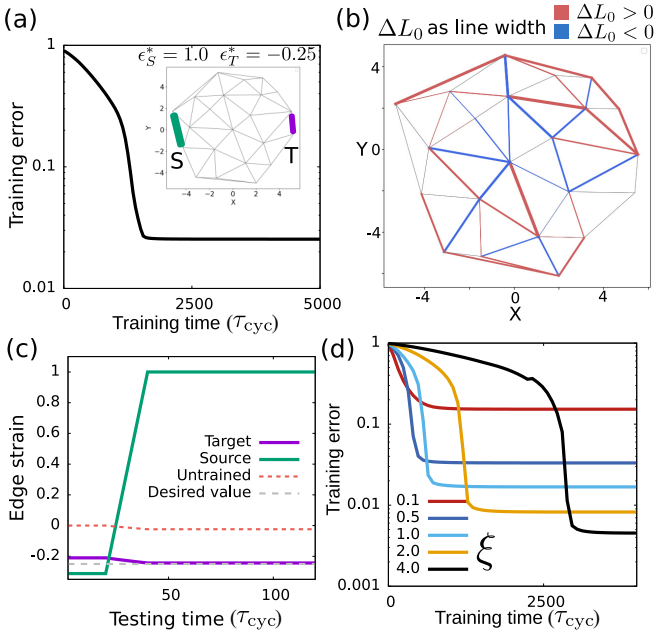

FIG. 2. Learning strain response in cytoskeletal networks by changing rest lengths. (a) Time evolution of training error over many cycles of exposure to desired behavior. (inset) The network at the initial unstrained state. The source and target edges are denoted by "S" and "T," respectively. (b) Trained network reflecting the magnitude of changes in the rest length ($|\Delta L_0|$) of the edges given by the edge thickness. The colors indicate if the rest length increased (red) or decreased (blue) with respect to the original rest length values after training. (c) The strain at the target edge on the application of the source strain on the trained network. The untrained response at the target edge is shown in a red dashed line which is far from the desired target strain (gray dashed line). (d) Training error vs training time at different values of activity given by the contractility parameter ξ. Here the rescaled parameter values are $k = 1$, $\xi = 0.5$, $\beta = 0.1$, $k_u = 0.5$, $\lambda_{\max} = 0.5$, and $\alpha = 1$.

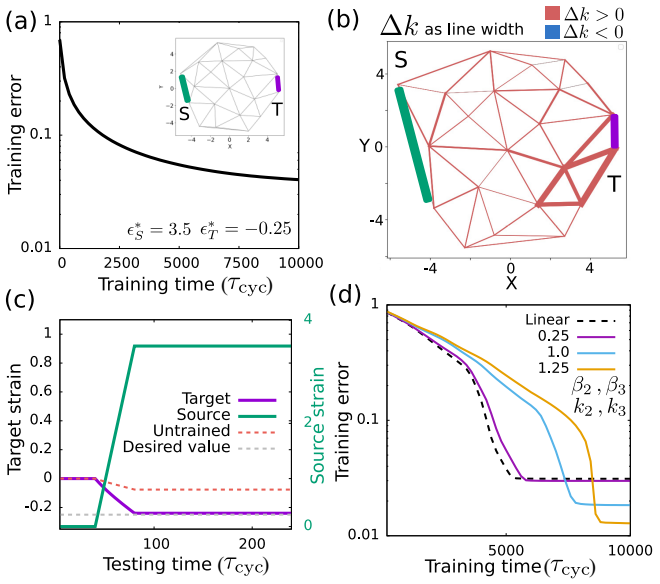

FIG. 3. Learning strain response in cytoskeletal networks by changing stiffness. (a) Time evolution of training error over many cycles of exposure to desired behavior. (b) The trained network topology and edge stiffness change $|\Delta k|$ which is proportional to line width. The source and target edges are denoted by "S" and "T," respectively. (c) The strain at the target edge on the application of the source strain in the trained network. The untrained response at the target edge is shown in a red dashed line, which is far from the desired target strain (gray dashed line). (d) Learning with nonlinear protein dynamics and nonlinear elasticity. Training error vs training time shows that increasing the strength of nonlinearity in both the protein dynamics ($\beta_2, \beta_3 \neq 0$) and network elasticity ($k_2, k_3 \neq 0$) leads to lower training errors. The numbers in the legend indicate these parameter values. We consider a simple case of the same values for the unitless nonlinear coefficients $\beta_2 = \beta_3 = 0.1$, $k_2 = -0.1$, and $k_3 = 0.1$, with other parameters the same as described in the caption of Fig. 2 except $\lambda_{\max} = 0.2$ and $\xi = 0.2$. For panels (a)–(c) the rescaled parameter values are $\xi = 5$, $\beta = 0.1$, $k_u = 0.5$, $\lambda_{\max} = 10$, and $\alpha = 10^3$.

ing to become better with increasing activity (ξ) reaching lower training error values as the activity increases [Fig. 2(d)]. Depending on various hyper-parameter values such as threshold active force (g_c) and timescales of driving (τ_f, τ_s) the learning may slowdown with increasing activity [Fig. 2(d)] which stems from increased unlearning (reversal of the change in rest length) in each cycle during training (see Fig. 15 of Appendix A).

We arrive at a different type of *solution* for the same learning problem (i.e., as described at the beginning of this section) if we consider active force-dependent remodeling of the edge stiffness values while the rest length values remain unaltered. In this case, the dynamics of edge stiffness depends on ϵf^a and for the edge between nodes j and k, is given by

$$\dot{k}_{jk} = -\alpha g\big(\epsilon_{jk} f^a_{jk}\big), \tag{9}$$

where we assume the same piecewise continuous form for the activation function g as discussed before, but change the threshold value g_c appropriately for the current context (see Appendixes A and B). Similar to the previous case, the driving at the target edge affects local motor density and strain values at each edge and that in turn changes the stiffness incrementally in every switching from free to clamped state. The training error reduces as the edges change their stiffness and the network learns the desired target strain value [Fig. 3(a)] and the network develops a spatial pattern of heterogeneous but always positive stiffness value changes over the network, with the largest increase close to the target [Fig. 3(b)]. Here, the trained network achieves the desired mechanical response which is distinctly different from the untrained response [Fig. 3(c)]. In contrast to the previous case, here the trained network does not have any pre-strain (see Fig. 13 of Appendix A). Rather, it reaches the correct solution by changing how the target edge contracts in response to the source edge extension [Fig. 3(c)].

In the above results, we use a linear approximation in the dynamics of the mechanosensitive proteins [Eq. (3)] $e^{-\beta\dot{\epsilon}_{jk}} \simeq 1 - \beta\dot{\epsilon}_{jk}$. We find no significant qualitative or quantitative changes in the learning dynamics (i.e., training error evolution over time) when we consider nonlinearities in the strain-rate-dependent unbinding term in the protein dynamics while learning a desired mechanical response by changing

rest length of the edges, same as described earlier in this section. We consider higher-order terms up to third order (i.e., $e^{-\beta\dot{\epsilon}_{jk}} \simeq 1 - \beta\dot{\epsilon}_{jk} + \frac{\beta^2}{2}\dot{\epsilon}_{jk}^2 - \frac{\beta^3}{6}\dot{\epsilon}_{jk}^3$), which changes the implicit memory the motor dynamics contains compared to the linear case. With the nonlinearity, the motor dynamics becomes dependent on a nonlinear function of strain rate $\delta\dot{m} \sim \beta_1\delta\dot{\epsilon} - \beta_2\delta\dot{\epsilon}^2 + \beta_3\delta\dot{\epsilon}^3$ (see Appendix E for details). Despite this imperfection in implicit memory, which is not simply the derivative of local strain at each edge and does not estimate the difference in local strain between the free and clamped states, the learning capacity of the network remains largely unaffected for moderate strength of the nonlinear terms (see Fig. 16 of Appendix E). This indicates that the linear approximation is adequate for describing the learning mechanism.

Apart from the nonlinearity in the mechanosensitive protein dynamics, one could also consider the nonlinear elastic response of the network, e.g., strain stiffening. Cytoskeletal networks are known to have nonlinear elastic properties [26,27]. We consider a network with nonlinear elasticity where the energy as a function of edge strain (ϵ) is given by: $E_{\rm el} = \frac{1}{2}k\epsilon^2 + \frac{1}{3}k_2\epsilon^3 + \frac{1}{4}k_3\epsilon^4$ to see the effect of nonlinear elasticity on learning. Here, k_2 and k_3 are the elastic constants for higher-order terms in the elastic energy. For a network with nonlinear elasticity, the nonlinearities in the mechanosensitive protein dynamics (up to third order) lead to the correct implicit memory required for the temporal contrastive learning (see Appendix E for details). Thus, nonlinearity in the protein dynamics enables contrastive learning in a network with nonlinear elasticity. Interestingly, we find lower training error with increasing strength of the nonlinearity indicating nonlinear elasticity can aid learning [Fig. 3(d)].

To better understand the nature of the learned response, we examine how the target strain (ϵ_T) varies with the source strain (ϵ_S) before and after training, even at values of ϵ_S for which ϵ_T was not trained. In the untrained state, the behavior is nearly linear, described by a line through the origin (since $\epsilon_T = 0$ when $\epsilon_S = 0$ in a network with no pre-strain). Training the network to produce a desired target strain ϵ_T^* in response to a specific source strain ϵ_S^* modifies this relation differently depending on the LDOF. When training is implemented by changing rest lengths, the ϵ_S–ϵ_T line shifts its intersection to achieve the target, introducing pre-strain [Fig. 4(a)]. By contrast, when training is implemented by changing stiffnesses, the line instead changes its slope [Fig. 4(b)].

We next train the network to perform a linear regression task, where a single target edge is required to follow a set of three desired strain values $\{\epsilon_T^*\}$ in response to a set of three applied source strains $\{\epsilon_S^*\}$. Since training by changing only rest lengths or only stiffness allows the line to shift either its intersection or its slope, but not both, neither mechanism alone can represent a general linear function. However, when both rest length and stiffness are allowed to vary (i.e., both are LDOF), the network can learn an arbitrary linear function with nonzero slope and intercept [Fig. 4(c)]. Finally, for nonlinear regression tasks, we find that training with rest-length alone as LDOF is insufficient, but training with stiffness as LDOF enables the network to capture nonlinear mappings between source and target strain values [Fig. 4(d)]. This nonlinearity of response is of geometric origin [28,29] and results from rotation of the network edges at large values of source strain.

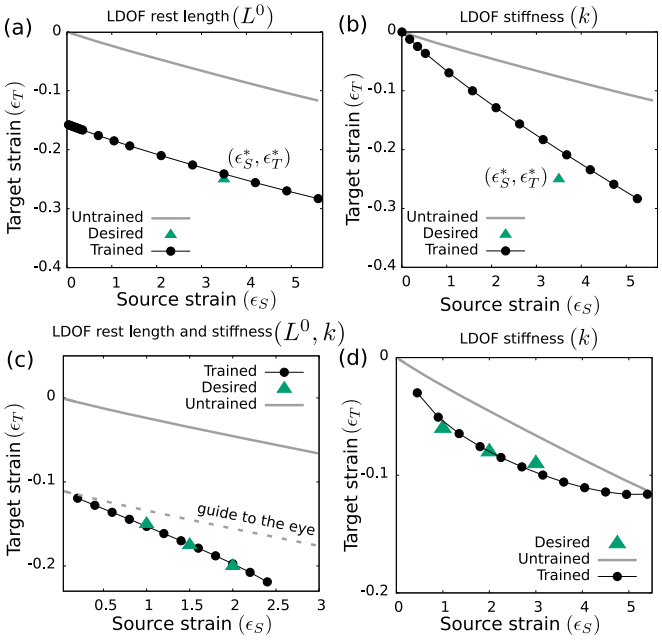

FIG. 4. Learning linear and nonlinear regression. (a) Target strain vs source strain curve after training the network for a desired point $(\epsilon_S^*, \epsilon_T^*)$ by changing the rest length of the edges. The gray line represents the untrained response. (b) Target strain vs source strain curve after training the network for a desired point $(\epsilon_S^*, \epsilon_T^*)$ by changing the edge stiffnesses. (c) Target strain vs source strain curve after training the network for linear regression, i.e., a desired line ($\{\epsilon_S^*, \epsilon_T^*\}$) by changing the edge stiffnesses and rest lengths. (d) Target strain vs source strain curve after training the network for nonlinear regression, i.e., a desired curve ($\{\epsilon_S^*, \epsilon_T^*\}$) by changing the edge stiffnesses. The parameter values are the same as described in the caption of Fig. 2 for learning by changing the rest length of the edges and Fig. 3 for learning by changing the edge stiffnesses.

