Learning by Training: Emergent Physical Memory from Cyclically Tuning Disordered Sphere Packings

Mengjie Zu and Carl P. Goodrich*
Institute of Science and Technology Austria, Am Campus 1, 3400 Klosterneuburg, Austria

(Received 31 August 2025; accepted 4 May 2026; published 18 June 2026)

Many living and artificial systems improve their fitness or performance by adapting to changing environments or diverse training data. However, it remains unclear how environmental variation shapes adaptation, what is learned, and when memory of past conditions is retained. Here we show how cyclic environmental change can produce robust memory. Using a model athermal disordered solid trained by inverse design to attain target elastic properties over a prescribed range, we find that the system evolves toward a marginally absorbing manifold (MAM), meaning that training is reversible within the training range but not beyond it, which encodes a memory of that range. We further propose a general mechanism for MAM formation and memory encoding based on discontinuities in the gradient of the trained quantity. These results provide a simple, broadly applicable physical framework for how adaptive systems learn under changing environments and retain memory of past conditions.

DOI: 10.1103/48k2-cw3b

I. INTRODUCTION

From evolution to artificial neural networks to nearly any inverse-design problem, there is a staggering list of systems where internal variables are iteratively adjusted to optimize one or more emergent features. The process of optimization can vary dramatically in these systems, for example, machine learning is predicated on global gradients [1–8], while many physical systems approximate gradient descent using only local information [9–27]. Many systems, however, also include fixed or slowly varying environmental variables that do not evolve through the same optimization process. These could include the presence of a predator that affects an organism's optimal running speed, the data used to train a large language model, or a desired self-assembly outcome. Such quantities clearly play a critical role in optimization, and if they change, so too will the optimal solution.

However, we still lack a broad understanding of how environmental variation affects optimization over time. In particular, when can a system retain memory of past environments, what is remembered, and how is that memory stored and read out? To address these questions, we show that cyclic environmental variation during training can drive a system to a so-called marginally absorbing manifold (MAM) that stores a memory of the training history, and we uncover a broadly applicable mechanism for how this occurs.

We study memory formation in the context of athermal disordered sphere packings, which we optimize, or train, to control their elastic constants [28,29]. We systematically alter the environment by cyclically changing the target elastic constant between two chosen values, which we refer to as the training range. We find that the system often falls into an absorbing manifold, returning to the same state after every cycle [30], but that this manifold is only marginally absorbing: Training remains reversible within the original range, but not if the target is driven beyond it. This allows both ends of the training range to be read out, indicating that the system has stored them in memory. We then present a general mechanism for the formation of MAMs based on discontinuities in the gradient of the quantity being trained. This mechanism is not specific to our sphere packings and provides a route to nontrivial memory formation under changing environments in a broad class of systems.

The memory we achieve is related to that observed in a wide variety of other physical systems that encode aspects of their past history [31–33]. Some of the many such systems include charge-density waves [34,35], crumpled sheets [36,37], sheared glasses and jammed solids [38–55], suspensions [56–62], and actin networks [63]. The phenomenology of cyclically sheared suspensions of particles provides a particularly relevant example for us: After repeated shearing, the system can reach an absorbing state where particles no longer collide, making the trajectories permanently reversible. However, if a system is subsequently driven beyond the initial maximum strain amplitude, particle collisions occur. In this way, the material encodes a memory of the initial strain amplitude. Our results show that such behavior is possible in systems governed by optimization dynamics, where memory is learned by cyclic training rather than cyclic shear, and provide an explanatory mechanism for such behavior.

The paper is organized as follows. In Sec. II we present our model for trainable disordered sphere packings and discuss our protocol for cyclic training. Section III details our observation that cyclically trained sphere packings approach a marginally absorbing manifold with memory and discusses the nontrivial role of contact changes during training. Section IV identifies the crucial feature of a contact change as

*Contact author: carl.goodrich@ist.ac.at

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

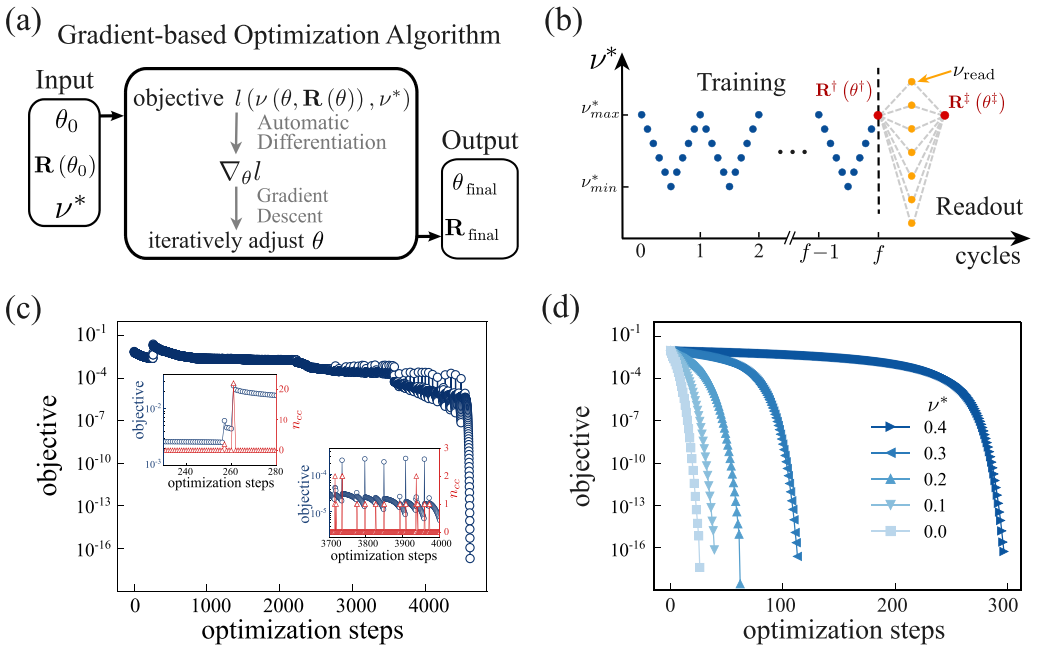


FIG. 1. Training and cyclic training. (a) Schematic of our gradient-based optimization routine to tune the Poisson's ratio ν toward a target value ν^*, which includes input, output, and a gradient-descent algorithm that iteratively modifies the parameters $\boldsymbol{\theta}$. Other elastic constants are trained in the same way. (b) Sketch of cyclic training. We start by training to ν^*_{max} before cyclically adjusting the target ν^* between ν^*_{max} and ν^*_{min}. Each blue point refers to an individual training as described in (a). After f complete cycles (first red point), the positions $\boldsymbol{R}^\dagger$ and parameters $\boldsymbol{\theta}^\dagger$ form the initial inputs for readout measurements. A set of parallel readout trainings are conducted first with a wide range of $\nu^* = \nu^*_{\text{read}}$ (orange points) and then with $\nu^* = \nu^*_{\text{max}}$, generating the final positions $\boldsymbol{R}^\ddagger$ and parameters $\boldsymbol{\theta}^\ddagger$ (second red point). (c) Objective ℓ for a representative example of the initial first training to $\nu^* = 0.5$. The optimization is successful, converging in just over 4500 steps. The insets zoom in on regions where ℓ exhibits sharp changes and highlight their correspondence with contact changes. (d) Objective ℓ during a half cycle of training following 23 complete training cycles in which the system approaches a MAM. The five curves correspond to sequential training steps: from $\nu = 0.5$ to $\nu^* = 0.4$, then from $\nu = 0.4$ to $\nu^* = 0.3$, and so on, down to $\nu^* = 0.0$.

presenting a discontinuity in the gradient of the property being trained, leading to sharp changes in the training dynamics, and develops a theory called gradient discontinuity learning (GDL) that explains how any trainable system with gradient discontinuities can in principle reach a MAM. We discuss these results in Sec. V and speculate on the applicability of GDL to a variety of different systems across fields.

II. CYCLIC TRAINING IN DISORDERED SPHERE PACKINGS

We consider two-dimensional athermal packings of $N = 128$ particles with equal mass m. The particles are evenly divided into $n_{\text{sp}} = 32$ species, with all particles of species α having diameter D_α. Particles i and j interact via a purely repulsive short-range Hertzian potential, given by

$$U_{ij}(r_{ij}) = \begin{cases} \frac{k}{2.5}(1 - r_{ij}/\sigma_{ij})^{2.5}, & r_{ij} < \sigma_{ij} \\ 0, & r_{ij} \geqslant \sigma_{ij}, \end{cases} \quad (1)$$

where r_{ij} is the center-center distance between the particles, σ_{ij} is the mean of their diameters, and k determines the energy scale. We set $m = k = 1$.

Initial systems are prepared by setting half the D_α to 1 and half to 1.4, creating a 50:50 mixture of small and large particles to avoid crystallization. Particles are then randomly placed within a square periodic simulation box of length L, which is set to achieve an area fraction of $\phi_0 = 0.95$. We use the FIRE algorithm [64,65] to obtain a mechanically stable athermal solid by quenching the potential energy to a local energy minimum.

Following the methods presented in Refs. [28,29], we then train various mechanical properties of the system by iteratively adjusting a chosen set of parameters $\boldsymbol{\theta}$. In this work we set $\boldsymbol{\theta}$ to be the n_{sp} species-level particle diameters $\{D_\alpha\}$, but we refer to them as $\boldsymbol{\theta}$ to emphasize that we could consider only a subset of them or include other variables such as energy scales. We focus primarily on training the Poisson's ratio ν, but also consider other elements of the elastic modulus tensor (see Appendix A).

Our training routine is sketched in Fig. 1(a). We start with an initial set of parameters $\boldsymbol{\theta}_0$, a set of particle positions corresponding to a local energy minimum $\boldsymbol{R}(\boldsymbol{\theta}_0)$, and a desired target Poisson's ratio ν^*. The training routine calculates the current value of the Poisson's ratio $\nu(\boldsymbol{\theta}, \boldsymbol{R}(\boldsymbol{\theta}))$ and compares it to ν^* to construct an objective function $\ell(\boldsymbol{\theta}) = [\nu(\boldsymbol{\theta}, \boldsymbol{R}(\boldsymbol{\theta})) - \nu^*]^2$. Here we have made the $\boldsymbol{\theta}$ dependence explicit. We then employ methods of automatic differentiation [66–68] to calculate the gradient $\nabla_{\boldsymbol{\theta}}\ell$, which is fed into a standard gradient-descent routine that iteratively adjusts $\boldsymbol{\theta}$ [and $\boldsymbol{R}(\boldsymbol{\theta})$], until either ℓ drops below 10^{-16} or a maximum number of iterations has been reached. During each iteration, the energy of the system is reminimized with the current $\boldsymbol{\theta}$. More details

can be found in Appendix B. The output of the routine is the final set of parameters and particle positions.

This training routine was sufficiently developed in Refs. [28,29] so that, for our purposes, we can consider it a black box for the inverse design of disordered solids. Here we are interested in exploring the effects that such training has on the system beyond simply controlling the target quantity, e.g., the Poisson's ratio. To explore this, we cyclically train a system, following a protocol outlined in Fig. 1(b). Using the Poisson's ratio ν as our primary example, we iteratively and incrementally train the system between $\nu^*_{\rm max}$ and $\nu^*_{\rm min}$, which are chosen in advance. In the first example in the next section, for instance, we first train at $\nu^* = \nu^*_{\rm max} = 0.5$. Then we use the final $\boldsymbol{\theta}$ and $\boldsymbol{R}(\boldsymbol{\theta})$ as inputs and train with $\nu^* = 0.4$, continuing in this manner with $\nu^* = 0.3$, etc., until $\nu^* = \nu^*_{\rm min} = 0.0$. We then reverse course, setting $\nu^* = 0.1$ and then $\nu^* = 0.2$, until we return to $\nu^* = \nu^*_{\rm max}$. This constitutes a single cycle, and we repeat this process for f cycles.

This protocol is related in spirit to the oscillatory training strategy of Falk *et al*. [69], who showed that switching between incompatible targets selects for adaptable regions of design space. We find that this produces not only adaptable solutions but a marginally absorbing manifold that encodes memory of the training range. To observe this, we also occasionally perform an array of parallel readout calculations. After some number of complete cycles, with $\nu^* = \nu^*_{\rm max}$, we first train with $\nu^* = \nu^*_{\rm read}$ and then train back to $\nu^* = \nu^*_{\rm max}$. This is repeated, always starting with the same initial $\boldsymbol{\theta}$ and $\boldsymbol{R}(\boldsymbol{\theta})$, for a range of $\nu^*_{\rm read}$ that is not limited to being between $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. We will describe the precise measurements and insights that we obtain from these readout calculations below.