VI. CLASSIFICATION OF ENVIRONMENTAL SIGNALS

Another classic learning task is solving a classification problem via learning to distinguish between input signals of different types (classes) and produce different responses as output by creating a map between these sets of input and output signals. Living cells continuously interpret noisy environmental signals to decide the future course of their action [30]. For example, both chemical and mechanical environmental signals are known to affect stem cell differentiation decisions [31]. Synthetic chemical reaction networks that can classify chemical signals within the cell have recently been reported [32]. Here we consider a task where the cytoskeletal network is trained to differentiate between different mechanical signals, in particular, different strain gradients in the environment [Fig. 5(a)]. Deciphering such environmental mechanical signals may be important for cellular functions such as cell motility [33] and tissue remodeling during development [34].

We consider increasing and decreasing strain patterns (i.e., positive and negative strain gradients) as two distinct

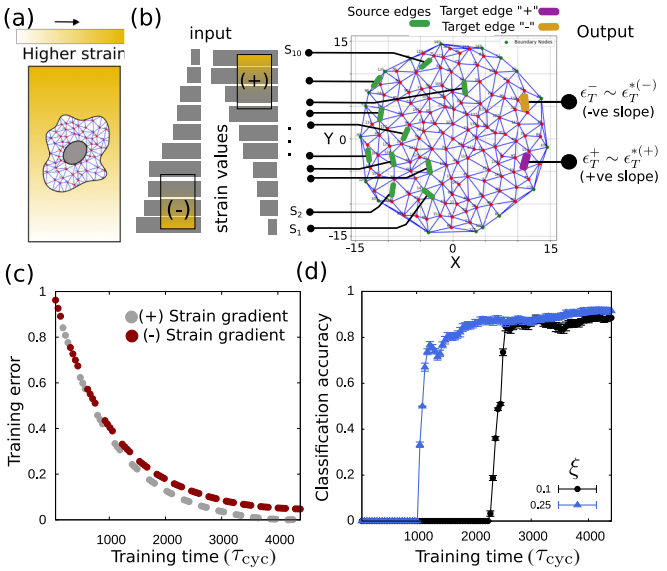

FIG. 5. Classification of environmental signals in cytoskeletal networks. (a) Schematic showing a cell in an environment with a strain gradient. (b) Environmental mechanical signal as the positive and negative gradient of strain. The strain pattern input is imposed on source edges (green) and two target edges are chosen to encode the two classes "+" (purple) and "−" (orange). (c) Training error vs training time for both positive and negative strain gradients as inputs. (d) Classification accuracy increases as the training progresses. The network learns to classify faster with higher activity (given by contractility parameter ξ). Here $\tau_f/\tau_s = 1/10$, $\epsilon_T^{*(+)} = -\epsilon_T^{*(-)} = 0.25$, $\mathcal{E}_{\rm tol} = 0.025$ and the rescaled parameter values are same as described in the caption of Fig. 2 except $\alpha = 10$.

types of environmental mechanical signals. This strain gradient is relative to the source edge indices chosen according to their spatial position along the y axis [Fig. 5(b)]. The slopes of the gradients are drawn from two uniform distributions $\mathcal{U}_1 \in (0.01, 0.1)$ and $\mathcal{U}_2 \in (-0.1, -0.01)$. We consider 10 source edges $(S_1, S_2, \ldots, S_{10})$ in the network and two target edges, target edge "+" and target edge "−" to encode the information of the positive and negative gradients as the desired target strain values $\epsilon_T^{*(+)}$ at the "+" and $\epsilon_T^{*(-)}$ at the "−" target edge correspondingly via training. During training, one slope value (s) was drawn randomly from either $\mathcal{U}_1$ or $\mathcal{U}_2$ in each switching cycle (i.e., a time duration of $\tau_s + \tau_f$) and depending on the sign of the slope, either the "+" or the "−" target edge was trained to reach the corresponding target strain values. The source strain of the nth source edge is given by $\epsilon^*_{S(n)} = nsr$ for the positive slope of strain and $\epsilon^*_{S(n)} = 1 - (n-1)sr$ for the negative slope of strain with r being a uniform random number used to introduce small noise (up to 20%, i.e., $r \in [0, 0.2]$) in the source strain values. The training error for both the positive ($|\frac{\epsilon_T^{(+)} - \epsilon_T^{*(+)}}{\epsilon_T^{*(+)}}|$) and negative ($|\frac{\epsilon_T^{(-)} - \epsilon_T^{*(-)}}{\epsilon_T^{*(-)}}|$) input strain-gradients reduces as the training progresses [Fig. 5(c)]. In the trained network, the target strain values obtained in response to the imposed source strain pattern will be used to evaluate the performance of the classification task. For example, if the target edge "+" reaches the closest to the desired strain value $\epsilon_T^{*(+)}$ and the difference is smaller than a tolerance value ($\mathcal{E}_{\rm tol}$), i.e., $|\epsilon_T^{(+)} - \epsilon_T^{*(+)}| < \mathcal{E}_{\rm tol}$ and $|\epsilon_T^{(+)} - \epsilon_T^{*(+)}| < |\epsilon_T^{(-)} - \epsilon_T^{*(-)}|$ when a positive strain gradient is presented in the sources, the classification of that strain signal by the network is considered successful. We define classification accuracy as the ratio of the number of successful classifications to the number of imposed source strain gradients, i.e., the number of tests.

A set of strain gradients was drawn from $\mathcal{U}_1$ and $\mathcal{U}_2$ and the classification output was checked as we trained the network. The behavior of the trained network indicates the successful classification of the two classes of environmental signals as the classification accuracy increases with the progression of training [Fig. 5(d)]. Since increasing the network activity, as measured by the connectivity parameter ξ, leads to lower training error, we find increasing activity to result in faster classification in the network [Fig. 5(d)].

VII. LEARNING IN THE PRESENCE OF TURNOVER OF NETWORK COMPONENTS

Cellular cytoskeletal networks undergo continuous turnover of their components. To understand how learning dynamics are affected by the turnover of network components, we introduce dynamic turnover of the network edges during training. The edge turnover dynamics is implemented by severing of edges with a timescale $\tau_{\rm sev}$ and reconnecting the nodes of the edge severed with a timescale $\tau_{\rm con}$. During reconnecting, the instantaneous distance between the nodes is set as the new rest length of the reconnected edge. Additionally, edges with strain higher than a critical value $\epsilon > \epsilon^{\rm crit}$ do not sever [Fig. 6(a)]. This strain dependence of the severing captures catch-bond-like dynamics seen in cytoskeletal networks, where parts of the network with higher tension are more stable [35,36]. Thus, in each update during training (i.e., from $t \to t + dt$), the edges with strain smaller than the critical strain ($\epsilon < \epsilon^{\rm crit}$) sever with a probability given by $p_{\rm sev} = dt/\tau_{\rm sev}$. Similarly to severing, reconnection of the severed edges occurs with a probability $p_{\rm con} = dt/\tau_{\rm con}$. We model the effect of strain on severing via a strain threshold based on previous models of cytoskeletal remodeling that occurs beyond a threshold strain value [20,25]. But, a more gradually changing strain-dependent model such as $p_{\rm sev} = dt/\tau_{\rm sev}(\frac{\delta}{\delta+\epsilon})$ also leads to qualitatively similar results (see Fig. 17 of Appendix F for details).

We train the network to achieve a desired mechanical response using the rest length of the edges to be the LDOF, the same as described in Sec. V. To avoid trivial errors, the source and target edges were excluded from the severing and reconnecting dynamics. We find the learning to be successful with a significant number of severing and reconnecting events at the edges [Fig. 6(b)] which leads to a large number of edges being severed during training (see Fig. 18 of Appendix F). Learning becomes slower and the ability to learn decreases as the severing timescale becomes smaller and similar to the timescale of reconnecting [Fig. 6(c)].

The learning altogether stops (i.e., target edge strain cannot reach the desired value) at small values of the ratio of the severing timescale and the reconnection timescale $\tau_{\rm sev}/\tau_{\rm con} \lesssim 10$. Note that we consider severing and reconnection timescales larger than the timescale of driving (i.e., $\tau_{\rm sev}$, $\tau_{\rm con} > \tau_{\rm cyc}$) and faster turnover dynamics would result in loss of learning

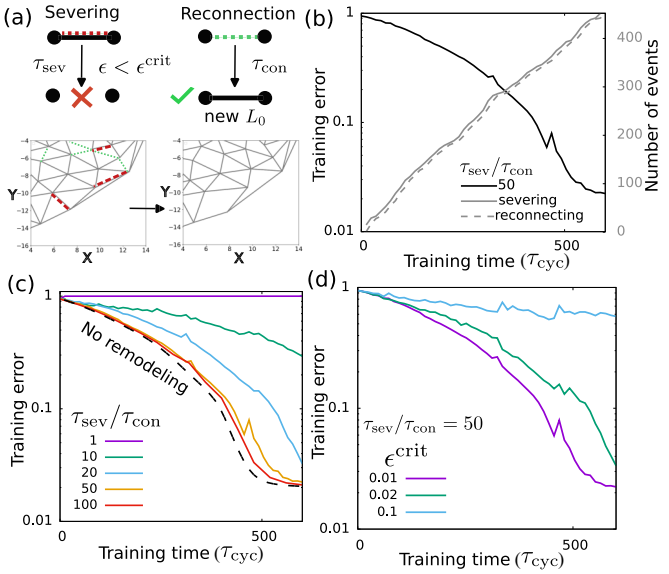

FIG. 6. Learning in the presence of network turnover. (a) The schematic on the top shows edge severing and reconnection processes. The bottom part shows an example of network remodeling. Green and red dashed lines mark the bonds to get created and severed, respectively. (b) Training in the presence of network remodeling. Training error decreases over time. The solid and the dashed gray lines show the number of severing and reconnection events as the training progresses. (c) Training error as a function of the training time at various severing timescales (τ_{sev}). The dashed line shows training without remodeling. (d) Training error with training time at different critical strain values ϵ^{crit}. $\tau_{sev}/\tau_{con} = 50$. The parameter values used here are the same as described in the caption of Fig. 2 except $\epsilon^{crit} = 0.01$, $\tau_{con} = 10\tau_{cyc}$.

even at high τ_{sev}/τ_{con} values. The connectivity, defined as the average number of neighbors of a node, of the dynamic network decreases as the severing timescale decreases, that is, the severing rate increases, which may lead to the loss of learning ability seen at low severing timescales (see Fig. 18 of Appendix F). The choice of the critical strain value (ϵ^{crit}) also significantly affects learning and leads to a slowdown of learning with increasing ϵ^{crit} and loss of learning ability for high ϵ^{crit} values [Fig. 6(d)]. This loss of learning may be the result of the loss of edges that propagate tension from the target edge to the network, getting remodeled.

Interestingly, we find that the network can recover from the loss of learning with increased activity [Fig. 7(a)]. At a fixed ϵ^{crit} value, increased activity reduces the probability of severing the edges by increasing the strain of the edges, which leads to the recovery of learning. The ability of the network to learn in the presence of edge turnover depends on the balance between the severing dynamics and the activity of the network. The network learns better at higher activity values (large ξ) and slower severing rates (larger τ_{sev}) while low activity and fast turnover lead to loss of learning in the network [Fig. 7(b)].

VIII. LEARNING VIA SELF-ORGANIZED PULSATION

The learning mechanism that we have discussed so far requires an external drive at the target edge. This driving

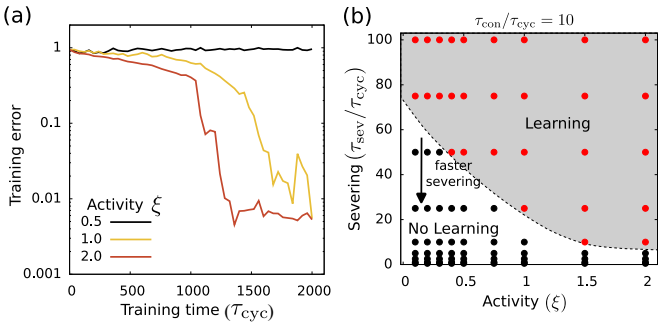

FIG. 7. Activity-dependent recovery from loss of learning. (a) Training error for different activity values (given by the contractility parameter ξ) for $\tau_{sev}/\tau_{con} = 9$. (b) A phase diagram in activity and rescaled severing timescale. The arrow indicates the direction of the faster turnover of network edges. The parameter values used here are the same as described in the caption of Fig. 2. Additionally $\epsilon^{crit} = 0.01$, $\tau_{con} = 10\tau_{cyc}$.

requires knowledge of the desired target strain ϵ_T^* [Eq. (7)] that determines the clamped state. In all of the above cases the training was performed with a fictional "supervisor" who knows the value of ϵ_T^* playing the crucial role of driving the system from the free to clamped state [Fig. 8(a)], thus making the above learning mechanism a supervised one. Although such a learning process may be realized in biomimetic systems of the actomyosin cortex with prescribed external force

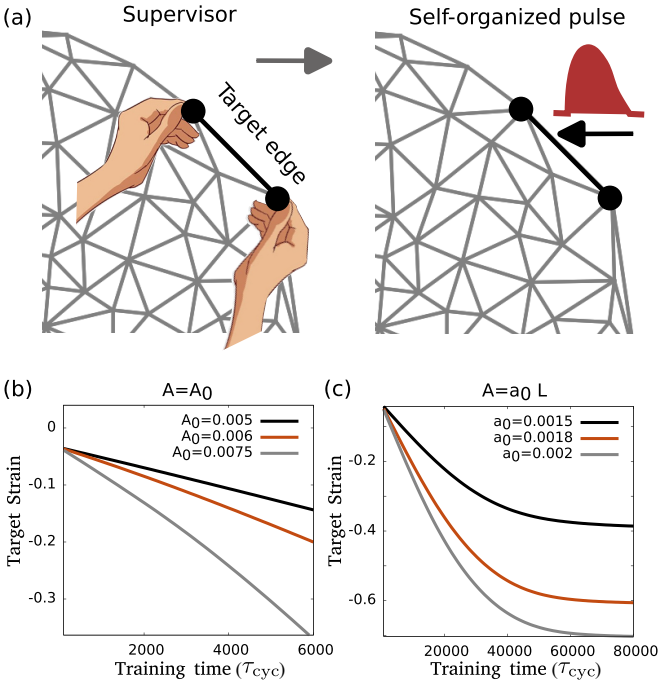

FIG. 8. Self-organized learning via actomyosin pulsation. (a) Schematic showing self-organized pulse driving the target edge instead of the supervisor. (b) Target strain with training time for driving by pulses of constant amplitude $A = A_0$. (c) Target strain with training time for driving by pulses with target edge length (L)-dependent feedback $A = a_0 L$. Here $\tau_f/\tau_s = 1/4$ and the rescaled parameter values are $\beta = 0.1$, $k_u = 0.4$, $\xi = 1$, and $\alpha = 1$.

application in a part of the network, it is not clear how it can happen in a living cell in its native environment.

Here we shall relax the consideration of having an external supervisor in the learning process and ask if meaningful learning is possible through a process that lacks the *a priori* knowledge of the desired target strain ϵ_T^* leaving the role of the supervisor obsolete. Cytoskeletal networks are often driven by self-organized actomyosin forces or *pulses* in various scenarios such as tissue morphogenesis and cell migration during development [37–40]. Such actomyosin pulsation can travel spatially [38,39] leading to asymmetric driving forces by giving rise to fast contractions due to the influx of myosin motors and slow relaxation determined by local turnover timescales. Thus, the forces applied by the supervisor can instead come from such pulses asymmetrically driving the system between the free and clamped states. The clamped state is not set *a priori* by any external drive but emerges from the local regulation of these pulses. Here we shall consider a single pulse train at the target edge and study the resulting learning dynamics [Fig. 8(a)].

We consider a case where the source edge is extended and the target edge experiences actomyosin pulsation that contracts the edge periodically. For simplicity, we shall consider the same sawtooth-like temporal dynamics for actomyosin pulsation defined by a fast and slow timescale (τ_f and τ_s) as we have used before. For pulse trains of constant amplitude $A = A_0$, i.e., the driving force given by Eq. (7) becomes $\boldsymbol{f}^e = \lambda(t)\hat{\boldsymbol{r}}$ with $\lambda_{\max} = A_0$. We find that the target edge learns to contract where the target strain increases with time and is only set by the duration of training, as the pulsation has no means of setting a particular target strain value [Fig. 8(b)]. With increasing training time, the network may encounter floppy mode and change its geometry due to large edge strains, but the target edge keeps contracting without showing any plateau.

Thus, a meaningful desired target strain value does not arise since there is no coupling between the driving force (the pulses) and the network. A specific target strain value may emerge in the case where the pulse amplitude is coupled with the state of the network via a quantity such as the target edge length. We find that if the actomyosin forces exerted on the target edge depend on the edge length (L), e.g., $A = a_0 L$ such that the driving force becomes $\boldsymbol{f}^e = \lambda(t)\hat{\boldsymbol{r}}$ with $\lambda_{\max} = a_0 L$ where a_0 is a constant coefficient, the learning dynamics converges to finite target strain values. With the length-dependent driving, a shorter target edge experiences a smaller force from the actomyosin pulsation, creating a negative feedback on the driving force. Such a negative feedback may arise via signaling pathways between actomyosin density and other regulatory proteins such as E-cadherin, which determines the coupling between actomyosin pulses and cytoskeletal structures around cell junction [41]. For example, actomyosin pulse to a junction decreases E-cadherin levels in that junction and that in turn reduces the chances of subsequent actomyosin pulsation arriving to that junction [41,42], creating a negative feedback. This target edge length-dependent negative feedback on the pulsation amplitude gives rise to specific target strain values in an amplitude-dependent manner [Fig. 8(c)]. Although the described learning mechanism does not say how to control the pulsation to train or test when needed, biochemical signaling pathways in the cell may provide such functionalities. It should be noted that a persistent change in the mechanical behavior of the network (i.e., the strain at the target edge) has occurred here via irreversible changes in edge rest length values, indicating that learning has taken place. Our results suggest that a learning mechanism may emerge from the coupling between the actomyosin pulsation and the cytoskeletal network making it possible to self-organize a physical learning process in a cytoskeletal network of a cell. The *self-organized learning* mechanism, in the described form, should not be thought of as a case of unsupervised learning, which is usually employed for tasks like clustering, dimensionality reduction etc. Rather, it indicates that in biological systems, cell intrinsic processes (e.g., actomysin generated forces) can play the role of a supervisor to facilitate learning.

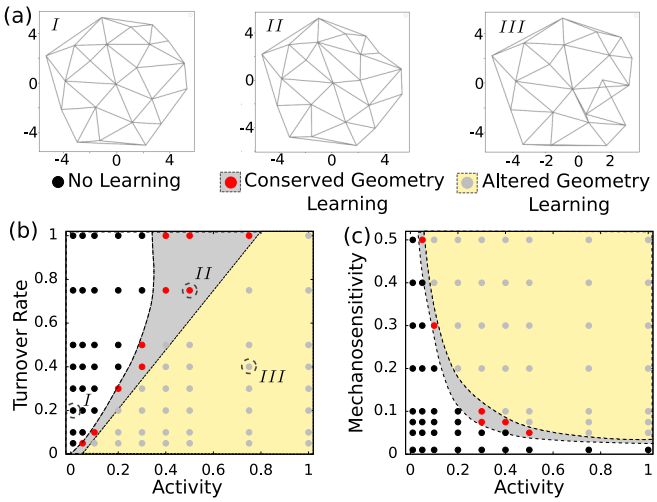

FIG. 9. Interplay between motor dynamics, activity and mechanosensitivity enables self-organized learning. (a) Example of trained networks in three different cases corresponding to the points highlighted by broken circles in the phase diagram. (b) Phase diagram showing phases of *No learning* (white), *Conserved geometry learning* (gray shaded) and *altered geometry learning* (yellow shaded) at different values of activity parameter ξ and motor turnover rate k_u. (c) Phase diagram showing phases of *no learning* (white), *conserved geometry learning* (gray shaded), and *altered geometry learning* (yellow shaded) at different values of activity parameter ξ and a measure of mechanosensitivity β_1 (same as β in rescaled units). The phases are determined from visualization of trained networks and the boundaries are lines drawn as guides to the eyes. Here $\tau_f/\tau_s = 1/4$ and the rescaled parameter values are $\beta = 0.1$ (in panel b), $k_u = 0.5$ (in panel c), and $\alpha = 10$.

We further explore the viability of such self-organized learning with negative feedback [Fig. 8(c)] at different regimes of activity, motor dynamics, and mechanosensitivity. We categorize the learning outcomes into three groups: (i) *no learning*, where the rest lengths do not change at all and no learning occurs; (ii) *conserved geometry learning*, where the rest length of the edges locally remodel to give rise to a well-defined target strain with no change in network geometry; and (iii) *altered geometry learning*, where the rest length changes are large and lead to network geometry changes [Fig. 9(a)]. Note that the network still learns to reach a finite target strain value in the *altered geometry learning* case, but goes through large deformations of some of its edges and changes its

geometry. This effect will crucially depend on the nature of the negative feedback and on the network size with smaller networks more prone to larger changes in geometry. We find that higher activity ξ and lower motor turnover rate k_u (i.e., slower relaxation of motor density) take the network from a nonlearning state to conserved geometry and altered geometry learning states [Fig. 9(b)]. Lower values of mechanosensitivity (β_1) and activity hinder learning in the network, while increasing values of these two parameters enable learning and take the system from conserved geometry learning at moderate values to altered geometry learning at large values of ξ and β_1 [Fig. 9(c)].

IX. ADAPTIVE RESPONSE FROM UNDERLYING LEARNING PROCESS

The results of the previous sections show how learning can happen in a cytoskeletal network. This capability to learn may aid the cell in various complex tasks that require cellular decision-making, but a clear connection between learning and cellular behavior or response is not present. Previous work hypothesized a relationship between learning in biological systems at the cellular scale and cellular adaptation and homeostasis [43]. Here, we show that a self-organized physical learning process can lead to an adaptive mechanical response in the network.

The cell nucleus carries the genetic material, the DNA of the cell, and plays a vital role in the essential physiological functions of the cell. In a noisy mechanical environment, cellular deformations impart mechanical perturbations on the nucleus via the cytoskeletal network [44] and may cause DNA damage [45], nuclear rupture and cell death [46]. To avert such adverse outcomes, the nucleus is known to actively maintain mechanical stability by avoiding large deformations [47,48]. The cytoskeletal network surrounding the nucleus is reported to help maintain nuclear stability by reducing its deformability [48]. Inspired by this observed mechanosensitivity of the nucleus and the need to adapt to mechanical perturbations, we propose an adaptation problem where the cytoskeletal network adapts to lower local strain around the nucleus (i.e., a designated area in the network).