Finally, note that this cyclic training protocol is computationally intensive. We typically train for 10–30 cycles, with each training cycle consisting of around ten distinct optimizations. Each optimization typically requires up to approximately 10^4 optimization steps to converge, with each step requiring a highly accurate energy minimization, which itself needs up to approximately 10^3 steps of the FIRE algorithm, and as many force calculations, to converge. To partially mitigate this, during cyclic training we use a variable learning rate and set the maximum number of optimization steps to be 10^3, meaning that we do not always reach the target ν^* during the first few training cycles. We also consider relatively small systems, with $N = 128$ particles, although Refs. [28,29] show that the training of ν is unaffected by system size.

III. CYCLIC TRAINING ENCODES A MEMORY

A. First observation

Figure 1(c) shows the objective function ℓ for the first iteration of training from a random initial state with $\nu = 0.587$ to one with $\nu = \nu^* = 0.5$. This typical training trajectory converges after approximately 4500 optimization steps and highlights two behaviors that are important to understand. First, around the 260th optimization step, the system undergoes a structural rearrangement where the minimum $\boldsymbol{R}(\boldsymbol{\theta})$ becomes a saddle point, resulting in a large-scale change in the system that causes ℓ to increase significantly and about 20 contacts to change (see the first inset). Second, throughout the latter half of the optimization, ℓ exhibits occasional, almost periodic, spikes [70] (see the second inset). The insets also show the number of particle-particle contact changes at each step. The spikes in ℓ occur when one or two contacts change, but are not associated with large-scale structural rearrangements.

This is in stark contrast to what we observe after numerous training cycles. For example, the dark blue curve in Fig. 1(d) shows the first training trajectory of the 24th cycle, where ν is trained from 0.5 to a target of $\nu^* = 0.4$. Subsequent curves show the training trajectories for the rest of this half cycle, ending with $\nu^* = 0$. These trajectories are qualitatively different. First, ℓ decreases smoothly and continuously: There are no spikes or sharp features of any kind, nor are there any structural rearrangements or even contact changes. This is already surprising, as the existence of a configuration that can span a wide range in ν without changing the contact topology is not trivial. Second, training is significantly easier, in that the number of required optimization steps decreases by one to two orders of magnitude.

Clearly, something dramatic beyond the Poisson's ratio has changed during cyclic training. We will see that after a sufficient number of training cycles, the trajectory of the parameters reaches an absorbing manifold, where the same configurations and parameters are revisited every cycle. An absorbing manifold refers to any region in $\boldsymbol{\theta}$ space that spans a certain range in ν (the quantity being trained) such that the parameters remain on this manifold when training within this range. Furthermore, this manifold is marginally absorbing, meaning that it becomes nonabsorbing (the parameters do not return to the manifold) if trained beyond the range defined by $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. Thus, such a marginally absorbing manifold encodes into memory the values of $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. These memories can be read out as jumps or kinks in various quantities, including (i) return-point changes in parameters, (ii) return-point changes in particle positions, (iii) the number of required optimization steps, and (iv) a particular measure of the change in parameters, defined in Eq. (2) below. The rest of this section will detail and quantify these observations and discuss the role of contact changes. In Sec. IV we will build a theory based on the effect of contact changes that can generically predict and explain how a system can learn a specific MAM through cyclic training.

B. Emergent memory after cyclic training

We now discuss a number of quantitative readout measurements, shown in Fig. 2, that demonstrate the MAM as well as the emergence of memory. Readout measurements are performed before the first training cycle but after the initial optimization to $\nu^*_{\rm max}$ (black), after the first complete training cycle (orange), and after the system converges to a MAM (red). The shaded gray regions indicate the training range defined by $\nu^*_{\rm min}$ to $\nu^*_{\rm max}$. We will begin with the left column, which shows a representative system with the Poisson's ratio trained between $\nu^*_{\rm min} = 0$ and $\nu^*_{\rm max} = 0.5$. The other columns, which show different target ranges and mechanical properties, will be discussed later.

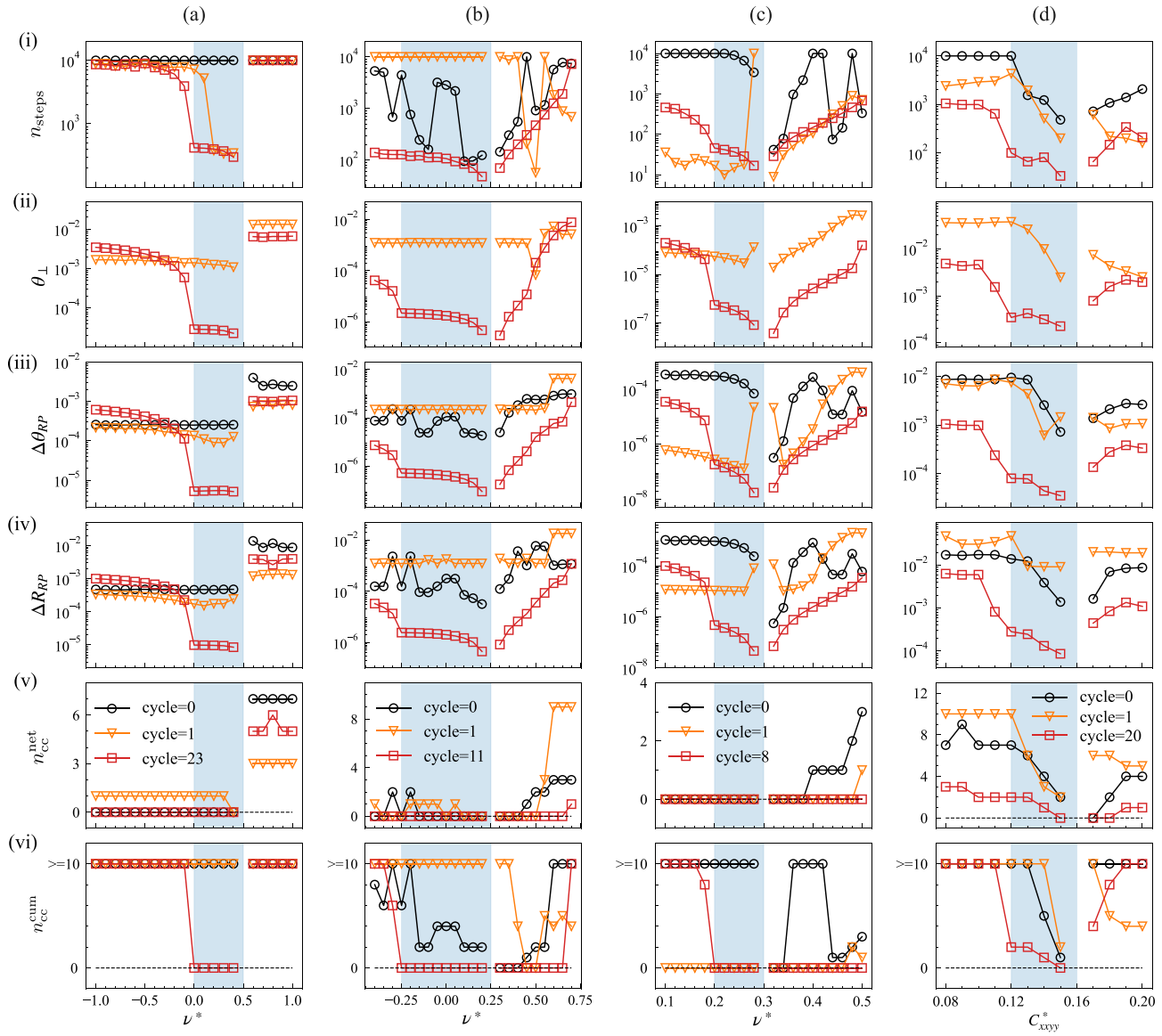


FIG. 2. Memory in four representative examples. Columns (a)–(c) show readout results following cyclic training targeting the Poisson's ratio over different target ranges, while column (d) corresponds to targeting the (arbitrarily chosen) elastic constant c_{xxyy}. The target training ranges are indicated by gray shading. The initial values of the mechanical properties are $\nu_0 = 0.587$, $\nu_0 = 0.681$, $\nu_0 = 0.714$, and $c_{xxyy,0} = 0.157$, respectively. All readout is obtained following the procedure described in the text and begins from a trained state at $\nu^*_{\text{max}} = 0.5$, $\nu^*_{\text{max}} = 0.25$, $\nu^*_{\text{max}} = 0.3$, and $c^*_{xxyy,\text{max}} = 0.16$, respectively. Each row displays a different readout measurement: (i) the number of optimization steps,n_{steps}, (ii) the perpendicular distance $\theta_\perp$ to the line defined by $\hat{\boldsymbol{\vartheta}}$, (iii) return-point changes in the parameters $\Delta\theta_{\text{RP}}$, (iv) return-point changes in particle positions ΔR_{RP}, (v) the net number of contact changes $n_{\text{cc}}^{\text{net}}$, and (vi) the cumulative number of contact changes $n_{\text{cc}}^{\text{cum}}$ measured during readout training with a given ν^*_{read}. Data are shown for readout measurements performed after zero training cycles (but after training to ν^*_{max} or $c^*_{xxyy,\text{max}}$) (black), after one complete training cycle (orange), and after the system reaches a MAM (red).

1. Ease of training

We begin by quantifying the apparent speedup in training observed in Figs. 1(c) and 1(d). Figure 2(a)(i) shows the number of optimization steps n_{steps} to train the system from ν^*_{max} to ν^*_{read}. Note that since we always start at ν^*_{max}, readout data for $\nu^*_{\text{read}} = \nu^*_{\text{max}}$ is meaningless and are not shown. Importantly, for readout calculations, we train with a fixed learning rate of 10^{-6}, while in normal cyclic training we use a variable learning rate (see Appendix C). This enables a more quantitative comparison of n_{steps} for different ν^*_{read}. We terminate training after 10^4 steps, so this should be interpreted as a bound of the true number of steps.

Figure 2(a)(i) shows that before cyclic training, $n_{\text{steps}} = 10^4$, meaning that the optimization has not yet converged. After a single training cycle, n_{steps} drops below 10^3 for some of the ν^*_{read}, while after 23 cycles, n_{steps} is below 10^3 for all the values of ν^*_{read} between ν^*_{min} and ν^*_{max}. However, even after 23 cycles, n_{steps} jumps by over a decade as soon as ν^*_{read} leaves this range. Apparently, whatever makes training easier after many cycles does not strongly affect training outside this range.

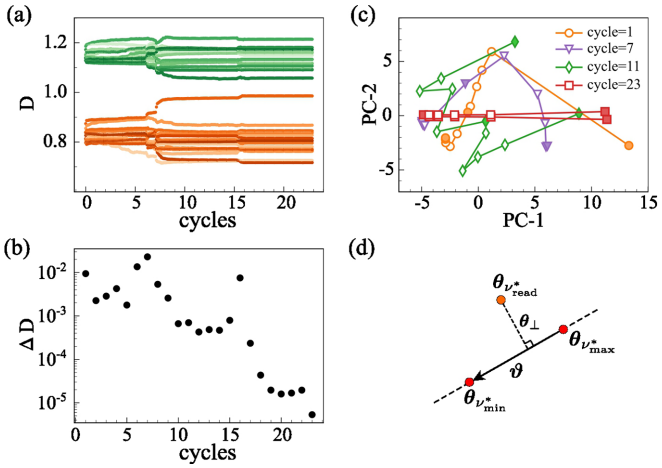


FIG. 3. Evolution of the parameters during cyclic training. (a) The 32 species-level diameters over 23 training cycles. (b) The magnitude of the change in parameters from the beginning to the end of each cycle, ΔD, decreases as the system approaches a MAM. (c) Projection of the parameter vectors onto the first two principal components, shown for a few training cycles. Data points from the same completed cycle are connected by lines, with the starting and ending points marked by solid symbols. After 23 cycles, the parameters appear close to collinear, aligning with the primary principal component. (d) Sketch of $\boldsymbol{\vartheta}$ and $\theta_\perp$. We define the vector $\boldsymbol{\vartheta}$ to point from the parameters at $\nu^*_{\rm max}$ to those at $\nu^*_{\rm min}$ from the last half training cycle. Here $\theta_\perp$ is the distance to the line defined by this vector.