We consider a particular interior region of the network as the perinuclear region, i.e., a target region defined by a set of target edges, and want it to remain in zero strain no matter how the network is deformed by external forces [Fig. 10(a)]. The edges where the network is perturbed or deformed, can be anywhere in the periphery and will be considered as source edges in the learning problem. For simplicity, we will consider three connected edges denoted by T_1, T_2, T_3, to constitute the low-strain target region and apply deformations (marked by magenta arrows) at three different source edges on the periphery of the network denoted by S_1, S_2, S_3 [Fig. 10(b)]. Since we want zero strain, $\epsilon^*_{T_i} = 0$ the driving force becomes

$$\boldsymbol{f}^e_i = -\lambda(t)\nabla\left(\tfrac{1}{2}(\epsilon_{T_i})^2\right), \tag{10}$$

where $i \in 1, 2, 3$ denotes the target edges. Note that the derivative is in the coordinates of the target node positions and the components of force can be written as $f^e_{ix} = -\lambda(t)\epsilon_{Ti}\cos(\theta)$ and $f^e_{iy} = -\lambda(t)\epsilon_{Ti}\sin(\theta)$ where θ denotes the orientation of the target edge. We can rewrite Eq. (10) to express $\boldsymbol{f}^e_i = -\lambda'(t)\hat{\boldsymbol{r}}_i$ with $\lambda'_{\max} = \lambda_{\max}\epsilon_{T_i}$ which is similar to the case of self-organized learning discussed in the previous section but with a strain-dependent negative feedback. For a detailed derivation of the driving force, see Appendix D. Since the driving force is completely dependent on the local strain at the target edge, training for this learning problem can be performed via self-organized forces without any external supervisor. Also, the learning process being continuous, the low-strain region can adapt to changing external mechanical perturbations. We find the training error, i.e., the total strain in the low-strain region $\sum_i |\epsilon_{T_i}|$ reaches a very low value ~ 0 as the training progresses [Fig. 10(c)]. Upon applying further mechanical perturbation, the strain values in the low-strain region initially increase but gradually return to very small values demonstrating the adaptive response of the network [Figs. 10(c) and 10(d)].

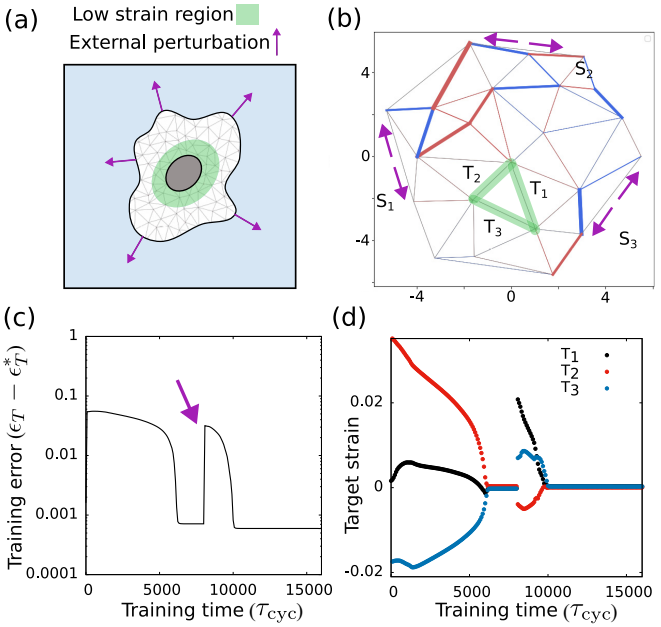

FIG. 10. Adaptation via learning. (a) Schematic showing desired low-strain region in the cell in the presence of external mechanical perturbations. (b) The trained network shows the external forces acting on the network (magenta arrows) and the low-strain region in green. (c) Training error reduces obtaining the desired target strain value $\epsilon^*_T = 0$ for all target edges. (d) Target strain values with training time at three target edges. The τ_{cyc} and rescaled parameter values are the same as Fig. 2 except $\alpha = 10$.

X. DISCUSSION AND CONCLUSION

Cytoskeletal networks in cells perform a diverse array of functions, from providing a basis for cellular motility to controlling cell shape, and to providing tracks for intercellular cargo transport. A large range of complex, conserved feedback mechanisms enable these networks to perform their task reliably [49]. Here, we contextualize such phenomenology through the lens of physical learning. We have demonstrated that cytoskeletal networks can exhibit learning behavior through mechanosensitive protein dynamics and activity-dependent remodeling, establishing a physical

foundation for learning at the cellular level. Our findings show how simple biomolecular ingredients can give rise to sophisticated computational capabilities without requiring specialized neural machinery. Our model reveals several fundamental mechanisms by which cytoskeletal networks can learn and adapt. We demonstrated that these networks can learn to achieve desired strain values at target locations through the remodeling of rest lengths across the network, with trained networks developing appropriate pre-strain patterns that enable specific mechanical responses when stimulated. This learning capability proved independent of network size and edge selection, indicating mechanism robustness. Importantly, we found that increasing contractility enhances learning performance, with higher activity levels leading to lower training errors. Our analysis also revealed that nonlinear elasticity actually aids the learning process, with stronger nonlinearity leading to lower training errors—indicating that complex mechanical properties typical of biological materials may enhance rather than hinder learning capabilities. Beyond basic learning, we demonstrated that trained networks can successfully classify environmental mechanical stimuli, specifically distinguishing between increasing and decreasing strain gradients with accuracy that improves through training and with increasing contractility. Interestingly, we find that learning persists even under continuous turnover of network components—a hallmark of living cytoskeletal networks. While learning becomes slower as component turnover increases, networks maintain their ability to learn despite significant edge severing and reconnection events, with performance dependent on the critical strain threshold and recoverable through increased activity. We further demonstrated that meaningful learning can occur without external supervision through self-organized cellular forces, e.g., actomyosin pulsation. By coupling pulse amplitude to network state, we created a negative feedback mechanism leading to convergent learning—a biologically plausible mechanism for cellular learning without external instruction. This self-organized learning enables practical adaptive responses, as we showed with networks maintaining low-strain regions around sensitive cellular components like the nucleus, demonstrating direct connections between learning mechanisms and biologically relevant cellular functions like maintaining mechanical homeostasis.

Our work can be useful in understanding unicellular learning, where a single cell shows learned behavior without any neurons. Although our results do not directly explain the tasks previously studied in the context of free-living unicellular organisms, such as habituation in Stentor [50,51] and classical conditioning in paramecium [52,53], such tasks may be contextualized for a mechanical network to study if our algorithm can achieve them. Habituation, where a cell learns to stop responding to known signals [50] (i.e., gets habituated), can be mapped to a problem where a mechanical network shows diminished response to a particular mechanical perturbation if presented many times. Contextualized in this manner, habituation may be achieved in a mechanical network using the elements of learning we discuss in this work. Apart from this, the learning paradigm specified here can, in principle, be applicable to a variety of scenarios. Recent works indicate that the memory of cytoskeletal network configurations controls cell motility and gaits in varying conditions [6,54]. Our learning paradigm might shed light on how mechanosensitivity in addition to external forcing, can allow cytoskeletal networks to encode such information and accurately adapt to changes in the environment. In a broader context, our work can potentially have applications in vertex-like models used to describe various developmental processes. In these scenarios, our work can offer insight to how myosin pulsation can help actuate desired mechanical responses [17,55,56].

More broadly, these findings can have implications for understanding cellular adaptation and information processing. The emergent elasticity and morphology of cytoskeletal networks is controlled by a range of mechano-sensitive feedabcks [19]. Our results provide a basis for analyzing and understand how this phenomenology is enabled through the lens of physical learning. Our findings suggest cells may possess mechanical memory encoded in cytoskeletal rest length and stiffness distributions, continuously updatable through mechanosensitive protein dynamics without requiring transcriptional changes. These findings open new avenues for understanding cellular decision-making in development, disease, and regeneration, and may inform the design of adaptive biomaterials that harness the inherent learning capabilities of cytoskeletal networks.

ACKNOWLEDGMENTS

We thank Arvind Murugan for his comments on the manuscript. We thank Rituparno Mandal for providing energy-minimized particle configurations for the initial network creation. D.S.B. thanks Matthew Du, Jordan Shivers, and Carlos Floyd for useful discussions. D.S.B. and S.V. were supported by the National Institute of General Medical Sciences of the NIH under Award No. R35GM147400. M.J.F. is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Sciences program. This research benefited from the Physics Frontier Center for Living Systems funded by the National Science Foundation (Grant No. PHY-2317138) and resources provided by the University of Chicago's Research Computing Center.

DATA AVAILABILITY

The data that support the findings of this article are openly available [57].

APPENDIX A: MODEL DETAILS

To understand how physical learning in a cytoskeletal network may occur, we consider the cytoskeletal network as a disordered network of nodes connected by edges. To create the initial disordered networks, we use an energy-minimized and force-balanced polydisperse particle mixture (with small and big particles in the 1:1 ratio) with harmonic interaction. The networks are obtained by creating a Voronoi tessellation of the particle positions. The initial network is assumed to be tension-free, i.e., the length of each edge (L) is the same as their equilibrium length (L^0). We describe the dynamics of the network by the dynamics of the positions of the nodes $\{\boldsymbol{r}\}$ which governs the length of the edge L and the density of the bound motor m and the bound mechanosensitive protein n at each edge. We define the strain at the edge as $\epsilon = (L - L^0)$.

We shall elaborate on the linearization used here by considering the dynamics of one edge in the network.

We consider a force-dependent dynamics for the number density of bound mechanosensitive proteins given by

$$\dot{n} = k_{bn} - k_{un}^0 e^{-\beta\dot{\epsilon}} n, \tag{A1}$$

where k_{bn} and k_{un}^0 are the bare binding and unbinding rate and β is the coefficient for strain-rate-dependent unbinding. Mechanosensitive protein dynamics can be very fast with a timescale of a few seconds. Here we consider the strain rate change to be a much slower process than the protein dynamics (i.e., consider constant strain rate when solving for n). Now we can linearize the dynamics around a steady-state value n^0 and zero strain $\epsilon^0 = 0$ and solve for the variation δn as

$$\begin{aligned}\delta\dot{n} &= k_{bn} - k_{un}^0(1 - \beta\delta\dot{\epsilon})(n^0 + \delta n)\\ &= \beta n^0 k_{un}^0 \delta\dot{\epsilon} - k_{un}^0 \delta n,\\ \delta n &= n^0\beta\delta\dot{\epsilon} + C_0 e^{-k_{un}^0 t},\end{aligned} \tag{A2}$$

where $n^0 = k_{bn}/k_{un}^0$ is the unstrained steady-state mechanosensitive protein density and C_0 is the constant of integration.

The molecular motor binding-unbinding kinetics is known to be mechanosensitive. Here, we consider a mechanosensitive protein-dependent bound motor dynamics given as

$$\dot{m} = k_b^0 + k_b^1 n - k_u m, \tag{A3}$$

where k_b^0, k_b^1, and k_u are the bare binding and unbinding rates. Now rewriting this above equation in terms of the motor density variation around the steady-state value $m^0 = (k_b^0 + k_b^1 n_0)/k_u$ as $m = m^0 + \delta m$ and linearizing the terms we get

$$\begin{aligned}\delta\dot{m} &= k_b^0 + k_b^1(n_0 + \delta n) - k_u(m_0 + \delta m)\\ &= k_b^1\delta n - k_u\delta m\\ &\simeq k_b^1\beta n_0\delta\dot{\epsilon} - k_u\delta m,\end{aligned} \tag{A4}$$

here we use the steady-state protein variation (as the protein dynamics is fast) to arrive at the above equation. Recent experiments report such strain-rate-dependent enrichment of myosin during junctional remodeling [25].

The dynamics of the learning degree of freedom (i.e., the rest length in this case) of an edge depends on the active force (f^a) on that edge which is a function of the motor density variation in the edge and it is given by

$$\dot{L_i^0} = \alpha g(f^a), \tag{A5}$$

where α is the learning rate and $g(x) = x$ for all $|x| \geqslant g_c$ and 0 otherwise. The learning in this model is continuous and driven by forcing the system from the free state to the clamped state fast and bringing back slowly over two timescales τ_f and τ_s, respectively.

Now, the dynamics of the jth node and the adjacent edge between nodes j and k can be described in terms of node position and the variation of motor density as

$$\begin{aligned}\gamma\dot{\boldsymbol{r}}_j &= \sum_k^{nn} -k_{jk}\big(|\boldsymbol{r}_j - \boldsymbol{r}_k| - L_{jk}^0\big)\hat{\boldsymbol{r}}_{jk} + f_{jk}^a\hat{\boldsymbol{r}}_{jk},\\ \delta\dot{m}_{jk} &= k_b^1\beta n_0\delta\dot{\epsilon}_{jk} - k_u\delta m_{jk},\\ \dot{L}_{jk}^0 &= \alpha g\big(f_{jk}^a\big),\end{aligned} \tag{A6}$$

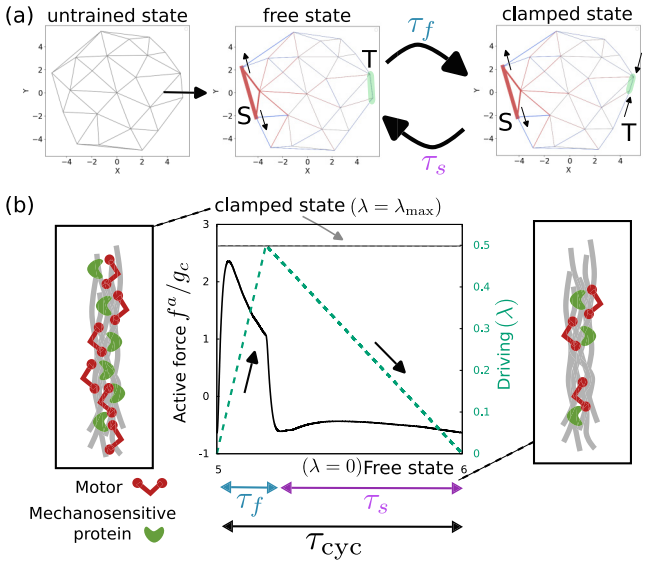

FIG. 11. Contrastive learning via driving. (a) The network in free and clamped states. The thickness indicates strain at each edge and the color indicates contraction (blue) or extension(red). (b) Representative active force dynamics (due to change in motor density) in one edge as the target edge is driven from the free to the clamped state over a fast timescale τ_f and brought back to the free state over a slow timescale τ_s. The motor and mechanosensitive protein density increases when the edge is extended in the clamped state. The figure shows unitless quantities and the rescaled parameter values used in panel (B) are the same as described in the caption of Fig. 2.

where L_{jk}^0 and k_{jk} are the instantaneous rest length and the stiffness of the edge. The active force in the bond is given by $f_{jk}^a = \xi\delta m_{jk}$. We shall use the disordered network with the above-described dynamics to study physical learning in cytoskeletal networks Fig. 11.

1. Rescaled dynamical equations

To derive a set of rescaled dynamical equations, we consider the average distance between the nodes in a network as a length scale l and the duration of a driving cycle $\tau_{\rm cyc} = \tau_f + \tau_a$ as a timescale. We can rewrite the rescaled dynamical equations in terms of unitless parameters as

$$\begin{aligned}\dot{\tilde{\boldsymbol{r}}}_j &= \sum_k^{nn} -\tilde{k}_{jk}\big(|\tilde{\boldsymbol{r}}_j - \tilde{\boldsymbol{r}}_k| - \tilde{L}_{jk}^0\big)\hat{\boldsymbol{r}}_{jk} + \tilde{f}_{jk}^a\hat{\boldsymbol{r}}_{jk},\\ \delta\dot{\tilde{m}}_{jk} &= \tilde{\beta}\delta\dot{\tilde{\epsilon}}_{jk} - \tilde{k}_u\delta\tilde{m}_{jk},\\ \dot{\tilde{L}}_{jk}^0 &= \tilde{\alpha}g\big(\tilde{f}_{jk}^a\big),\end{aligned} \tag{A7}$$

where $\tilde{\boldsymbol{r}} = \boldsymbol{r}/l$, $\delta\tilde{m}_{jk} = \delta m_{jk}/m_0$, $\tilde{L}_{jk}^0 = L_{jk}^0/l$, and $\tilde{f}_{jk}^a = \tilde{\xi}\tilde{m}_{jk}$. The unitless parameters are defined as $\tilde{k}_{jk} = \frac{k\tau_{\rm cyc}}{\gamma}$, $\tilde{\xi} = \frac{\xi m_0\tau_{\rm cyc}}{\gamma l}$, $\tilde{\beta} = \frac{\beta k_b^1 n_0 l}{m_0}$, $\tilde{k}_u = k_u\tau_{\rm cyc}$, and $\tilde{\alpha} = \frac{\alpha\tau_{\rm cyc}^2\xi}{\gamma l^2}$. We shall drop the $\tilde{}$ sign in the discussions to keep notations simple. Here the motor density is rescaled by an equilibrium motor density m_0. The average distance in the networks we used is $l \simeq 2.5\,\mu m$. Here, the kernel timescale is determined by the motor turnover timescale $\tau_k = k_u^{-1}$ (see Appendix C) and molecular motors like the myosin motor have turnover timescale of a few

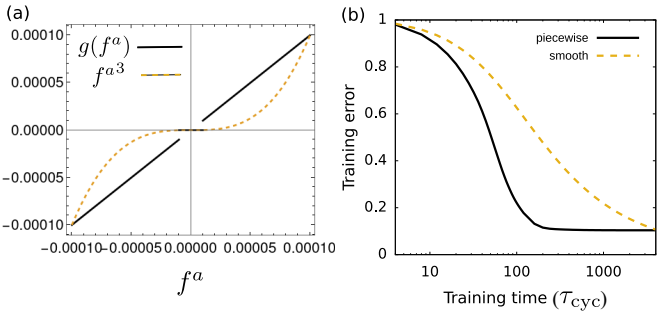

FIG. 12. Learning with turnover of network edges. (a) Average network connectivity at different τ_{sev}/τ_{con} ratio during training. (b) Fraction of all edges that have not been severed at any time during the training (survival fraction). Insets show the network at the beginning and ending of training. The edges that have never been severed are indicated in red. Parameter values used here are the same as described in the caption of Fig. 6.

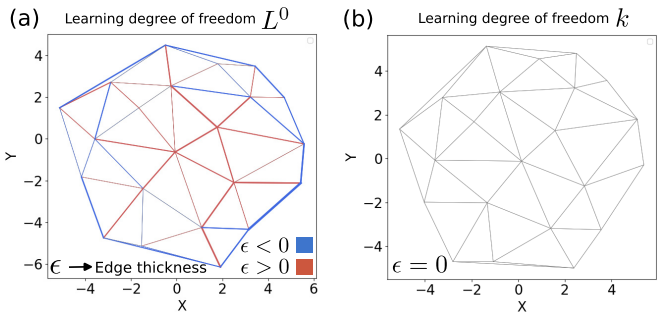

FIG. 13. Strain in the trained network. (a) Strain in the equilibrium state (i.e., while source strain is not applied) in a trained network with the rest length (L^0) as the learning degree of freedom. The edge thickness and color indicate local strain magnitude and sign. (b) A trained network with the edge stiffness (k) as the learning degree of freedom has no strain in the equilibrium state (i.e., without any source strain). The parameters are the same as described in the captions of Figs. 2 and 3.

seconds. For learning via driving at the target, the timescale of driving should be much larger than the timescale of the memory kernel ($\tau_k < \tau_f < \tau_s$) [12]. Hence, we consider a cycle duration much larger than motor turnover timescale $\tau_{cyc} = 100\,\mathrm{s}$ with $\tau_f/\tau_s = 1/4$ unless otherwise specified. In the case of classification of environmental signals and self-organized learning we have used $\tau_{cyc} = 10\,\mathrm{sec}$ and $\tau_{cyc} = 50\,\mathrm{sec}$ correspondingly. We consider an equilibrium motor number density $m_0 = 10\,\mu\mathrm{m}^{-1}$ and the factor $k_b^1 n_0 = 1\,\mu\mathrm{m}^{-1}\mathrm{s}^{-1}$. We shall use these described values to derive a set of unitless parameters for our analysis. The threshold value g_c in the activation function $g(x)$ is a hyper-parameter of learning and set at an optimal value in the range of 10^{-6}–10^{-5} for learning via rest-length change and in the range of 10^{-8}–10^{-6} for learning via stiffness as. We have considered a discontinuous form for g but, using a smooth nonlinearity also leads to qualitatively similar results (Fig. 12).

2. Supervised learning

Here we consider a learning scheme based on the above-described supervised temporal contrastive learning method where the supervisor controls the driving mechanism in which the system is taken from the free state to the clamped state and back according to the desired behavior (strain at the target edge in this case). This driving force at the target nodes can be described as

$$\boldsymbol{f} = -\lambda(t)\nabla\left(\tfrac{1}{2}|\epsilon_T - \epsilon_T^*|^2\right), \tag{A8}$$

where ϵ_T and ϵ_T^* are the instantaneous and desired strain at the target edge. The function $\lambda(t)$ is a sawtooth function devised according to previously used driving in Falk *et al.* [12]. It controls the timescale of driving by incorporating a fast transition (over a time duration τ_f) from free to clamped state and a slow relaxation (over a time duration τ_s) back to the free state. Physical learning, implemented in the above described way, leads to different learned solutions based on the choice of LDOF (Fig. 13) but the overall learning capability does not depend on the choice of source and target edges or the size of the network (Fig. 14). We found the learnig efficiency (i.e., how small the final error is) to depend strongly on the activity of the network. Higher activity led to more learning (i.e., larger changes in LDOF) but also had higher amount of unlearning in each cycle (Fig. 15) leading to the learning to reach a smaller error but slowly.

APPENDIX B: RELATION TO CONTRASTIVE LEARNING FRAMEWORKS

The coupled dynamics of network elasticity and protein response effectively implement a form of *contrastive learning*. Contrastive learning was first introduced for Boltzmann machines [58] and later extended to both stochastic [59] and deterministic [60,61] systems. In deterministic systems, the dynamics minimize an energy (or Lyapunov) function E, so modifying the minima of E changes the input–output relation. These minima are set by the weights (or LDOF) w_{jk}.

In contrastive learning, the system alternates between two states: a *free* state, where only the input is applied, and a *clamped* state, where both input and desired output are enforced. After cycling between these states, weights are

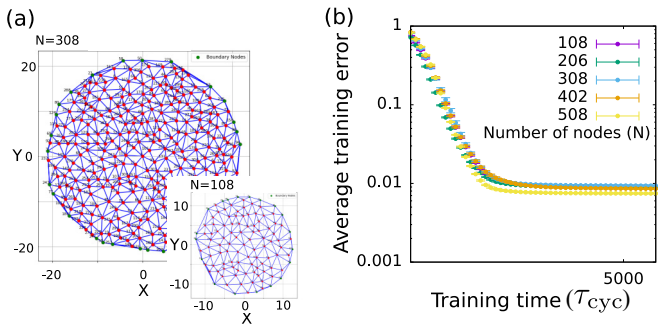

FIG. 14. Effect of network size on learning. (a) Networks of different sizes are used for the learning task. One source and one target edge were randomly chosen from the edges on the network periphery. (b) Training error with training time for different sizes shows no significant effect of network size on learning. The parameters used here are the same as described in the caption of Fig. 2 except $\tau_{cyc} = 800\,\mathrm{s}$.