2. Training dynamics

To better understand what makes training easier after 23 cycles, Fig. 3(a) shows how the parameters (which again are species-level particle diameters) evolve during the 23 training cycles. While there are occasionally clear and dramatic parameter changes, e.g., during the seventh and eighth cycles, they eventually reach a steady state and change very little. Figure 3(b) shows the change in the parameters between successive cycles, $\Delta D_k = \sqrt{\frac{1}{n_{\rm sp}}\sum_{\alpha=1}^{n_{\rm sp}}(D_{\alpha,k} - D_{\alpha,k-1})^2}$, where k indicates the cycle and the sum is over the $n_{\rm sp}$ species. This decreases by three orders of magnitude over time, indicating convergence of the cyclic training process and suggesting that the system is progressively approaching an absorbing manifold. To better visualize the parameter changes, we perform a principal component analysis on the data in Fig. 3(a). Figure 3(c) shows the parameters projected onto the two largest principal components for a few select cycles. Unsurprisingly, only for the 23rd cycle, after the system reaches the absorbing manifold, do the parameters return very nearly to where they began.

What is perhaps surprising, however, is that the parameters during the 23rd cycle appear to be nearly collinear. Apparently, once on the absorbing manifold, the training dynamics only explores a single degree of freedom when tuning continuously between $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. To show that this is not a consequence of the particular projection in Fig. 3(c), we define $\theta_\perp$ as the distance (in parameter space) from a given set of parameters $\boldsymbol{\theta}$ to the line that connects the parameters at $\nu^*_{\rm max}$ and $\nu^*_{\rm min}$ from the last half cycle of training [see Fig. 3(d)]. More precisely, given these reference parameters $\boldsymbol{\theta}_{\nu^*_{\rm max}}$ and $\boldsymbol{\theta}_{\nu^*_{\rm min}}$, then

$$\theta_\perp = \left\| \left(\boldsymbol{\theta} - \boldsymbol{\theta}_{\nu^*_{\rm max}}\right) - \frac{\left(\boldsymbol{\theta} - \boldsymbol{\theta}_{\nu^*_{\rm max}}\right)\cdot\boldsymbol{\vartheta}}{\|\boldsymbol{\vartheta}\|^2}\cdot\boldsymbol{\vartheta} \right\|, \tag{2}$$

where

$$\boldsymbol{\vartheta} = \boldsymbol{\theta}_{\nu^*_{\rm min}} - \boldsymbol{\theta}_{\nu^*_{\rm max}}. \tag{3}$$

Figure 2(a)(ii) shows $\theta_\perp$ obtained during the readout calculations. Within the training range, $\theta_\perp$ is less than 10^{-4}, indicating that indeed these parameters are very nearly collinear. This is likely connected to the results of Falk *et al.* [69], who show significantly increased adaptability after cyclicly training elastic networks. However, this collinearity breaks down immediately outside the training range. While the data are not perfectly correlated, this at least informs the apparent ease of training within the absorbing manifold.

3. Reversible training

Both $n_{\rm steps}$ and $\theta_\perp$ show clear features at $\nu^*_{\rm read} = \nu^*_{\rm min}$ and $\nu^*_{\rm read} = \nu^*_{\rm max}$, indicating memory. We now define two return-point measurements that compare the system at $\nu^*_{\rm max}$ before and after intermediate training to $\nu^*_{\rm read}$, both of which also exhibit memory. First, we define

$$\Delta\theta_{\rm RP} = \sqrt{\frac{1}{n_{\rm sp}}\sum_{\alpha=1}^{n_{\rm sp}}(\theta^{\dagger\dagger}_\alpha - \theta^{\dagger}_\alpha)^2}, \tag{4}$$

where the sum runs over the $n_{\rm sp}$ parameters, and $\theta^\dagger$ and $\theta^{\dagger\dagger}$ are the parameters before and after readout training, respectively [Fig. 1(b)]. The $\Delta\theta_{\rm RP}$ quantifies how the parameters change from before readout training, where $\nu = \nu^*_{\rm max}$, to after readout training, where ν again equals $\nu^*_{\rm max}$, so that if the training was completely reversible, then $\Delta\theta_{\rm RP} = 0$. As shown in Fig. 2(a)(iii), $\Delta\theta_{\rm RP}$ is never strictly zero but is quite small ($\Delta\theta_{\rm RP} < 10^{-5}$) after cyclic training for $\nu^*_{\rm min} \leqslant \nu^*_{\rm read} \leqslant \nu^*_{\rm max}$. Note that the nonzero parameter changes are not caused by imperfect convergence during training, but rather are influenced by the learning rate (see Appendix D). However, even after 23 cycles there is a dramatic increase in $\Delta\theta_{\rm RP}$ outside this range.

Similarly, we define

$$\Delta R_{\rm RP} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R^{\dagger\dagger}_i - R^{\dagger}_i)^2}, \tag{5}$$

which quantifies the same return-point change but in the particle positions rather than in the parameters. Figure 2(a)(iv) shows that the behavior of $\Delta R_{\rm RP}$ is qualitatively identical to that of $\Delta\theta_{\rm RP}$. Together, these results show that the readout training is almost completely reversible for $\nu^*_{\rm min} \leqslant \nu^*_{\rm read} \leqslant \nu^*_{\rm max}$, but not reversible otherwise. The sharp jumps in these quantities clearly indicate two-sided memory and are very reminiscent of the memory observed in a variety of cyclically sheared systems [31–33,56–62].

4. MAMs with different training ranges and target quantities

Figures 2(a)(i)–2(a)(iv) demonstrate that cyclic training leads to a MAM, thus storing memory of $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. Im-

portantly, this result is not specific to this one example system or the choice of $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$. Figures 2(b) and 2(c) show similar data for systems with different training ranges for ν, first with a lower range ($\nu^*_{\rm min} = -0.25$ to $\nu^*_{\rm max} = 0.25$) and then with a smaller range ($\nu^*_{\rm min} = 0.2$ to $\nu^*_{\rm max} = 0.3$). Finally, Fig. 2(d) shows the results for cyclically training a different elastic constant, specifically c_{xxyy}.

All cases are representative (see, e.g., Appendix E), and in all cases the system finds a marginally absorbing state and stores memory. The details differ before the MAM is reached (black and orange data), but the final readout measurements (red data) tell the same story. The only anomaly is in Fig. 2(b)(i), where $n_{\rm steps}$ does not show a kink at $\nu^*_{\rm min}$ but remains small for small $\nu^*_{\rm read}$ [71].

Despite the relative similarity in the data, we do not claim that all systems converge to a MAM, and we do find exceptions. Most notably, we have not found an absorbing manifold when the training range includes very large Poisson's ratio. This is because soft spheres exhibit large ν at lower densities where the solid approaches marginal stability and structural rearrangements are unavoidable.

C. Dynamics of contact changes

We now turn our attention to understanding how a system reaches an absorbing manifold and why this manifold is marginally absorbing. First, note that large-scale structural rearrangements are not observed after the system reaches a MAM. Such rearrangements occur for earlier training cycles, e.g., Fig. 1(c), and presumably help steer the system away from minimally stable configurations, but they do not appear critical for the precise encoding of memory. However, particle-particle contacts can form and break without causing a structural rearrangement, e.g., Fig. 1(c), and these topological changes to the configuration will prove to be the key to understanding MAM formation.

To study the dynamics of contact changes, we define the contact map $\mathcal{C}_{ij}(t)$ as

$$\mathcal{C}_{ij}(t) = \Theta(\sigma_{ij}(t) - r_{ij}(t)), \tag{6}$$

where Θ is the Heaviside step function, so that $\mathcal{C}_{ij}(t) = 1$ if particles i and j are in contact at optimization step t and $\mathcal{C}_{ij}(t) = 0$ otherwise.

First, we compare the contact network before and after training from $\nu^*_{\rm max}$ to $\nu^*_{\rm read}$ to compute the net number of contacts that change during training:

$$n_{\rm cc}^{\rm net} = \sum_{i,j>i} |\mathcal{C}_{ij}(n_{\rm steps}) - \mathcal{C}_{ij}(0)|. \tag{7}$$

Figure 2(v) shows that after cyclic training, there are few, if any, contact changes between $\nu^*_{\rm max}$ and $\nu^*_{\rm read}$ provided $\nu^*_{\rm read}$ is within the training range. Unlike the other quantities we have looked at, however, we do not observe consistent kinks or jumps at $\nu^*_{\rm min}$ and $\nu^*_{\rm max}$ that indicate memory.

Interestingly, the small values of $n_{\rm cc}^{\rm net}$ do not mean that contacts do not form and break repeatedly during optimization. To see this, we define the cumulative number of contact changes during optimization:

$$n_{\rm cc}^{\rm cum} = \sum_{t=1}^{n_{\rm steps}} \sum_{i,j>i} |\mathcal{C}_{ij}(t) - \mathcal{C}_{ij}(t-1)|. \tag{8}$$

Unlike $n_{\rm cc}^{\rm net}$, if a contact forms and then breaks, or vice versa, this does not cancel out in $n_{\rm cc}^{\rm cum}$. Figure 2(vi) shows that $n_{\rm cc}^{\rm cum} = n_{\rm cc}^{\rm net}$ within the training range: On the rare occasion that a contact changes, this change persists during optimization. This is not true outside the training range: In most cases, $n_{\rm cc}^{\rm cum}$ becomes much larger than $n_{\rm cc}^{\rm net}$, indicating a large number of contact changes that are then undone.

This suggests that one of the consequences of reaching an absorbing manifold is that temporary contact changes are suppressed along this manifold. For example, in Fig. 1(c) a single contact repeatedly forms and breaks, but this is not observed after cyclic training [Fig. 1(d)]. But why do contact changes matter? The answer is that the Poisson's ratio and other elastic constants depend on the Hessian matrix of second derivatives of the energy with respect to particle positions (see Appendix A). Importantly, our interaction potential (1) is zero when particles do not overlap and proportional to their overlap raised to the power 2.5 when they do. This means that the Hessian, and thus ν, is continuous when contacts form or break. However, training depends on the gradient of ν, not ν itself, and the gradient of ν is discontinuous when a contact changes because it depends on the third derivative of the energy. In the next section we will see that these gradient discontinuities are the critical feature that drives the system into a MAM with encoded memory.

IV. GRADIENT DISCONTINUITY LEARNING

We now present a general mechanism called gradient discontinuity learning that can lead to memory formation through cyclic training. We consider an arbitrary training function $F : \mathcal{R}^n \to \mathcal{R}$ that takes in n parameters $\boldsymbol{\theta}$ and returns a scalar quantity that we wish to cyclically train. For clarity, we do not construct an objective function, but instead consider training dynamics that follow steepest-ascent/descent paths in parameter space. We also assume that F is continuous, but crucially we allow discontinuities in its gradient. As we will see, these gradient discontinuities (GDs) guide the exploration of the parameters under cyclic training, leading to the formation of a MAM that stores memory.

While we will keep this discussion general in this section, one can think of F as the Poisson's ratio, the parameters as the various particle diameters, and the GDs as deriving from contact changes. Here we are not considering any effects analogous to structural rearrangements caused by instabilities in the packing, which lead to discontinuities in the value of F rather than its gradient.

A. Gradient discontinuities guide parameter exploration

We start with an illustrative example that demonstrates how GDs can lead to nontrivial parameter exploration under cyclic training. Consider the simple $n = 2$ dimensional training function shown in Fig. 4(a), which is given by

$$F(x, y) = \tilde{F}(x, y) - f(x, y)\Theta(f(x, y)), \tag{9}$$

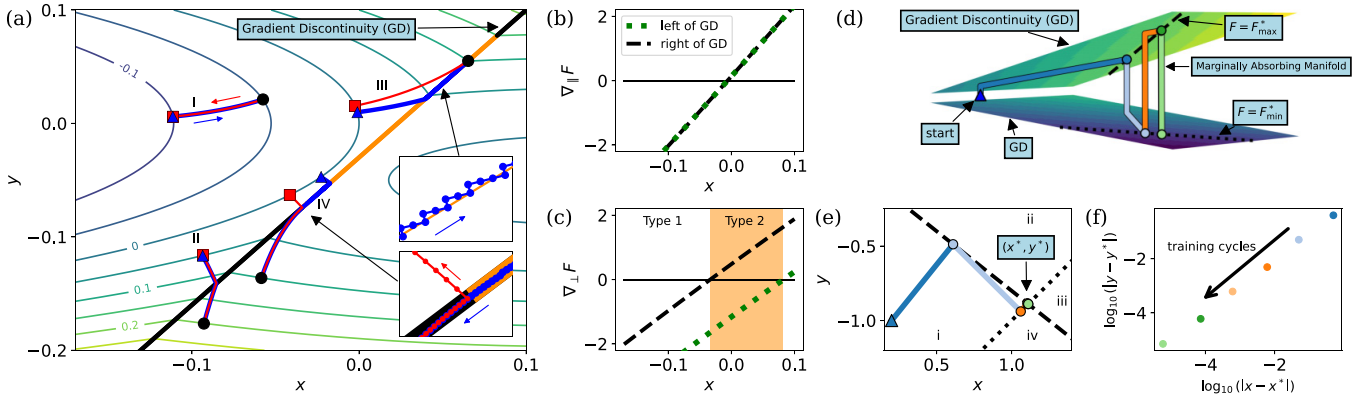


FIG. 4. Gradient discontinuity learning. (a) Example training function $F(x, y)$ with a GD shown by the thick line (black when it is type 1, orange when it is type 2). Training cycles from four cases (labeled I–IV) begin at blue triangles, follow blue steepest-ascent paths until a specified value of F is reached (black circles), and then follow red steepest-descent paths until F returns to its original value (red squares). The insets zoom in on particular regions of interest. (b) Parallel and (c) perpendicular components of ∇F just to the left and to the right of the GD, shown as a function of x. The $\nabla_\parallel F$ must be continuous because F is continuous. A type 2 GD is defined as when $\nabla_\perp F$ changes sign at the GD, as indicated by the orange region. (d) Example training function of three variables where $F(x, y, z) = z$ between two planar type 2 GDs. Starting at the blue triangle, cyclic training between F^*_{max} and F^*_{min} brings the system to a MAM. The dashed lines show where values of F at the GDs equal F^*_{max} and F^*_{min}. (e) Projection of the parameters during training onto the x-y plane. (f) Cyclic training brings parameters iteratively closer to (x^*, y^*), as discussed in the text.

where Θ is the Heaviside function,

$$\tilde{F}(x, y) = x + x^2 + 10y^2 + x^3, \tag{10}$$

and

$$f(x, y) = 1.3x - y - 0.03. \tag{11}$$

$F(x, y)$ has a GD along the line given by $f(x, y) = 0$ (shown by the black/orange solid line). Figure 4(a) also shows four example training cycles, labeled I–IV. Each case begins at an initial set of parameters (blue triangles) with function value F^*_{min}, follows the steepest-ascent path (blue lines) until it reaches F^*_{max} (black circles), and then follows the steepest-descent path (red lines) back down until it returns to F^*_{min} (red squares).

Case I shows that when the ascent path does not cross the GD, the descent path simply returns the system to where it began, because the steepest-descent direction is always the opposite of the steepest-ascent direction [72]. In case II, the ascent path crosses the GD, leading to a sharp and noticeable change in the direction of the path. Nevertheless, the descent path still reverses this, returning the system to where it began. In both of these cases, the system begins in an absorbing manifold because training is completely reversible. However, neither of these manifolds is marginally absorbing because they remain absorbing if the training range is widened.

In contrast, cases III and IV do not end where they begin, and there is a clear nontrivial interaction with the GD. To understand what makes these cases different, Fig. 4 quantifies the discontinuity in the gradient by showing $\nabla_\parallel F$ (b) and $\nabla_\perp F$ (c), which are the components of ∇F parallel and perpendicular to the GD, respectively, evaluated just to the left and just to the right of the GD. Since F itself is continuous, $\nabla_\parallel F$ must be the same on both sides of the GD, as confirmed by Fig. 4(b). However, Fig. 4(c) shows that there is a strong change in $\nabla_\perp F$ from one side to the other.

Now, when the sign of $\nabla_\perp F$ is the same on both sides of the GD, as it is in case II, then a gradient pointing towards the GD on one side will point away from the GD on the other. This allows a gradient-ascent/descent path to "go through" the GD, and we call this a type 1 GD. However, the orange region in Fig. 4(c) shows where $\nabla_\perp F$ changes sign across the GD, and the gradient points towards the GD on both sides. Thus, whichever side one is on, gradient-ascent paths will always return one towards the GD. We call this a type 2 GD. In Fig. 4(a) we color the GD line black to indicate where it is type 1 and orange to indicate where it is type 2.

The consequence of a type 2 GD can be seen in case III in Fig. 4(a): When the blue ascent path hits the GD, it becomes bound to the discontinuity, oscillating back and forth between the two sides of the GD while simultaneously moving in the direction of $\nabla_\parallel F$ (up and to the right). This oscillation can be seen clearly in the first inset, where individual steps of the steepest-ascent algorithm are shown by blue dots. The process stops when F reaches the desired F^*_{max} (in this case $F^*_{max} = 0.1$) at the black dot. The key is that this oscillation about the type 2 GD is not reversible. Starting from the black dot, the red steepest-descent path is not bound to the GD but instead flows downhill away from it [73]. By the time the system returns to F^*_{min} (red square), the parameters have shifted along the $F = 0$ contour line. The red path is an absorbing manifold, and continued training between $F^*_{min} = 0$ and $F^*_{max} = 0.1$ will return the system right up to the GD at the black dot. Furthermore, it is marginally absorbing because training even a little beyond F^*_{max} will require further irreversible movement along the GD [74]. Thus, the value of F^*_{max} constitutes a memory as it can be retrieved through return-point measurements.

Interestingly, a type 2 GD can turn into a type 1 GD, and case IV shows that an ascent path bound to the GD will fall off and follow a "normal" path precisely where this transition occurs. This leads to a nonmarginally absorbing manifold (red

path) that intersects the GD precisely at the point where it transitions from type 2 to type 1. Note that even though the manifold is not marginally absorbing, intersecting the GD at this precise point can be viewed as a different type of memory, although we will not pursue this intriguing possibility further here.

B. Multiple GDs can lead to MAMs with two-sided memory

Figure 4(a) demonstrates a mechanism for exploring functions of multiple variables through training, whereby steepest-ascent/descent paths become bound to and move along type 2 GDs. We now demonstrate how multiple GDs can lead to MAMs with two-sided memory.

Figure 4(d) shows two planar type 2 GDs in an $n = 3$ dimensional training function $F(x, y, z)$; $F(x, y, z) = z$ between the GDs, and the shading along the GDs indicates F. Starting between the two GDs at the blue triangle, the steepest-ascent path (dark blue) first goes straight up until it reaches the top GD and then moves along the GD until it reaches $F^*_{\max}$ (dashed line). This is completely analogous to case III in Fig. 4(a). The system next follows the steepest-descent path (light blue), which starts off going straight down until it reaches the bottom GD and then moves along the GD until it reaches $F^*_{\min}$ (dotted line).

Crucially, the gradient $\nabla_\parallel F$ along the bottom GD is unrelated to $\nabla_\parallel F$ along the top GD, so the descent path does not reverse the ascent path. Figure 4(e) shows the projection of these paths onto the x-y plane, making it clear that the ascent and descent paths affect (x, y) in different ways. Now, a second ascent path up to $F^*_{\max}$ (orange) will again hit and move along the top GD. Figure 4(f) shows that continued cyclic training brings the values of x and y iteratively closer to x^* and y^*, defined by the intersection of the dashed and dotted lines in Fig. 4(e). At this point, the ascent and descent paths only move vertically and end precisely at the GDs, meaning the system has approached a MAM with memory at both $F^*_{\min}$ and $F^*_{\max}$.

Figure 4(e) divides the x-y plane into four quadrants. Cyclic training will result in a MAM with two-sided memory at (x^*, y^*) whenever the initial parameters are in quadrant i (provided they are also between the two GDs). In quadrant iii, no memory is formed because the top and bottom GDs are above $F^*_{\max}$ and below $F^*_{\min}$, respectively, so the ascent/descent paths do not hit the GDs [as in case I in Fig. 4(a)]. In quadrants ii and iv, one-sided memory at either the dashed or dotted line is formed in a single training cycle (as in case III).

C. Gradient discontinuity learning in practice

So far in this section we have used simple examples to show how GDs can guide the evolution of parameters during cyclic training. In real systems with n parameters, GDs are (up to) $(n-1)$-dimensional surfaces [75]. When a steepest-ascent/descent path encounters a type 2 GD, it binds to the GD and follows the steepest-ascent/descent path given by $\nabla_\parallel F$, falling off if and when the GD becomes type 1. We call this process gradient discontinuity learning. Nontrivial learning occurs when cyclic training encounters multiple GDs, and the ascent/descent paths become much more complicated for higher-dimensional training functions and with curved and intersecting GDs.

We have presented these ideas by asserting that the training dynamics follows steepest-ascent/descent paths until a desired function value is reached. In practice, however, it is common to minimize an objective function, such as $\ell = (F - F^*)^2$. Clearly, discontinuities in $\nabla_{\boldsymbol{\theta}} F$ lead to discontinuities in $\nabla_{\boldsymbol{\theta}} \ell = 2(F - F^*)\nabla_{\boldsymbol{\theta}} F$, and the theory is unaffected. Interestingly, many inverse design problems include constraints on the parameters. One way to incorporate such constraints into the objective is to define a function $c(\boldsymbol{\theta})$ that is positive when the constraint is active and define the objective as

$$\ell = (F - F^*)^2 + ac(\boldsymbol{\theta})\Theta(c(\boldsymbol{\theta})), \tag{12}$$

where Θ is again the Heaviside step function and a is a constant. Now, the gradient of ℓ becomes

$$\nabla_{\boldsymbol{\theta}} \ell = 2(F - F^*)\nabla_{\boldsymbol{\theta}} F + a[\nabla_{\boldsymbol{\theta}} c(\boldsymbol{\theta})]\Theta(c(\boldsymbol{\theta})), \tag{13}$$

meaning that as long as $\nabla_{\boldsymbol{\theta}} c|_{c=0} \neq 0$, there is a discontinuity in the gradient of ℓ when the constraint becomes active. Furthermore, this GD is always a type 2 GD in the limit of hard constraints, $|a| \to \infty$. Therefore, even if $\nabla_{\boldsymbol{\theta}} F$ is continuous, GDL can nevertheless exploit constraints to encode memory.

Learning without discontinuities in $\nabla_{\boldsymbol{\theta}} F$ can also result from the practical details of optimization. Perfectly reversible, nonabsorbing manifolds like case I in Fig. 4(a) rely on training dynamics following precise ascent/descent paths. However, training dynamics rarely follows such paths exactly. Finite step sizes always lead to small deviations from ideal paths, and many algorithms (e.g., conjugate gradient, LBFGS, or any momentum-based algorithm) intentionally move in directions informed by, but not parallel to, the gradient. The consequence of this is that GDs are not strictly necessary: Regions of high curvature can cause actual training paths to be nonabsorbing, causing nontrivial movement in the parameters in directions perpendicular to the gradient. Gradient discontinuity learning should be viewed as a precise description of the ideal case, where parameter exploration in directions perpendicular to the gradient is guided by GDs. While other mechanisms for such parameter exploration exist (e.g., Dillavou *et al.* [27] show that hardware imperfections in experimental learning networks introduce deterministic biases into the update rule that drive parameter drift during repeated training), GDL provides a simple mechanistic understanding of memory formation.

Finally, can GDL apply to systems, like sheared solids and suspensions, that are trained through an applied field rather than through explicit inverse design? We can generalize GDL by considering any system characterized by parameters $\boldsymbol{\theta}$ subject to environmental variables e, where the parameters respond to the environmental variables so as to maintain a constraint $G(e, \boldsymbol{\theta}) = 0$ imposed by the physics of the system. This constraint could, for example, specify a desired target property, i.e., $G(F^*, \boldsymbol{\theta}) = F(\boldsymbol{\theta}) - F^* = 0$, or impose force balance on all particles in a sheared suspension (discussed below). The solutions to this constraint define an implicit function $\boldsymbol{\theta}^*(e)$, so that for any e, $G(e, \boldsymbol{\theta}^*(e)) = 0$. When e changes, the parameters will change in directions determined in part by $\nabla_{\boldsymbol{\theta}} G$ so that the parameter trajectories are reversible

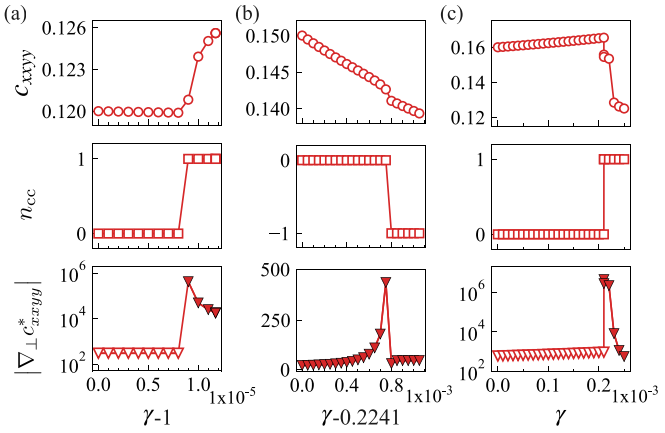


FIG. 5. Confirming predictions made by gradient discontinuity learning. Using the system from Fig. 2(d) after cyclic training, we systematically set the parameters to be along the line defined by $\hat{\boldsymbol{\vartheta}}$ [Eq. (3)] to identify contact changes and probe the GD type. Specifically, we set $\boldsymbol{\theta} = \boldsymbol{\theta}_{c^*_{xxyy,\max}} + \gamma\hat{\boldsymbol{\vartheta}}$, so that $\boldsymbol{\theta} = \boldsymbol{\theta}_{c^*_{xxyy,\max}}$ at $\gamma = 0$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_{c^*_{xxyy,\min}}$ at $\gamma = 1$. Data are shown as a function of γ and is focused around three contact changes: (a) just below the training range, near $\gamma = 1$; (b) in the middle of the training range, near $\gamma = 0.2241$; and (c) just above the training range, near $\gamma = 0$. The top row shows the value of c_{xxyy}, the middle row shows the change in the number of contacts, allowing us to identify the moment of contact change, and the bottom row shows the magnitude $|\nabla_\perp c_{xxyy}|$, with positive (negative) values of $\nabla_\perp c_{xxyy}$ given by open (closed) symbols. As predicted, the GD in the middle of the MAM (b) is of type 1 because $\nabla_\perp c_{xxyy}$ does not change sign (prediction 1), while the two GDs at the edges of the MAM are type 2 because $\nabla_\perp c_{xxyy}$ does change sign (prediction 2).

provided the gradients of G are continuous. When the system encounters gradient discontinuities, irreversible dynamics will move the parameters along the gradient discontinuities. Gradient discontinuity learning therefore provides a straightforward mechanism for moving the parameters in directions perpendicular to $\nabla_{\boldsymbol{\theta}} G$, though we leave a more thorough investigation of this scenario to future work.