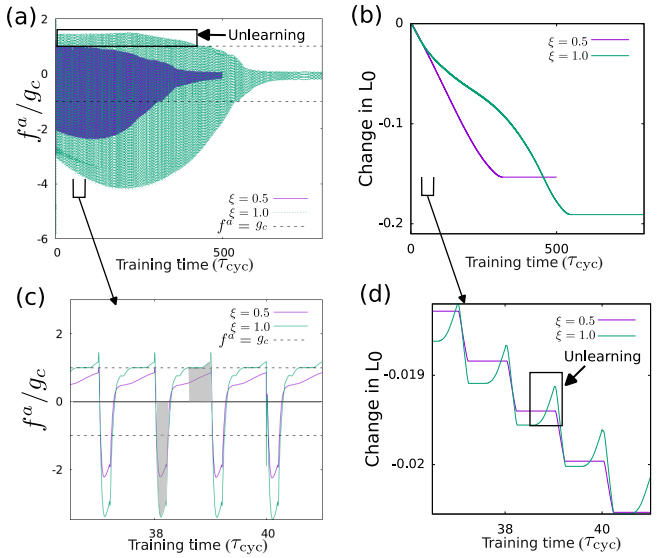

FIG. 15. Effect of activity on learning at a single edge. (a) Rescaled active force with time during training in one specific edge of the network. Higher activity shows longer and more learning but also has a significant period of unlearning (highlighted). (b) Higher activity leads to an increased change in rest length (L^0) of the edge during training. (c) A zoomed-in version of rescaled active stress vs training time shows the difference in active force at different activity values. The shaded region shows learning (when $f^a < 0$) and unlearning (when $f^a > 0$). (d) A zoomed-in version of change in rest length during training shows a larger amount of learning in each cycle and a large amount of unlearning at a higher activity value. The parameters used here are the same as described in the caption of Fig. 2.

updated as

$$\Delta w_{jk} = \alpha\left(s_{jk}^{\text{free}} - s_{jk}^{\text{clamped}}\right), \tag{B1}$$

where α is the learning rate and $s_{jk} = \frac{\partial E}{\partial w_{jk}}$ is the "synaptic current." For example, a Hookean spring network with variable rest lengths would have weights $w_{jk} = L^0_{jk}$ and hence synaptic currents $s_{jk} \propto \epsilon_{jk}$ (the strain across edge jk). In case of stiffness as LDOF, $w_{jk} = k_{jk}$ and the synaptic currents $s_{jk} \propto \epsilon^2_{jk}$.

This update rule is local in space (only involving the strain on edge jk), but nonlocal in time, since it requires comparing the free and clamped states at different times. Digital systems handle this with memory storage, but such explicit memory may not be available in biological systems.

Several strategies have been proposed to overcome this issue, such as using two coupled copies of a system [62] or signals in different domains [63]. Here we follow the approach of Ref. [12], which showed that explicit memory can be replaced by implicit memory generated through feedback control in response to synaptic currents.

Specifically, Ref. [12] demonstrated that contrastive updates can be approximated by

$$\dot{L}^0_{jk} = g(u_{jk}(t)), \tag{B2}$$

where u_{jk} approximates $\dot{\epsilon}_{jk}$ and g is a nonlinear high-pass filter. Such a variable can be expressed as

$$u_{jk}(t) = \int_{-\infty}^{t} K(t-t')\,\epsilon_{jk}(t')\,dt', \tag{B3}$$

with K a nonmonotonic memory kernel. This condition is satisfied by the myosin dynamics in Eqs. (4) and (5). Thus, the active force $f^a_{jk} \propto m_{jk}$ plays the role of u_{jk}, enabling contrastive-like updates without explicit memory storage.

Next, we shall discuss the equivalence of the learning rules in our model with that of a standard contrastive learning based on equilibrium propagation [61], such as coupled learning [11]. In the mechanical network, the learning rules for rest length and stiffness are given by

$$\begin{aligned}
\dot{L}^0_{jk} &= \alpha\partial_{L^0}(E^{\text{free}} - E^{\text{clamped}}) \\
&= -\alpha k(\epsilon^{\text{free}} - \epsilon^{\text{clamped}}), \\
\dot{k}_{jk} &= \alpha\partial_k(E^{\text{free}} - E^{\text{clamped}}) \\
&= \frac{\alpha}{2}((\epsilon^{\text{free}})^2 - (\epsilon^{\text{clamped}})^2),
\end{aligned} \tag{B4}$$

where the energy is given by $E = \sum_{j,k} \frac{1}{2} k_{jk}(L_{jk} - L^0_{jk})^2$ and the strain on the edge is $\epsilon_{jk} = L_{jk} - L^0_{jk}$.

Let us consider the learning rule for the case of learning with rest length as LDOF Eq. (6). For the learning algorithm we discuss here, we can estimate the change in the rest length over one cycle of training as

$$\begin{aligned}
\Delta L^0_{jk} &= \int_0^{\tau_f} \frac{dL^0_{jk}}{dt} dt \\
&= \alpha\xi \int_0^{\tau_f} \delta m dt \\
&\simeq \alpha\xi \int_0^{\tau_f} \frac{d\delta\epsilon}{dt} dt \\
&= \alpha\xi(\epsilon(\tau_f) - \epsilon(0)),
\end{aligned} \tag{B5}$$

where we have used the solution of the myosin dynamics at constant strain-rate $\delta m \propto \delta\dot{\epsilon}$. We also assume the change in LDOF to be negligible over the rest of the cycle, i.e., $\int_{\tau_f}^{\tau_s} dL^0_{jk}/dt dt = 0$. Now, as we drive the system from free to clamped state as the time goes from 0 to τ_f, the above equation can be written as $\Delta L^0_{jk} = \alpha\xi(\epsilon^{\text{clamped}} - \epsilon^{\text{free}}) = -\alpha\xi(\epsilon^{\text{free}} - \epsilon^{\text{clamped}})$, which correctly estimate the difference in strain same as described in the Eq. (B4) for rest length. Note, that this rule does not exactly follow the gradient of the energy since it differs from Eq. (B4) by a factor of k.

Similarly, we can consider the case of stiffness as LDOF,

$$\begin{aligned}
\Delta k_{jk} &= \int_0^{\tau_f} \frac{dk_{jk}}{dt} dt \\
&= -\alpha\xi \int_0^{\tau_f} \delta m \delta\epsilon dt \\
&\simeq -\alpha\xi \int_0^{\tau_f} \delta\epsilon \frac{d\delta\epsilon}{dt} dt
\end{aligned}$$

$$= -\frac{\alpha\xi}{2}\int_0^{\tau_f}\frac{d\delta\epsilon^2}{dt}dt$$
$$= -\frac{\alpha\xi}{2}(\epsilon^2(\tau_f)-\epsilon^2(0)), \tag{B6}$$

and by considering the driving from free to clamped state, we can rewrite the above equation as $\Delta k_{jk} = -\frac{\alpha\xi}{2}((\epsilon^{\text{clamped}})^2 - (\epsilon^{\text{free}})^2) = \frac{\alpha\xi}{2}((\epsilon^{\text{free}})^2 - (\epsilon^{\text{clamped}})^2)$, which is the same as in the standard contrastive learning update rule [Eq. (B4)].

APPENDIX C: MEMORY KERNEL

The motor dynamics coupled with the mechanosensitive proteins, possesses the memory of local strain. We can write the motor density variation in the integral form and use the integration by parts to show

$$\delta m = \beta_1\int_{-\infty}^{t} e^{-\frac{(t-t')}{\tau_k}}\delta\dot{\epsilon}(t')dt'$$
$$= \beta_1\delta\epsilon - \frac{\beta_1}{\tau_k}\int_{-\infty}^{t} e^{-\frac{(t-t')}{\tau_k}}\delta\epsilon(t')dt'$$
$$= \int_{-\infty}^{t}\mathcal{K}(t-t')\delta\epsilon(t')dt', \tag{C1}$$

where the memory kernel is given by

$$\mathcal{K}(t-t') = \beta_1\left(\delta(t-t') - \frac{1}{\tau_k}e^{-\frac{(t-t')}{\tau_k}}\right). \tag{C2}$$

The consistency between the integral form and the motor dynamics described in Eq. (A6) can be understood if we take a derivative of the above integral form [Eq. (C1)] and use Leibniz's integral rule

$$\delta\dot{m} = \beta_1\frac{d}{dt}\left[\int_{-\infty}^{t} e^{-\frac{(t-t')}{\tau_k}}\delta\dot{\epsilon}(t')dt'\right]$$
$$= \beta_1\frac{d}{dt}\left[\int_{-\infty}^{t} f(t,t')dt'\right]$$
$$= \beta_1 f(t,t')\Big|_t\frac{d}{dt}(t) - \beta_1 f(t,t')\Big|_{-\infty}\frac{d}{dt}(-\infty)$$
$$+\beta_1\int_{-\infty}^{t}\frac{\partial}{\partial t}(f(t,t'))$$
$$= \beta_1\delta\dot{\epsilon}(t) - \frac{\beta_1}{\tau_k}\int_{-\infty}^{t} e^{-\frac{(t-t')}{\tau_k}}\delta\dot{\epsilon}(t')dt'$$
$$= \beta_1\delta\dot{\epsilon}(t) - \frac{1}{\tau_k}\delta m, \tag{C3}$$

which we can now compare with the motor dynamics [Eq. (A6)] to identify the parameters $\tau_k = k_u^{-1}$ and $\beta_1 = k_b^1\beta n_0$. In the rescaled parameters, the memory kernel parameters will be slightly different given by $\tilde{\beta}_1 = \tilde{\beta}$ and $\tilde{\tau}_k = \frac{1}{k_u\tau_{\text{cyc}}}$.

APPENDIX D: SUPERVISED FORCE AND SELF-ORGANIZED LEARNING

The learning mechanism we discuss here involves a supervised force at the target edge [see Eq. (7)]. The force at the target nodes depends on the desired target strain at the target and is given by

$$f_x^{e(1)} = -\partial_{x_1}\left(\frac{\lambda}{2}|\epsilon_T - \epsilon_T^*|^2\right),$$
$$f_y^{e(1)} = -\partial_{y_1}\left(\frac{\lambda}{2}|\epsilon_T - \epsilon_T^*|^2\right), \tag{D1}$$

here we consider the force on one node (node 1, given by $\{x_1, y_1\}$) of the target edge and the force on the other node (node 2) will be just opposite $\boldsymbol{f}^{e(2)} = -\boldsymbol{f}^{e(1)}$. Notice that the derivative in the above equation is w.r.t. the node coordinates $\{x_1, y_1\}$. The target strain is given by $\epsilon_T = L_T - L_T^0 = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} - L_T^0$ and ϵ_T^* denotes the desired target strain. We shall derive the form for the x component of the force to understand the nature of supervision. The x component is can be written as

$$f_x^{e(1)} = -\partial_{x_1}\left(\frac{\lambda}{2}|L_T - L_T^0 - \epsilon_T^*|^2\right)$$
$$= -\lambda\left(L_T - L_T^0 - \epsilon_T^*\right)\partial_{x_1}(L_T)$$
$$= \lambda\left(L_T - L_T^0 - \epsilon_T^*\right)\frac{(x_2-x_1)}{L_T}$$
$$= \lambda\left(L_T - L_T^0 - \epsilon_T^*\right)\cos\theta, \tag{D2}$$

where θ is the angle that the edge makes with the horizontal axis. Similarly to the x component of the force, we can derive the y component as $f_y^{e(1)} = \lambda(L_T - L_T^0 - \epsilon_T^*)\sin\theta$. These expressions clearly show that without the knowledge of the desired target strain ϵ_T^* the forces cannot be determined, making the training process supervised, i.e., the supervisor drive the target edge by $\boldsymbol{f}^e$ depending on the desired target strain.

In the Sec. IX, we define a problem with a desired region (set of edges, $i = 1, \ldots, n$) of zero stain $\epsilon_{Ti}^* = 0$ for all i. In this case the driving force at the nodes of the target edges becomes $f_x^{e(1)} = \lambda(L_{Ti} - L_{Ti}^0)\cos\theta_i = \epsilon_{Ti}\cos\theta_i$ and $f_y^{e(1)} = \lambda(L_{Ti} - L_{Ti}^0)\sin\theta_i = \epsilon_{Ti}\sin\theta_i$. Thus, it is clear that the force is completely determined by local quantities and driving can be facilitated by a self-organized force that is proportional to the local (i.e., at the target edge) strain $\boldsymbol{f}^e \propto \epsilon_T$.