D. GDL explains the observed MAMs in disordered packings

Does GDL actually explain the results of Figs. 1–3? As already stated, GDs arise from particles coming in and out of contact, and typical training encounters many such events, so GDs have the opportunity to influence the training. Furthermore, our mechanism makes precise predictions that we can test. If a MAM with two-sided memory is reached through this GD-based mechanism by cyclically training between $\nu^*_{\min}$ and $\nu^*_{\max}$, then we make the following predictions.

Prediction 1. If any contacts change in the middle of the MAM, i.e., between $\nu^*_{\min}$ and $\nu^*_{\max}$, they must correspond to type 1 GDs.

Prediction 2. At both ends of the MAM, i.e., at $\nu^*_{\min}$ and $\nu^*_{\max}$, a contact between two particles should be just on the verge of forming or breaking, with the corresponding GD being type 2. If the MAM only has one-sided memory, then this applies only to that end.

Prediction 3. During the learning process, there should be times of rapid contact-change oscillations, corresponding to the system moving along a type 2 GD.

These predictions are not trivial: They are unlikely to occur by chance and could easily be wrong if memory was formed through a different mechanism. Unlike prediction 1, predictions 2 and 3 assume that learning is guided precisely by a GD rather than a region of high curvature (see discussion in Sec. IV C), and therefore may not always hold. This is consistent with our observations: Predictions 1 and 3 always hold, while prediction 2 is true most of the time, but with a few exceptions.

We have already seen the oscillations of prediction 3 in the inset to Fig. 1(c), which is representative and generic. Predictions 1 and 2 are demonstrated in Fig. 5, which shows contact changes and $\nabla_\perp c_{xxyy}$ at various points of the MAM from Fig. 2(d). Specifically, we zoom in around three contact changes (shown by the middle row) that occur at the beginning [Fig. 5(a)], middle [Fig. 5(b)], and end [Fig. 5(c)] of the MAM. Figures 5(a) and 5(c) show that contacts change almost immediately outside of the MAM and that $\nabla_\perp c_{xxyy}$ changes sign (prediction 2). Conversely, this MAM happens to have a contact change in the middle, but here $\nabla_\perp c_{xxyy}$ does not change sign (prediction 1). The counterexamples for prediction 2 are the systems in Figs. 2(b) and 2(c), where no contacts are on the verge of changing near $\nu^*_{\max}$. These nontrivial observations are consistent with what we would expect from GDL, and the counterexamples motivate future investigations into the more complicated scenario of learning without GDs.

V. DISCUSSION

We have shown that cyclic training in tunable disordered solids can produce physical memory by driving the system toward a marginally absorbing manifold that encodes the training range. This memory is highly reminiscent of that observed in cyclically sheared suspensions but is distinct from the return-point memory seen in cyclically sheared disordered solids, which is characterized by periodic but not reversible dynamics [31–33]. We further proposed a general mechanism, gradient discontinuity learning, in which discontinuities in the gradient of the training function guide the training dynamics and help generate MAMs.

These results bring us closer to a general understanding of the effects of a slowly changing environment on a system undergoing optimization. While we exclusively consider changes to the desired value of an observable, many environmental variables instead affect observables directly. In our case, for example, we could keep the desired Poisson's ratio fixed while cyclically varying the total packing fraction, which would systematically affect the optimized solution in much the same way as changing the target behavior. Gradient discontinuity learning theory can apply here as well (see Sec. IV C) because changing such variables will cause the optimal solution to change, and if this optimal solution encounters one or more type 2 GDs, the system may approach a MAM.

While GDL provides a mechanism for memory formation, there are no guarantees that a given system will actually reach a MAM. Even if there are GDs in the training function, if

their density is too low then training may not encounter them, resulting in an absorbing manifold that is not marginal and thus does not store memory [as in case I in Fig. 4(a)]. On the other hand, if the density of GDs is too high, then training may not be able to find an absorbing manifold that spans the training range. In our athermal sphere packings, it is somewhat surprising that it is possible to train between $\nu = 0.5$ and $\nu = 0$ without contact changes, and we do not find MAMs when training to higher ν's.

In this paper we have focused on training a single observable. However, just as one can form memory from cyclically shearing in multiple directions [76,77], one could consider training multiple properties. Bulk mechanical properties of disordered solids are known to be fully independent [29], meaning that they can be tuned separately and even simultaneously, opening the door for a variety of interesting training protocols. As a simple preliminary demonstration of this, Fig. 10 in Appendix G shows that we can approach an absorbing manifold by alternately training between two different elastic constants, i.e., first a cycle of one and then a cycle of the other. However, we leave a proper quantification and analysis of the memory contained in such systems to future work.

One major consideration that we have not discussed is the role of noise. Keim and Nagel [59] showed that in sheared suspensions, thermal noise plays a critical role in the formation of multiple transient memories, and it is not clear if GDL can lead to multiple memories in the same way. As discussed above, training does not exactly follow steepest-ascent/descent paths, which could be interpreted as a form of noise, but it would be interesting to systematically investigate the consequence of introducing random noise either to the initial parameters at the beginning of each optimization or during every optimization step. It would also be interesting to study what happens if the system is only allowed to partially relax during the training process: Rather than fully minimizing the energy of the configuration during each training step, one could update parameters on the same timescale as the system relaxes, potentially achieving faster and more robust memory.

Our results are closely related to, but distinct from, the oscillatory training framework of Falk *et al*. [69], who showed that switching between two incompatible target functionalities selects for adaptable regions of design space where the two solutions are close together in parameter space. Both approaches demonstrate that cyclic training produces nontrivial emergent structure beyond simply achieving the current target, and both rely critically on the degeneracy of high-dimensional design spaces to make this possible. While Falk *et al*. frame this emergent structure in terms of adaptability, we find that cyclic training also encodes memory of the training range in the form of a MAM. These two perspectives are complementary: An adaptable system in the sense of Falk *et al*. has implicitly found a region of parameter space that is low cost to traverse between functions, which is related to our finding that the MAM makes training within the range dramatically easier. We further propose GDL as a potential mechanism underlying such behavior, though establishing this connection more rigorously, particularly for training protocols that differ from gradient descent, is an interesting direction for future work.

While we have focused on training the elastic properties of disordered sphere packings, GDL is much more general and should be applicable in a wide range of settings. First and foremost, our results encourage cyclic training in all manner of systems that can be inverse designed, especially those with clear GDs or complex constraints. However, GDL may also play a role even in systems that are not explicitly inverse designed. We now speculate briefly about a few such possibilities.

Sheared suspensions. The phenomenology we observe in trained disordered solids is very similar to that of cyclically sheared suspensions [56–62]. While the physics of these systems can be understood on their own, they can also be viewed through the lens of our theory. Since the net force on every particle, which must vanish at all times, depends on the strain γ and the nonaffine position of every particle, $R_{\rm na}$, we have $F_{\rm net}(\gamma, R_{\rm na}) = 0$. This maps exactly to GDL, as discussed at the end of Sec. IV C, with GDs deriving from particle contacts.

Sheared jammed solids. This presumably also can be done for memory formation in sheared jammed solids, but this situation is more complex with a very different form of memory [31]. First, as observed by Arceri *et al*. [78], marginal stability is crucial for memory formation, implying the presence of structural instabilities within the training range. Structural instabilities can lead to both irreversible or reversible rearrangements [46], and it is well known that such systems can develop memory while undergoing reversible rearrangements [32,43,44]. However, this implies a discontinuity in the training function, not just in its gradient, and further work is needed to incorporate reversible rearrangements, discontinuities, and hysteretic cycles into the paradigm presented here.

Crumpled sheets. Another well-established form of physical memory is when a system can recall the maximum value of a previously applied perturbation. First observations include the Kaiser effect, describing the acoustic emission of a metal under strain [79], and the Mullins effect, describing stretched rubber [80]. This is also seen in crumpled sheets compressed by a piston of mass m [31,36,37]. The height of the piston is a reproducible function of m, provided m does not exceed the previous maximum applied mass, effectively encoding this previous maximum mass. Conceptually, this is very similar to the single-shot memory discussed in case III in Fig. 4(a), and the constraint that the crumpled sheet is internally stable presumably leads to discontinuities caused by further crumpling. While the application of GDL to crumpled sheets is therefore quite plausible, further work is again need to solidify this connection.

Evolution. The role of environmental variation in shaping adaptive behavior has been studied in population genetics models, where temporal correlations in environmental fluctuations have been shown to favor phenotypic switching strategies that exploit memory of past states [81]. This motivates the search for GDL-like mechanisms in biological evolution. Importantly, the genotype-phenotype (GP) map between an organism's genotypes and phenotypes is highly degenerate [82–85], and genotypes can combine in highly nontrivial, collective ways to affect fitness [86–90]. Effective GDs can arise, e.g., whenever a phenotype is limited by multiple genotypes or if a genotype can be deleterious past a threshold that depends on other genotypes. Furthermore, environmental changes are well known to play a key

role in evolution, with comparable timescales for evolutionary and ecological change [91]. With all the necessary ingredients in place, it is at least plausible that GDL has played a role in the exploration of the apparent degeneracy in the GP map.

Phenotypic plasicity and (re)adaptation. Phenotypic plasticity, where a single genotype can exhibit multiple phenotypes depending on environmental inputs, may provide more direct examples than pure genetic evolution. For instance, the number of nuclei in muscle cells increases with muscle mass, but then remains roughly constant during muscle atrophy [92,93]. This makes it significantly easier to gain muscle mass the second time, but only back to previous levels, creating a memory familiar to anyone who frequents the gym. Here the training function (muscle mass) is constrained by the number of nuclei, creating a GD that guides the initial increase in nuclei. However, the easiest way to decrease muscle mass does not involve removing nuclei, creating a scenario similar to case III in Fig. 4(a).

As a second example that mixes phenotypic plasticity with genetic evolution, Ho *et al.* [94] show that "organisms generally 'remember' their ancestral environments via phenotypic plasticity" and present a potential mechanism for long-term memories. We can reformulate this mechanism in the context of GDL, as follows. The amount of expression of a hypothetical gene can be regulated through transcription factors that are controlled, in part, by environmental cues. However, such phenotypic plasticity is limited by regulatory motifs, which are genetically controlled. This creates a discontinuity in the fitness gradient when transcription factors and regulatory motifs are in balance, with $\nabla_{\parallel} F$ projecting onto the regulatory motifs. Therefore, new environmental pressures requiring an increase in gene expression would require evolutionary adaptation to increase regulatory motifs. However, the decrease in gene expression required by a subsequent return to the original environment can be accommodated purely by phenotypic plasticity through a decrease in transcription factors. Therefore, one cycle of environmental change leads to a MAM with one-sided memory.

There are many other situations where GDL may apply but where identifying the GDs would require further investigation. Clearly, both biological and artificial neural networks are possibilities. For example, it is possible that the topological connections of neurons in the brain encode history of past experiences, making it easier to, for example, relearn a forgotten language or skill. Gradient discontinuity learning may also play a role in mimetic evolution, or the formation of cultural, social, and political norms. While these examples are highly speculative, they highlight the potential for a far-reaching understanding of memory and learning.

Finally, it is worth putting our results in the context of the growing initiative exploring learning as a unifying concept across fields, which is the subject of this special collection. Specifically, it is important to clearly distinguish between the solving of an individual inverse problem, which we refer to as training or optimizing, from the system approaching a MAM and the emergence of memory. While the process of optimization can vary dramatically in different systems, GDL is agnostic to these details provided they have the effect of approximating some form of gradient descent. In this paper we reserve the word learning to refer to the emergence of memory after cyclic training. A key distinction between this and many other phenomena associated with the word learning is that here the system was never asked to learn. Our learned behavior is not associated with an objective, either explicit or implicit, there is no reward and no bias, and no local rules are intelligently set up to guide a predetermined behavior. Whether or not this distinction is important will surely be decided in the coming years.

ACKNOWLEDGMENTS

We thank Nathan Keim, Aayush Desai, Nicholas Barton, and Gašper Tkačik for important and stimulating discussions. The work was funded by the Institute of Science and Technology Austria.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX A: CALCULATION OF ELASTIC PROPERTIES

The linear elastic properties of a system can be represented by the elastic modulus tensor c_{ijkl}, which describes the second-order terms of the Taylor expansion of the energy with respect to external bulk deformations,

$$\frac{\Delta U}{V^0} = \sigma^0_{ij}\epsilon_{ji} + \frac{1}{2}c_{ijkl}c_{ijkl}\epsilon_{ij}\epsilon_{kl} + O(\epsilon^3),$$

where σ^0 gives the residual stresses of the initial state, ϵ is the strain tensor that describes applied boundary deformations, V^0 is the volume of the initial state, and ΔU is the change in the potential energy of the system.

If a small strain $\epsilon = \gamma\tilde{\epsilon}$ is applied, for some fixed $\tilde{\epsilon}$, the change in the energy is

$$\frac{\Delta U}{V^0} = \tilde{\sigma}^0\gamma + \frac{1}{2V^0}\left(\frac{\partial^2 U}{\partial\gamma^2} - \Xi^T(H^0)^{-1}\Xi\right)\gamma^2 + O(\gamma^3),$$

where $\tilde{\sigma}^0 = \sigma^0_{ij}\tilde{\epsilon}_{ji}$, $\Xi = -\frac{\partial^2 U}{\partial\gamma\partial R}$, and $H^0 = \frac{\partial^2 U}{\partial R^2}$ is the Hessian matrix of the energy with respect to particle positions R. This allows one to determine the elements of c_{ijkl} by calculating $\frac{1}{2V^0}[\frac{\partial^2 U}{\partial\gamma^2} - \Xi^T(H^0)^{-1}\Xi]$ for different $\tilde{\epsilon}$. We use the implementation of this found in the jaxmd library [95]; further details and documentation can be found there.

More common elastic constants like the bulk modulus, shear modulus, and Poisson's ratio are simple functions of the elements of c_{ijkl} [96]. Since any finite disordered system is technically anisotropic, even if it is statistically isotropic in the thermodynamic limit, its elastic response depends on the direction of applied strain. In d dimensions, there are $\frac{d(d+1)}{2} - 1$ independent shear moduli. Thus, for finite systems, we use the so-called angle-averaged Poisson's ratio $\nu = \frac{d-2\frac{\bar{G}}{B}}{2(d-1)\frac{\bar{G}}{B}+d}$, where $\bar{G}$ is the angle-averaged shear modulus

$$\bar{G} = \frac{1}{\pi}\int_0^\pi G(\theta)d\theta,$$

where $G(\theta)$ is the shear modulus corresponding to a shear applied at angle θ. Details of this construction are discussed in [96]. The bulk modulus is calculated in terms of the elastic constants c_{ijkl}, $B = \frac{1}{d^2}\sum_{i,j} c_{iijj}$.

APPENDIX B: GRADIENT-BASED OPTIMIZATION ALGORITHM

Once an energy-minimized configuration is reached for a given initial set of parameters $\boldsymbol{\theta}$, the Poisson's ratio $\nu(\boldsymbol{\theta})$ is calculated via linear response. Given a target Poisson's ratio ν^*, the objective function is defined as the squared deviation $l(\boldsymbol{\theta}) = [\nu(\boldsymbol{\theta}) - \nu^*]^2$. Thus, our aim is to decrease $l(\boldsymbol{\theta})$ to zero using gradient-descent algorithms.

The energy minimization and the calculation of $\nu(\boldsymbol{\theta})$ and $l(\boldsymbol{\theta})$ are implemented in python using the `jaxmd` library [95], which is built upon `jax` [97] and supports automatic differentiation. This framework allows for end-to-end differentiability of the entire calculations, allowing direct calculation of the gradient $\nabla_{\boldsymbol{\theta}} l$.

To accurately and efficiently propagate gradients through the energy minimization step, we employ implicit differentiation, implemented via the `jaxopt` library [98]. This approach mitigates memory overhead typically associated with unrolled gradient computation. The calculated gradients $\nabla_{\boldsymbol{\theta}} l$ are then used to update the parameters $\boldsymbol{\theta}$ using the RMSProp algorithm, as implemented in the optax library [99]. To enhance optimization efficiency, we incorporate a metalearning scheme that dynamically adjusts the learning rate throughout training. The base learning rate is initialized at $lr_0 = 1 \times 10^{-4}$, with a metalearning rate of $lr_m = 3 \times 10^{-4}$.

At each iteration, the final configuration from the previous step is used as the starting configuration for the subsequent energy minimization, thereby maintaining consistency with the given energy minimum throughout training.

APPENDIX C: CYCLIC TRAINING PROTOCOL

As an example, we implement cyclic training between a maximum target Poisson's ratio of $\nu^*_{\text{max}} = 0.5$ and a minimum of $\nu^*_{\text{min}} = 0.0$. The maximum number of optimization steps is set to 3000. The initial configuration is generated by placing particles randomly in the simulation box and quenching to a local energy minimum, which serves as the input state $\boldsymbol{R}(\boldsymbol{\theta})$.

Cyclic training begins with a target value of $\nu^* = \nu^*_{\text{max}} = 0.5$ and proceeds through a sequence of decreasing and then increasing target values. The procedure is outlined as follows.

Step 1. Train the input state $\boldsymbol{R}(\boldsymbol{\theta})$ with the target Poisson's ratio $\nu^*_{\text{max}} = 0.5$, using the optimization algorithm described in Appendix B. This yields the final parameters $\boldsymbol{\theta}_f$ and the corresponding configuration $\boldsymbol{R}$.

Step 2. Using the output from step 1 as input, repeat the optimization with a decreased target value $\nu^* = 0.4$.

Step 3. Continue decreasing the target value in steps (e.g., $\nu^* = 0.3, 0.2, \ldots$), repeating the optimization process until $\nu^* = \nu^*_{\text{min}} = 0.0$ is reached.

Step 4. Reverse the direction by incrementally increasing the target value from $\nu^* = 0.1$ back to $\nu^* = \nu^*_{\text{max}} = 0.5$, using the final configuration from each step as the initial configuration for the next.

Step 5. Evaluate the objective function curves for every optimization in the cycle. If all final objective function values remain below a predefined threshold and all curves are smooth and continuous, i.e., without discontinuities or spikes, the cyclic training is deemed successful. Otherwise, repeat steps 1–4 until either a successful cycle is obtained or the number of cycles exceeds 30.

After successful termination, an additional refinement cycle is performed in which the learning rate is fixed at 10^{-6} for each optimization step to minimize the influence of learning rate adaptation.

For the readout cycle, we run a separate sequence with fixed learning rate $l_r = 10^{-6}$, following the path $\nu^*_{\text{max}} = 0.5 \rightarrow \nu^*_{\text{read}} \rightarrow \nu^*_{\text{max}} = 0.5$, starting from the final configuration and parameters obtained at the end of certain training cycles. For comparison with the cyclic training, we perform a successful single-target training with a target $\nu^*_{\text{max}} = 0.5$, starting from the initial state and parameters. The resulting optimized configuration and parameters are used as the input for the readout cycle, which we defined as "cycle=0" in Fig. 2.

APPENDIX D: ROLE OF THE LOSS THRESHOLD AND LEARNING RATE IN MEMORY FORMATION

To determine whether the memory observed in the main text depends on numerical choices in the optimization procedure, we systematically varied both the objective tolerance l_{tol} used to terminate retraining and the learning rate. We focus on the example shown in Fig. 2(a), with the results presented in Fig. 6. In the main text, retraining or readout was stopped once the objective function fell below 10^{-16}. We find that varying this threshold over several orders of magnitude does not affect the magnitude of the deviations in parameters or particle positions observed in Fig. 2(a). This demonstrates that the behavior is not a consequence of incomplete retraining due to the chosen cutoff.

In contrast, the magnitude of the deviations decreases with decreasing the learning rate l_r, as shown in Fig. 6, indicating that they arise from finite-step-size effects in the optimization dynamics. Note that reducing the learning rate does not qualitatively affect the existence of the MAM.

APPENDIX E: ENSEMBLE-AVERAGED READOUT MEASUREMENTS

To demonstrate the robustness of MAMs and physical memory, Fig. 7 shows results, similar to Fig. 2, for an additional representative system trained between Poisson's ratio $\nu^*_{\text{min}} = 0.0$ and $\nu^*_{\text{max}} = 0.5$, along with ensemble-averaged data over ten independent samples. Figure 7(a) replicates the training protocol used in Fig. 2(a), but starting from a different initial packing configuration. After eight cycles, the system converges to a MAM, and the readout for the final state displays qualitatively similar memory behavior as discussed in the main text. The only notable difference is the presence of contact changes between $\nu^*_{\text{min}} = 0.0$ and $\nu^*_{\text{max}} = 0.5$, which corresponds to a type 1 GD as expected.

Figure 7(b) shows the same data but averaged over ten different systems. Clear kinks and jumps in the various quantities appear at both edges of the training range, highlighting

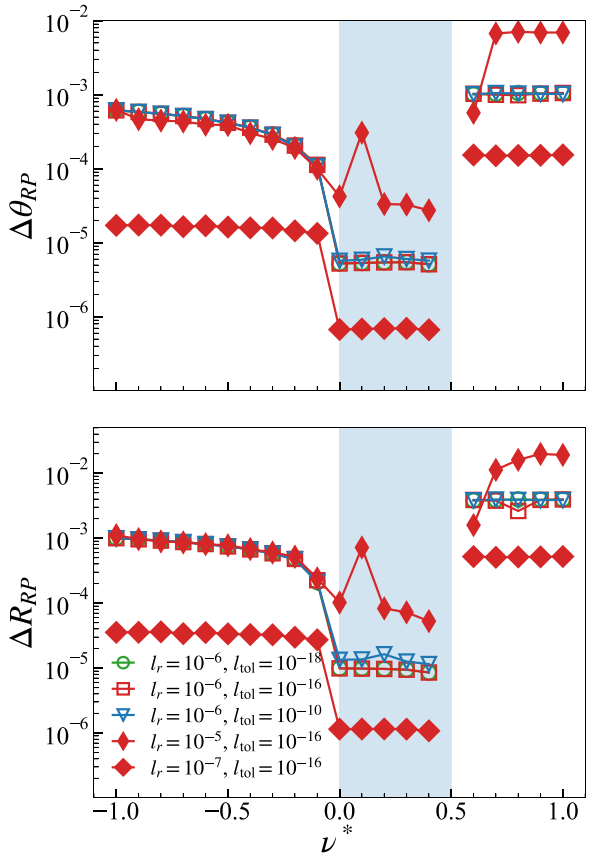


FIG. 6. Robustness of the observed memory. Results for the example shown in Fig. 2(a) are retrained, with varying objective tolerances $l_{\rm tol}$ and learning rates l_r spanning several orders of magnitude. With changing $l_{\rm tol}$, $\Delta\theta_{\rm RP}$ and $\Delta R_{\rm RP}$ after retraining remain unchanged, indicating that the observed memory is not due to incomplete convergence at the chosen cutoff. However, reducing l_r decreases the magnitude of the deviations, demonstrating that they arise from numerical drift in the optimization dynamics.