APPENDIX E: LEARNING IN NONLINEAR REGIME

We have taken a linear approximation in the mechanosensitive protein to derive a motor-dependent implicit memory of local strain. We find that this linear approximation is not a necessary condition for learning. Here we consider higher-order terms in the protein dynamics given by Eq. (3) in the main text to arrive at

$$\delta\dot{n} \simeq k_{bn} - k_{un}^0\left(1 - \beta\delta\dot{\epsilon} + \frac{\beta^2}{2}\delta\dot{\epsilon}^2 - \frac{\beta^3}{6}\delta\dot{\epsilon}^3\right)(n_0 + \delta n)$$
$$= \beta n_0 k_{un}^0\phi(\delta\dot{\epsilon}) - k_{un}^0\delta n, \tag{E1}$$

where $\phi(\delta\dot{\epsilon}) = \delta\dot{\epsilon} - \frac{\beta}{2}\delta\dot{\epsilon}^2 + \frac{\beta^2}{6}\delta\dot{\epsilon}^3$. We have ignored the contributions of mixed terms (e.g., $\delta n\delta\dot{\epsilon}$) for analytical tractability. Similar to Appendix A, we can obtain the solution for protein dynamics given by

$$\delta n = n_0\beta\phi(\delta\dot{\epsilon}) + C_0' e^{-k_{un}^0 t}, \tag{E2}$$

where C_0' is a constant of the integration.

Now, using the steady-state solution of protein density variation dynamics given above and Eq. (A4), we arrive at motor density variation dynamics given by

$$\begin{aligned}\delta\dot{m} &= k_b^1\delta n - k_u\delta m \\ &\simeq k_b^1\beta n_0\phi(\delta\dot{\epsilon}) - k_u\delta m \\ &= \beta_1\delta\dot{\epsilon} - \beta_2\delta\dot{\epsilon}^2 + \beta_3\delta\dot{\epsilon}^3 - k_u\delta m, \end{aligned} \tag{E3}$$

where $\beta_1 = k_b^1 n_0\beta$, $\beta_2 = k_b^1 n_0\beta^2$, and $\beta_3 = k_b^1 n_0\beta^3$.

To understand the implicit memory in the motor dynamics we consider the integral form given by

$$\begin{aligned}\delta m &= \beta_1\int_{-\infty}^{t} e^{-\frac{(t-t')}{\tau_k}}\left(\delta\dot{\epsilon} - \frac{\beta}{2}\delta\dot{\epsilon}^2 + \frac{\beta^2}{6}\delta\dot{\epsilon}^3\right)dt' \\ &= I_1 + I_2 + I_3, \end{aligned} \tag{E4}$$

where the integral I_1 is same as described in Eq. (C1) and integrals I_2 and I_3 can be written as

$$I_2 = -\frac{\beta}{2}\int_{-\infty}^{t}\mathcal{K}(t-t')h(t')dt' \tag{E5}$$

and

$$I_3 = \frac{\beta^2}{6}\int_{-\infty}^{t}\mathcal{K}(t-t')p(t')dt', \tag{E6}$$

where $h(t) = \int_{-\infty}^{t}\delta\dot{\epsilon}^2 dt$ and $p(t) = \int_{-\infty}^{t}\delta\dot{\epsilon}^3 dt$. We approximate these integrals as

$$\begin{aligned}h(t) &= \int_{-\infty}^{t}\delta\dot{\epsilon}^2 dt \\ &\simeq \int_0^{\epsilon}\delta\dot{\epsilon}d\epsilon \simeq \int_0^{\epsilon}\frac{\epsilon}{\tau}d\epsilon \\ &= \frac{\epsilon^2}{2\tau}, \end{aligned} \tag{E7}$$

where we approximate the strain rate $\sim\frac{\epsilon}{\tau}$ using a characteristic timescale τ. We can rewrite the integral form of motor density variation as

$$\delta m = \int_{-\infty}^{t}\mathcal{K}(t-t')\left(\epsilon(t') - \frac{\beta}{4\tau}\epsilon^2(t') + \frac{\beta^2}{18\tau^2}\epsilon^3(t')\right)dt', \tag{E8}$$

which shows motor density has an implicit memory of a nonlinear function of local strain rather than simply strain. The learning mechanism still works with the same learning rule, even though the motor dynamics does not accurately estimate the difference in strain between the free and the clamped state with the above-described form. The training error indicates no significant change in the nonlinear regime compared to the linear approximation [Fig. 16(a)].

Cytoskeletal networks are known to have nonlinear elasticity. We consider an elastic energy with higher-order terms given by

$$E_{\rm el} = \tfrac{1}{2}k\epsilon^2 + \tfrac{1}{3}k_2\epsilon^3 + \tfrac{1}{4}k_3\epsilon^4, \tag{E9}$$

where k_2 and k_3 are elastic constants corresponding to the higher-order terms. The cubic term in the elastic energy originates from activity in the cytoskeletal networks and can take both signs and the quartic term indicates effects like strain-stiffening. Considering nonlinear elasticity, make the contrastive update [11,12] depend on a nonlinear function of strain

$$\begin{aligned}\dot{L}_0 &= \left(\frac{\partial E_{\rm el}}{\partial L_0}\right)\Big|_{\rm free} - \left(\frac{\partial E_{\rm el}}{\partial L_0}\right)\Big|_{\rm clamped} \\ &= (k\epsilon + k_2\epsilon^2 + k_3\epsilon^3)|_{\rm clamped} \\ &\quad -(k\epsilon + k_2\epsilon^2 + k_3\epsilon^3)|_{\rm free}. \end{aligned} \tag{E10}$$

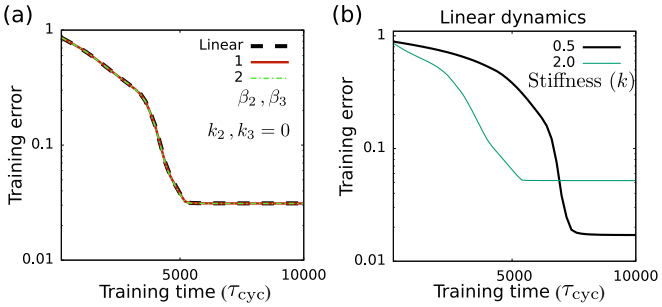

FIG. 16. Learning with nonlinear mechanosensitive protein dynamics. (a) Temporal evolution of training error does not significantly change in presence of moderate nonlinearity. (b) Training error time evolution indicates better learning when the network edges are softer. The parameter values used here are the same as Fig. 2 except $\lambda_{\rm max} = 0.2$ and $\xi = 0.2$.

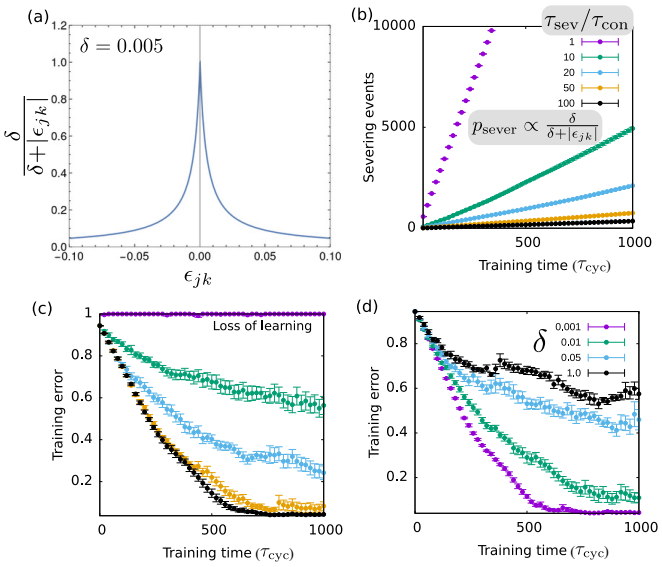

FIG. 17. Learning outcomes with a strain-dependent severing of network edges are qualitatively similar to the strain-threshold model, where edge severing is only allowed below a critical strain $\epsilon^{\rm crit}$. (a) Strain-dependent severing probability is proportional to $\delta/(\delta+\epsilon)$, i.e., $p_{\rm sev} = dt/\tau_{\rm sev}(\frac{\delta}{\delta+\epsilon_{jk}})$. (b) Severing events increase with decreasing $\tau_{\rm sev}/\tau_{\rm con}$ value. (c) Training error as a function of training time shows a slowdown of learning and finally a loss of learning with decreasing $\tau_{\rm sev}/\tau_{\rm con}$ values. (d) The training error as a function of training time shows a slowdown in learning with increasing δ values.

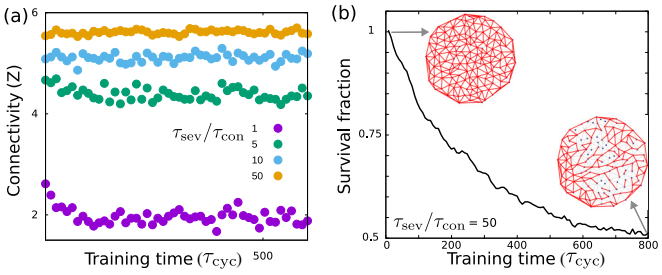

FIG. 18. Learning with turnover of network edges. (a) Average network connectivity at different τ_{sev}/τ_{con} ratio during training. (b) Fraction of all edges that have not been severed at any time during the training (survival fraction). Insets show the network at the beginning and ending of training. The edges that have never been severed are indicated in red. Parameter values used here are the same as described in the caption of Fig. 6.

Hence, the implicit memory of a nonlinear function of strain in motor dynamics can now enable contrastive learning for nonlinear elastic interactions. Together with the nonlinearities in mechanosensitive protein dynamics, we find the strength of nonlinearity to affect learning and with increasing strength of nonlinearity the training error becomes smaller [Fig. 3(d)]. The network softening due to the cubic nonlinearity may play a role in affecting learning as a softer network learns better and reaches lower training error [Fig. 16(b)].

APPENDIX F: NETWORK TURNOVER MODEL

We model the turnover of the cytoskeletal network by severing edges and reconnecting the severed edges at rates given by the timescale of severing τ_{sev} and the timescale of reconnecting τ_{con}. Cytoskeletal networks show catch-bond like behaviour where bundles under higher tension (i.e., at higher strain) are more stable [35,36]. To implement this strain-dependence in the severing mechanism, we apply an additional condition such that only edges below a critical strain ϵ^{crit} can undergo severing. We implement this approach of strain-dependent turnover (or remodeling) beyond a threshold strain (i.e., as a step function of strain) based on previous models [20,25]. A slowly varying strain-dependent model, e.g., strain-dependence in the form of $\frac{\delta}{\delta+\epsilon}$ where the severing probability, given by $dt(\frac{\delta}{\delta+\epsilon})/\tau_{sev}$, decreases as the strain in the edge (ϵ) increases [Fig. 17(a)]. We find the cumulative network severing and learning capability to be qualitatively similar in both the severing models, i.e., the strain-thresholded severing and smoothly varying strain-dependent severing (Fig. 17).

Over the course of training, the survival fraction of the edges, defined as the fraction of all edges that did not undergo severing, decreases (Fig. 18). With increasing network severing rate (i.e., decreasing values of τ_{sev}), the average connevctivity in the network decreases (Fig. 18) and the network loses the ability to learn which may stem from the loss of rigidity in the network.

[1] S. J. Gunst, D. D. Tang, and A. O. Saez, Cytoskeletal remodeling of the airway smooth muscle cell: A mechanism for adaptation to mechanical forces in the lung, Respir. Physiol. Neurobiol. **137**, 151 (2003).

[2] A. Nestor-Bergmann, G. A. Stooke-Vaughan, G. K. Goddard, T. Starborg, O. E. Jensen, and S. Woolner, Decoupling the roles of cell shape and mechanical stress in orienting and cueing epithelial mitosis, Cell Rep. **26**, 2088 (2019).

[3] C. Guillot and T. Lecuit, Mechanics of epithelial tissue homeostasis and morphogenesis, Science **340**, 1185 (2013).