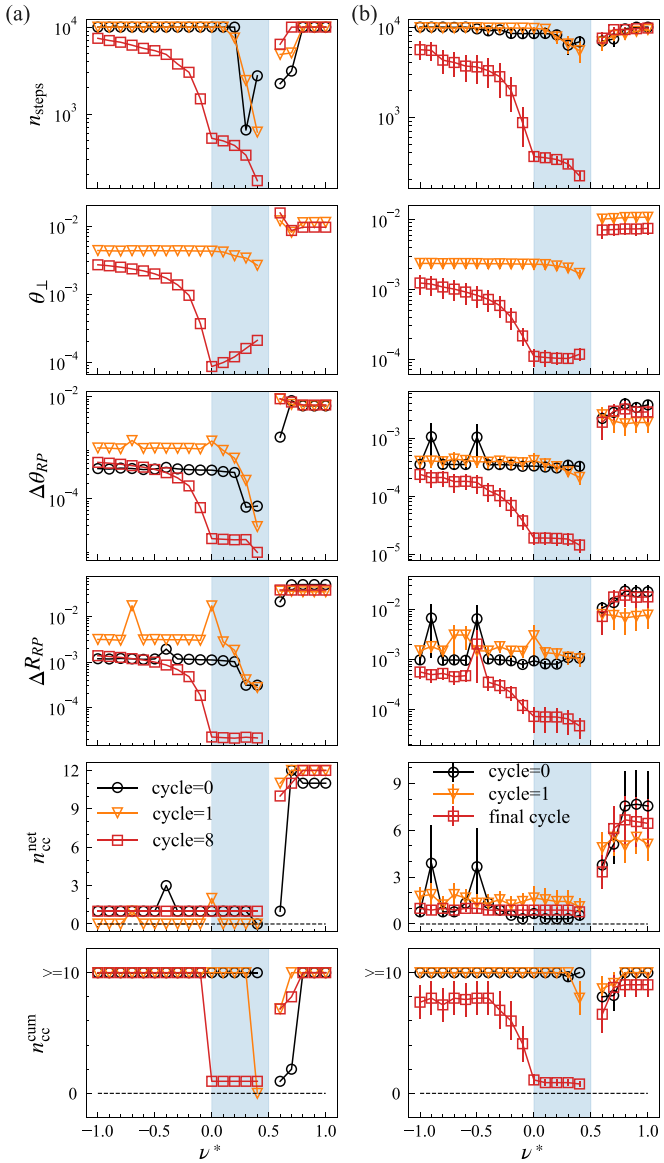


FIG. 7. Robustness of MAMs. Column (a) shows a second representative training example that is identical to Fig. 2(a) except it starts with a different particle configuration. The only qualitative difference in the results is that one contact changes during readout within the training range, corresponding to a type 1 GD, as expected. Column (b) shows averaged measurements over ten independent training samples starting from different initial packings. The error bars represent the standard error on the mean.

that the systems fall into a MAM and encode the training range. The consistency of these signatures across different realizations underscores the robustness of memory formation in disordered packings under cyclic training. These ensemble results emphasize that memory not only is encoded at the individual-system level but is also statistically reproducible across an ensemble.

Moreover, outside the training interval, we observe an asymmetry in the system's ability to adapt: Absorbing states obtained through cyclic training are more readily reoptimized toward lower Poisson's ratio than toward higher ones. Understanding the origin of this asymmetry may be related to the low volume fractions required for systems to reach high values of ν; we leave this exploration to future work.

APPENDIX F: LIMITS FOR A LARGE POISSON'S RATIO

We now discuss the limitations of our model in the regime of large Poisson's ratio. When cyclic training is performed between $\nu^*_{\rm max} = 1.0$ and $\nu^*_{\rm min} = 0.5$, we observe a pronounced decrease in particle diameters. This reduction leads to a significant drop in area fraction from 0.95 to 0.86 as shown in Fig. 8, approaching the jamming threshold $\phi_J = 0.848$. In our model, particles interact via a finite-range purely repulsive potential, which results in more frequent structural rearrangements near the jamming point. This dynamic is reflected in the cumulative number of contact changes $n_{\rm cc}^{\rm cum}$, shown in Fig. 8, which exhibits large fluctuations without a clear trend toward convergence. Consequently, cyclic training in this regime becomes increasingly unstable and fails to reliably reach a marginally absorbing manifold.

Despite these challenges at high Poisson's ratio, we demonstrate that training remains feasible for moderately

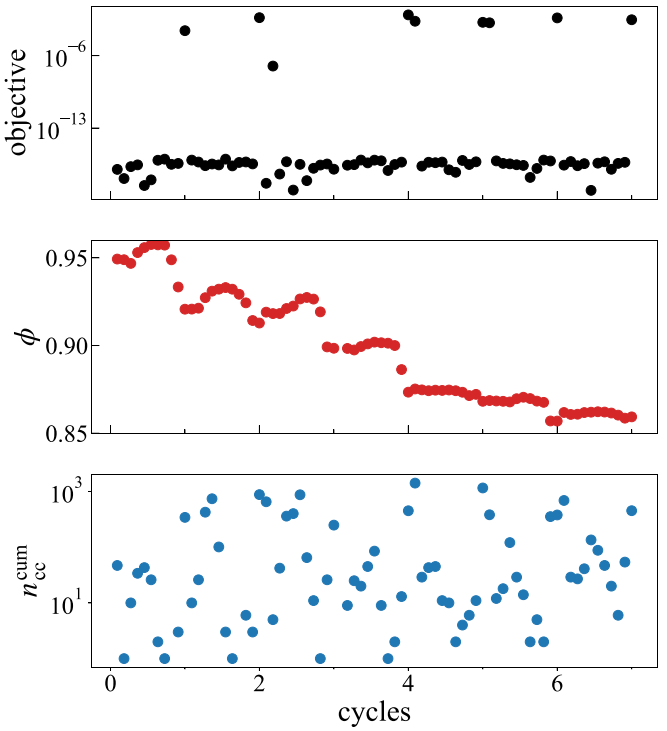


FIG. 8. Cyclicly training the Poisson's ratio between $\nu^*_{max} = 1.0$ and $\nu^*_{min} = 0.5$. From top to bottom, the plot illustrates the evolution of the objective function, measured area fraction ϕ, and the cumulative contact changes during training n_{cc}^{cum}. After seven training cycles, the system loses mechanical stability and becomes unjammed, preventing a well-defined calculation of the Poisson's ratio.

large values. Specifically, as shown in Fig. 9, when cyclic training is performed between $\nu^*_{max} = 0.6$ and $\nu^*_{min} = 0.4$, the system successfully converges to a MAM after approximately 25 cycles and retains a memory of the training range. While the number of optimization steps required to reach $\nu^* = 0.4$ from the initial state obtained via individual training is smaller than that required after cyclic training (black circles), the optimization still involves significant structural rearrangements.

These findings suggest that the model's performance becomes increasingly sensitive to packing constraints at higher ν^*. Further improvements to the optimization procedure, such as incorporating area-fraction constraints directly into the objective function, could enhance stability in this regime.

APPENDIX G: CYCLIC TRAINING ON MULTIPLE FEATURES

In the main text we focus on cyclic training targeting a single mechanical feature, specifically, either the Poisson's ratio or the arbitrarily chosen elastic constant c_{xxyy}. Here we extend the framework to explore cyclic training with multiple features, which leads to more complex and intriguing characteristics of the resulting absorbing manifold.

We take a randomly generated mechanically stable packing of $N = 64$ particles, in which particles are evenly divided into $n_{sp} = 16$ species, such that 16 independent particle diameters serve as tunable parameters. The system is first trained to achieve a target value of the elastic constant c_{xxyy}, beginning

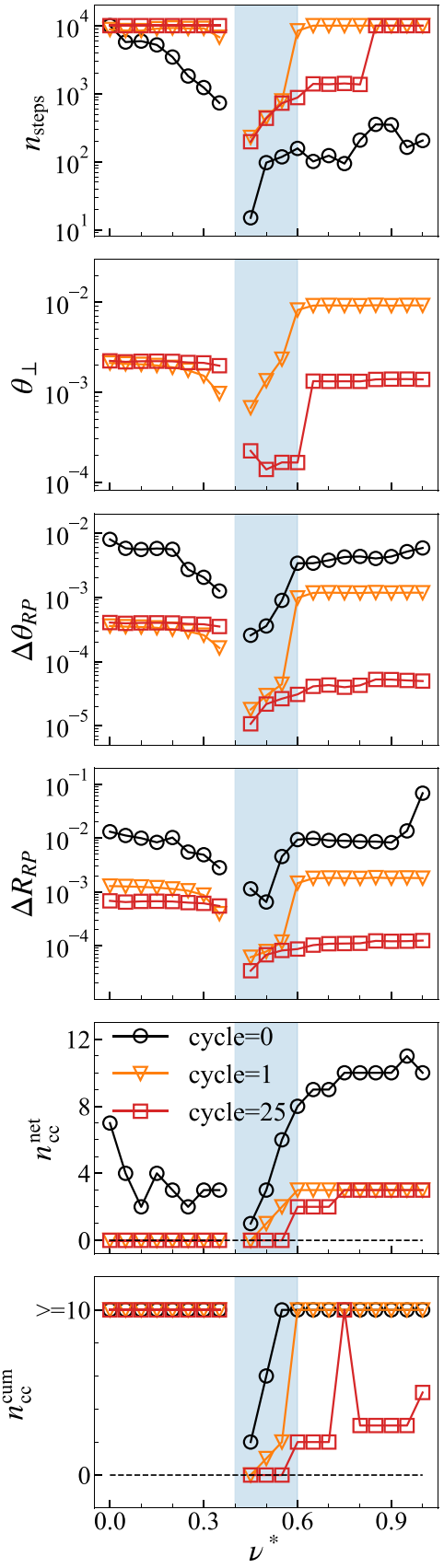


FIG. 9. Memory in a representative example trained on a range between $\nu^*_{max} = 0.6$ and $\nu^*_{min} = 0.4$. The initial Poisson's ratio of the system is $\nu_0 = 0.605$.

at $c^*_{xxyy,\min} = 0.12$. Then training proceeds sequentially by incrementing the target value, using the optimized configuration and parameters from the previous step as the starting point for the next, until reaching $c^*_{xxyy,\max} = 0.16$. We then reverse the training sequence, returning to $c^*_{xxyy,\min} = 0.12$.

After this full cycle of training c_{xxyy}, we switch to a second mechanical feature c_{xxxx} and apply a similar cyclic sequence for the target c^*_{xxxx}: $0.2 \to 0.21 \cdots \to 0.23 \to 0.22 \cdots \to 0.2$. We then alternate back and forth so that a single training cycle involves a complete cycle in each quantity: Training in c_{xxyy} is followed by training in c_{xxxx}, forming a combined cyclic protocol involving both features. Note that while training one of the target features, the other feature is not constrained and is free to vary.

After 15 such combined training cycles, the system converges to a MAM, as shown in Fig. 10(a). To further characterize the structure of this absorbing state, we perform a principal component analysis on the parameter evolution during the final cycle. Interestingly, the results, shown in Fig. 10(b), reveal two distinct trajectories in the space of the first two principal components, each corresponding to training on one of the two target features. This separation suggests that memory encoding for each feature follows a distinct pathway in parameter space. Despite this, the switch from training c_{xxyy} to c_{xxxx} shows a change along PC-1, which may reflect the coupling or competition between these two features. These preliminary results motivate further quantitative investigations into the nature of memory in systems trained for multiple features.

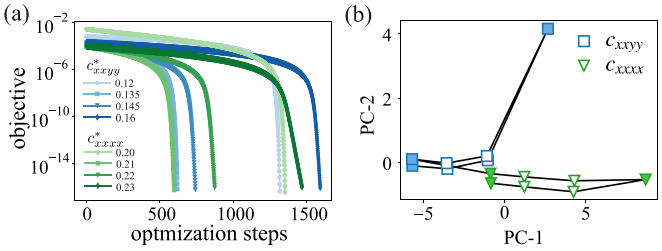


FIG. 10. Cyclic training on two elements of elastic modulus tensor. (a) Objective function during a half cycle of training following 15 combined training cycles. The curves refer to sequential training steps: from $c^*_{xxxx} = 0.2$ to $c^*_{xxyy} = 0.12$, following from $c^*_{xxyy} = 0.12$ to $c^*_{xxyy} = 0.135$ until $c^*_{xxyy} = 0.16$, and then from $c^*_{xxyy} = 0.16$ to $c^*_{xxxx} = 0.2$, up to $c^*_{xxxx} = 0.23$. (b) Projection of the parameter vectors onto the first two principal components for the final cycle. The closed symbols refer to the maximum and minimum targets.

[1] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).

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

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, edited by D. E. Rumelhart, J. L. McClelland, and PDP Research Group (MIT Press, Cambridge, 1986), Vol. 1, pp. 318–362.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (2002).

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, 2012*, edited by P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran, Red Hook, 2012), pp. 1106–1114.

[6] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[7] S. Ruder, An overview of gradient descent optimization algorithms, arXiv:1609.04747.

[8] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. **60**, 223 (2018).

[9] B. Scellier and Y. Bengio, Equilibrium propagation: Bridging the gap between energy-based models and backpropagation, Front. Comput. Neurosci. **11**, 24 (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. Stern and A. Murugan, Learning without neurons in physical systems, Annu. Rev. Condens. Matter Phys. **14**, 417 (2023).