[4] S. Arzash, I. Tah, A. J. Liu, and M. L. Manning, Rigidity of epithelial tissues as a double optimization problem, Phys. Rev. Res. **7**, 013157 (2025).

[5] I. Tah, D. Haertter, J. M. Crawford, D. P. Kiehart, C. F. Schmidt, and A. J. Liu, A minimal vertex model explains how the amnioserosa avoids fluidization during *Drosophila* dorsal closure, Proc. Natl. Acad. Sci. USA **122**, e2322732121 (2025).

[6] Y. Kalukula, M. Luciano, G. Simanov, G. Charras, D. B. Brückner, and S. Gabriele, The actin cortex acts as a mechanical memory of morphology in confined migrating cells, Nat. Phys. **21**, 1451 (2025).

[7] J. F. Fuhrmann, A. Krishna, J. Paijmans, C. Duclut, G. Cwikla, S. Eaton, M. Popović, F. Jülicher, C. D. Modes, and N. A. Dye, Active shape programming drives drosophila wing eversion, Sci. Adv. **10**, eadp0860 (2024).

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

[9] J. W. Rocks, N. Pashine, I. Bischofberger, C. P. Goodrich, A. J. Liu, and S. R. Nagel, Designing allostery-inspired response in mechanical networks, Proc. Natl. Acad. Sci. USA **114**, 2520 (2017).

[10] N. Pashine, D. Hexner, A. J. Liu, and S. R. Nagel, Directed aging, memory, and nature's greed, Sci. Adv. **5**, eaax4215 (2019).

[11] 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).

[12] M. J. Falk, A. T. Strupp, B. Scellier, and A. Murugan, Temporal contrastive learning through implicit non-equilibrium memory, Nat. Commun. **16**, 2163 (2025).

[13] I. Linsmeier, S. Banerjee, P. W. Oakes, W. Jung, T. Kim, and M. P. Murrell, Disordered actomyosin networks are sufficient to produce cooperative and telescopic contractility, Nat. Commun. **7**, 12615 (2016).

[14] C. P. Broedersz and F. C. MacKintosh, Molecular motors stiffen non-affine semiflexible polymer networks, Soft Matter **7**, 3186 (2011).

[15] C. P. Broedersz and F. C. MacKintosh, Modeling semiflexible polymer networks, Rev. Mod. Phys. **86**, 995 (2014).

[16] K. Hayakawa, N. Sato, and T. Obinata, Dynamic reorientation of cultured cells and stress fibers under mechanical stress from periodic stretching, Exp. Cell Res. **268**, 104 (2001).

[17] N. Noll, M. Mani, I. Heemskerk, S. J. Streichan, and B. I. Shraiman, Active tension network model suggests an exotic mechanical state realized in epithelial tissues, Nat. Phys. **13**, 1221 (2017).

[18] F. Rückerl, M. Lenz, T. Betz, J. Manzi, J.-L. Martiel, M. Safouane, R. Paterski-Boujemaa, L. Blanchoin, and C. Sykes,

Adaptive response of actin bundles under mechanical stress, Biophys. J. **113**, 1072 (2017).
[19] K. E. Cavanaugh, M. F. Staddon, E. Munro, S. Banerjee, and M. L. Gardel, Rhoa mediates epithelial cell shape changes via mechanosensitive endocytosis, Dev. Cell **52**, 152 (2020).
[20] M. F. Staddon, K. E. Cavanaugh, E. M. Munro, M. L. Gardel, and S. Banerjee, Mechanosensitive junction remodeling promotes robust epithelial morphogenesis, Biophys. J. **117**, 1739 (2019).
[21] C. A. Anderson, D. R. Kovar, M. L. Gardel, and J. D. Winkelman, Lim domain proteins in cell mechanobiology, Cytoskeleton **78**, 303 (2021).
[22] H. B. Schiller, C. C. Friedel, C. Boulegue, and R. Fässler, Quantitative proteomics of the integrin adhesome show a myosin II-dependent recruitment of lim domain proteins, EMBO Rep. **12**, 259 (2011).
[23] T. P. Lele, J. A. Y. Pendse, S. Kumar, M. Salanga, J. Karavitis, and D. E. Ingber, Mechanical forces alter zyxin unbinding kinetics within focal adhesions of living cells, J. Cell. Physiol. **207**, 187 (2006).
[24] M. A. Smith, L. M. Hoffman, and M. C. Beckerle, Lim proteins in actin cytoskeleton mechanoresponse, Trends Cell Biol. **24**, 575 (2014).
[25] K. Nishizawa, S.-Z. Lin, C. Chardès, J.-F. Rupprecht, and P.-F. Lenne, Two-point optical manipulation reveals mechanosensitive remodeling of cell–cell contacts in vivo, Proc. Natl. Acad. Sci. USA **120**, e2212389120 (2023).
[26] J. Xu, Y. Tseng, and D. Wirtz, Strain hardening of actin filament networks: Regulation by the dynamic cross-linking protein α-actinin, J. Biol. Chem. **275**, 35886 (2000).
[27] P. A. Janmey, S. Hvidt, J. Käs, D. Lerche, A. Maggs, E. Sackmann, M. Schliwa, and T. P. Stossel, The mechanical properties of actin gels. Elastic modulus and filament motions, J. Biol. Chem. **269**, 32503 (1994).
[28] M. A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures* (Wiley, New York, NY, 1991).
[29] R. W. Ogden, *Non-linear Elastic Deformations* (Courier Corporation, North Chelmsford, MA, 1997).
[30] T. J. Perkins and P. S. Swain, Strategies for cellular decision-making, Mol. Syst. Biol. **5**, 326 (2009).
[31] D. E. Discher, D. J. Mooney, and P. W. Zandstra, Growth factors, matrices, and forces combine and control stem cells, Science **324**, 1673 (2009).
[32] Z. Chen, J. M. Linton, S. Xia, X. Fan, D. Yu, J. Wang, R. Zhu, and M. B. Elowitz, A synthetic protein-level neural network in mammalian cells, Science **386**, 1243 (2024).
[33] F. Yang, P. Chen, H. Jiang, T. Xie, Y. Shao, D.-H. Kim, B. Li, and Y. Sun, Directional cell migration guided by a strain gradient, Small **20**, 2302404 (2024).
[34] G. Reig, M. Cerda, N. Sepúlveda, D. Flores, V. Castaneda, M. Tada, S. Härtel, and M. L. Concha, Extra-embryonic tissue spreading directs early embryo morphogenesis in killifish, Nat. Commun. **8**, 15431 (2017).
[35] K. Hayakawa, H. Tatsumi, and M. Sokabe, Actin filaments function as a tension sensor by tension-dependent binding of cofilin to the filament, J. Cell Biol. **195**, 721 (2011).
[36] K. Sato, T. Adachi, M. Matsuo, and Y. Tomita, Quantitative evaluation of threshold fiber strain that induces reorganization of cytoskeletal actin fiber structure in osteoblastic cells, J. Biomech. **38**, 1895 (2005).
[37] A. C. Martin, M. Kaschube, and E. F. Wieschaus, Pulsed contractions of an actin–myosin network drive apical constriction, Nature (London) **457**, 495 (2009).
[38] A. Munjal, J.-M. Philippe, E. Munro, and T. Lecuit, A self-organized biomechanical network drives shape changes during tissue morphogenesis, Nature (London) **524**, 351 (2015).
[39] D. S. Banerjee, A. Munjal, T. Lecuit, and M. Rao, Actomyosin pulsation and flows in an active elastomer with turnover and network remodeling, Nat. Commun. **8**, 1121 (2017).
[40] N. Balaghi and R. Fernandez-Gonzalez, Waves of change: Dynamic actomyosin networks in embryonic development, Curr. Opin. Cell Biol. **91**, 102435 (2024).
[41] M. Rauzi, Cell intercalation in a simple epithelium, Philos Trans. R. Soc. B **375**, 20190552 (2020).
[42] M. Rauzi, P.-F. Lenne, and T. Lecuit, Planar polarized actomyosin contractile flows control epithelial junction remodelling, Nature (London) **468**, 1110 (2010).
[43] J. Gunawardena, Learning outside the brain: Integrating cognitive science and systems biology, Proc. IEEE **110**, 590 (2022).
[44] K. Haase, J. K. L. Macadangdang, C. H. Edrington, C. M. Cuerrier, S. Hadjiantoniou, J. L. Harden, I. S. Skerjanc, and A. E. Pelling, Extracellular forces cause the nucleus to deform in a highly controlled anisotropic manner, Sci. Rep. **6**, 21300 (2016).
[45] P. Shah, C. M. Hobson, S. Cheng, M. J. Colville, M. J. Paszek, R. Superfine, and J. Lammerding, Nuclear deformation causes DNA damage by increasing replication stress, Curr. Biol. **31**, 753 (2021).
[46] C. M. Denais, R. M. Gilbert, P. Isermann, A. L. McGregor, M. Te Lindert, B. Weigelin, P. M. Davidson, P. Friedl, K. Wolf, and J. Lammerding, Nuclear envelope rupture and repair during cancer cell migration, Science **352**, 353 (2016).
[47] A. C. Rowat, J. Lammerding, and J. H. Ipsen, Mechanical properties of the cell nucleus and the effect of emerin deficiency, Biophys. J. **91**, 4649 (2006).
[48] X. Wang, H. Liu, M. Zhu, C. Cao, Z. Xu, Y. Tsatskis, K. Lau, C. Kuok, T. Filleter, H. McNeill *et al.*, Mechanical stability of the cell nucleus–roles played by the cytoskeleton in nuclear deformation and strain recovery, J. Cell Sci. **131**, jcs209627 (2018).
[49] A. Haupt and N. Minc, How cells sense their own shape–mechanisms to probe cell geometry and their implications in cellular organization and function, J. Cell Sci. **131**, jcs214015 (2018).
[50] D. Rajan, T. Makushok, A. Kalish, L. Acuna, A. Bonville, K. C. Almanza, B. Garibay, E. Tang, M. Voss, A. Lin *et al.*, Single-cell analysis of habituation in stentor coeruleus, Curr. Biol. **33**, 241 (2023).
[51] L. Eckert, M. S. Vidal-Saez, Z. Zhao, J. Garcia-Ojalvo, R. Martinez-Corral, and J. Gunawardena, Biochemically plausible models of habituation for single-cell learning, Curr. Biol. **34**, 5646 (2024).
[52] B. Gelber, Investigations of the behavior of paramecium aurelia: I. Modification of behavior after training with reinforcement, J. Compar. Physiol. Psychol. **45**, 58 (1952).
[53] S. J. Gershman, P. E. M. Balbi, C. R. Gallistel, and J. Gunawardena, Reconsidering the evidence for learning in single cells, Elife **10**, e61907 (2021).

[54] B. T. Larson, J. Garbus, J. B. Pollack, and W. F. Marshall, A unicellular walker controlled by a microtubule-based finite-state machine, Curr. Biol. **32**, 3745 (2022).

[55] R. Farhadifar, J.-C. Röper, B. Aigouy, S. Eaton, and F. Jülicher, The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing, Curr. Biol. **17**, 2095 (2007).

[56] S. Alt, P. Ganguly, and G. Salbreux, Vertex models: From cell mechanics to tissue morphogenesis, Philos. Trans. Roy. Soc. B **372**, 20150520 (2017).

[57] D. S. Banerjee, Learning-in-cytoskeletal-networks, https://github.com/DebsankarBanerjee/Learning-in-Cytoskeletal-Networks, Year of last significant update or creation, Accessed: 2025-06-02.

[58] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, Cognit. Sci. **9**, 147 (1985).

[59] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[60] J. R. Movellan, Contrastive Hebbian learning in the continuous hopfield model, in *Connectionist Models* (Elsevier, Amsterdam, 1991), pp. 10–17.

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

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

[63] V. R. Anisetti, B. Scellier, and J. M. Schwarz, Learning by non-interfering feedback chemical signaling in physical networks, Phys. Rev. Res. **5**, 023024 (2023).