[13] V. Lopez-Pastor and F. Marquardt, Self-learning machines based on Hamiltonian echo backpropagation, Phys. Rev. X **13**, 031020 (2023).

[14] 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).

[15] V. R. Anisetti, A. Kandala, B. Scellier, and J. Schwarz, Frequency propagation: Multimechanism learning in nonlinear physical networks, Neural Comput. **36**, 596 (2024).

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

[17] M. Stern, S. Dillavou, M. Z. Miskin, D. J. Durian, and A. J. Liu, Physical learning beyond the quasistatic limit, Phys. Rev. Res. **4**, L022037 (2022).

[18] L. E. Altman, M. Stern, A. J. Liu, and D. J. Durian, Experimental demonstration of coupled learning in elastic networks, Phys. Rev. Appl. **22**, 024053 (2024).

[19] S. Dillavou, B. D. Beyer, M. Stern, A. J. Liu, M. Z. Miskin, and D. J. Durian, Machine learning without a processor: Emergent learning in a nonlinear analog network, Proc. Natl. Acad. Sci. USA **121**, e2319718121 (2024).

[20] Y. Tang, W. Ye, J. Jia, and Y. Chen, Learning stiffness tensors in self-activated solids via a local rule, Adv. Sci. **11**, 2308584 (2024).

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

[22] C. Arinze, M. Stern, S. R. Nagel, and A. Murugan, Learning to self-fold at a bifurcation, Phys. Rev. E **107**, 025001 (2023).

[23] C. G. Evans, J. O'Brien, E. Winfree, and A. Murugan, Pattern recognition in the nucleation kinetics of non-equilibrium self-assembly, Nature (London) **625**, 500 (2024).

[24] 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).
[25] A. K. Behera, M. Rao, S. Sastry, and S. Vaikuntanathan, Enhanced associative memory, classification, and learning with active dynamics, Phys. Rev. X **13**, 041043 (2023).
[26] J. Veenstra, C. Scheibner, M. Brandenbourger, J. Binysh, A. Souslov, V. Vitelli, and C. Coulais, Adaptive locomotion of active solids, Nature (London) **639**, 935 (2025).
[27] S. Dillavou, M. Guzman, A. J. Liu, and D. J. Durian, Understanding and embracing imperfection in physical learning networks, arXiv:2505.22887.
[28] M. Zu and C. P. Goodrich, Designing athermal disordered solids with automatic differentiation, Commun. Mater. **5**, 141 (2024).
[29] M. Zu, A. Desai, and C. P. Goodrich, Fully independent response in disordered solids, Phys. Rev. Lett. **134**, 238201 (2025).
[30] M. W. Hirsch, C. C. Pugh, and M. Shub, in *Invariant Manifolds*, edited by A. Dold and B. Eckmann, Lecture Notes in Mathematics Vol. 583 (Springer, Berlin, 1977).
[31] N. C. Keim, J. D. Paulsen, Z. Zeravcic, S. Sastry, and S. R. Nagel, Memory formation in matter, Rev. Mod. Phys. **91**, 035002 (2019).
[32] J. D. Paulsen and N. C. Keim, Mechanical memories in solids, from disorder to design, Annu. Rev. Condens. Matter Phys. **16**, 61 (2025).
[33] M. Mungan, E. Clément, D. Vandembroucq, and S. Sastry, Self-organization, memory, and learning: From driven disordered systems to living matter, Annu. Rev. Condens. Matter Phys. **17**, 207 (2026).
[34] J. Gill, Transient non-linear phenomena and metastable states of charge-density waves in niobium triselenide, Solid State Commun. **39**, 1203 (1981).
[35] R. M. Fleming and L. F. Schneemeyer, Transient electrical response of $K_{0.30}MoO_3$, Phys. Rev. B **28**, 6996 (1983).
[36] K. Matan, R. B. Williams, T. A. Witten, and S. R. Nagel, Crumpling a thin sheet, Phys. Rev. Lett. **88**, 076101 (2002).
[37] Y. Lahini, O. Gottesman, A. Amir, and S. M. Rubinstein, Nonmonotonic aging and memory retention in disordered mechanical systems, Phys. Rev. Lett. **118**, 085501 (2017).
[38] E. M. Bertin, J. P. Bouchaud, J. M. Drouffe, and C. Godrche, The Kovacs effect in model glasses, J. Phys. A Math. Gen. **36**, 10701 (2003).
[39] L. F. Cugliandolo, G. Lozano, and H. Lozza, Memory effects in classical and quantum mean-field disordered models, Eur. Phys. J. B **41**, 87 (2004).
[40] D. Fiocco, G. Foffi, and S. Sastry, Encoding of memory in sheared amorphous solids, Phys. Rev. Lett. **112**, 025702 (2014).
[41] N. C. Keim and P. E. Arratia, Mechanical and microscopic properties of the reversible plastic regime in a 2D jammed material, Phys. Rev. Lett. **112**, 028302 (2014).
[42] M. Mungan, S. Sastry, K. Dahmen, and I. Regev, Networks and hierarchies: How amorphous materials learn to remember, Phys. Rev. Lett. **123**, 178002 (2019).
[43] N. C. Keim, J. Hass, B. Kroger, and D. Wieker, Global memory from local hysteresis in an amorphous solid, Phys. Rev. Res. **2**, 012004(R) (2020).
[44] M. Adhikari and S. Sastry, Memory formation in cyclically deformed amorphous solids and sphere assemblies, Eur. Phys. J. E **41**, 105 (2018).
[45] C. W. Lindeman and S. R. Nagel, Multiple memory formation in glassy landscapes, Sci. Adv. **7**, eabg7133 (2021).
[46] I. Regev, I. Attia, K. Dahmen, S. Sastry, and M. Mungan, Topology of the energy landscape of sheared amorphous solids and the irreversibility transition, Phys. Rev. E **103**, 062614 (2021).
[47] M. O. Lavrentovich, A. J. Liu, and S. R. Nagel, Period proliferation in periodic states in cyclically sheared jammed solids, Phys. Rev. E **96**, 020101(R) (2017).
[48] M. D. Haw, W. C. K. Poon, P. N. Pusey, P. Hebraud, and F. Lequeux, Colloidal glasses under shear strain, Phys. Rev. E **58**, 4673 (1998).
[49] S. Mukherji, N. Kandula, A. K. Sood, and R. Ganapathy, Strength of mechanical memories is maximal at the yield point of a soft glass, Phys. Rev. Lett. **122**, 158001 (2019).
[50] J. R. Royer and P. M. Chaikin, Precisely cyclic sand: Self-organization of periodically sheared frictional grains, Proc. Natl. Acad. Sci. USA **112**, 49 (2015).
[51] J. Ren, J. A. Dijksman, and R. P. Behringer, Reynolds pressure and relaxation in a sheared granular system, Phys. Rev. Lett. **110**, 018302 (2013).
[52] M. Toiya, J. Stambaugh, and W. Losert, Transient and oscillatory granular shear flow, Phys. Rev. Lett. **93**, 088001 (2004).
[53] D. Fiocco, G. Foffi, and S. Sastry, Memory effects in schematic models of glasses subjected to oscillatory deformation, J. Phys.: Condens. Matter **27**, 194130 (2015).
[54] Y. Chen, Q. Zhang, S. Ramakrishnan, and R. L. Leheny, Memory in aging colloidal gels with time-varying attraction, J. Chem. Phys. **158**, 024906 (2023).
[55] Y. Chen, S. A. Rogers, S. Narayanan, J. L. Harden, and R. L. Leheny, Microstructural and rheological training and memory of nanocolloidal soft glasses under cyclic shear, Phys. Rev. Mater. **9**, 025601 (2025).
[56] D. J. Pine, J. P. Gollub, J. F. Brady, and A. M. Leshansky, Chaos and threshold for irreversibility in sheared suspensions, Nature (London) **438**, 997 (2005).
[57] L. Corté, P. M. Chaikin, J. P. Gollub, and D. J. Pine, Random organization in periodically driven systems, Nat. Phys. **4**, 420 (2008).
[58] G. I. Menon and S. Ramaswamy, Universality class of the reversible-irreversible transition in sheared suspensions, Phys. Rev. E **79**, 061108 (2009).
[59] N. C. Keim and S. R. Nagel, Generic transient memory formation in disordered systems with noise, Phys. Rev. Lett. **107**, 010603 (2011).
[60] N. C. Keim, J. D. Paulsen, and S. R. Nagel, Multiple transient memories in sheared suspensions: Robustness, structure, and routes to plasticity, Phys. Rev. E **88**, 032306 (2013).
[61] P. Pham, B. Metzger, and J. E. Butler, Particle dispersion in sheared suspensions: Crucial role of solid-solid contacts, Phys. Fluids **27**, 051701 (2015).
[62] J. D. Paulsen, N. C. Keim, and S. R. Nagel, Multiple transient memories in experiments on sheared non-Brownian suspensions, Phys. Rev. Lett. **113**, 068301 (2014).
[63] D. R. Scheff, S. A. Redford, C. Lorpaiboon, S. Majumdar, A. R. Dinner, and M. L. Gardel, Actin filament alignment causes mechanical hysteresis in cross-linked networks, Soft Matter **17**, 5499 (2021).
[64] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, Structural relaxation made simple, Phys. Rev. Lett. **97**, 170201 (2006).

[65] J. Guénolé, W. G. Nöhring, A. Vaid, F. Houllé, Z. Xie, A. Prakash, and E. Bitzek, Assessment and optimization of the fast inertial relaxation engine (FIRE) for energy minimization in atomistic simulations and its implementation in LAMMPS, Comput. Mater. Sci. **175**, 109584 (2020).

[66] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, Automatic differentiation in machine learning: A survey, J. Mach. Learn. Res. **18**, 1 (2018).

[67] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, Nature (London) **323**, 533 (1986).

[68] R. E. Wengert, A simple automatic derivative evaluation program, Commun. ACM **7**, 463 (1964).

[69] M. J. Falk, J. Wu, A. Matthews, V. Sachdeva, N. Pashine, M. L. Gardel, S. R. Nagel, and A. Murugan, Learning to learn by using nonequilibrium training protocols for adaptable materials, Proc. Natl. Acad. Sci. USA **120**, e2219558120 (2023).

[70] Note that these spikes are not observed in Ref. [28]. As will become clear in Sec. IV, this is due to the smooth attractive potential used in that paper.

[71] While we do not fully understand this, it is worth noting that n_{steps} is the weakest of our measurements since it depends on the details of the optimization algorithm, which are not controlled for.

[72] In practice, the finite step size of any iterative ascent/descent algorithm will always result in some small amount of parameter drift.

[73] Note that the red descent path happens to start very slightly to the left of the GD, so it flows leftward, but it could just as easily have started to the right of the GD and flowed in the opposite direction, depending on where in the oscillation the ascent path stops.

[74] Note that $\Delta\Theta_{\text{RP}}$ is the distance moved along the $F = 0$ contour, which increases continuously from zero when training is pushed beyond F^*_{max}. This explains the smooth increases in the readout measurements observed in Fig. 2.

[75] Lower-dimensional GDs are of course possible, but they are probably much less important due to the smaller chance of encountering them. They do not appear in our sphere packings, and we do not consider them further.

[76] M. Adhikari, R. Sharma, and S. Karmakar, Encoding fast and fault-tolerant memories in bulk and nanoscale amorphous solids, Phys. Rev. Lett. **134**, 018202 (2025).

[77] C. W. Lindeman, Multi-dimensional memory in low-friction granular materials, Soft Matter **21**, 4890 (2025).

[78] F. Arceri, E. I. Corwin, and V. F. Hagh, Marginal stability in memory training of jammed solids, Phys. Rev. E **104**, 044907 (2021).

[79] J. Kaiser, An investigation into the occurrence of noises in tensile tests, Ph.D. thesis, University of California, Berkeley, 1964.

[80] L. Mullins, Effect of stretching on the properties of rubber, Rubber Chem. Technol. **21**, 281 (1948).

[81] A. Mayer, T. Mora, O. Rivoire, and A. M. Walczak, Transitions in optimal adaptive strategies for populations in fluctuating environments, Phys. Rev. E **96**, 032412 (2017).

[82] P. Alberch, From genes to phenotype: Dynamical systems and evolvability, Genetica **84**, 5 (1991).

[83] M. C. Cowperthwaite and L. A. Meyers, How mutational networks shape evolution: Lessons from RNA models, Annu. Rev. Ecol. Evol. Syst. **38**, 203 (2007).

[84] M. Pigliucci, Genotype–phenotype mapping and the end of the 'genes as blueprint' metaphor, Philos. Trans. R. Soc. B **365**, 557 (2010).

[85] S. E. Ahnert, Structural properties of genotype–phenotype maps, J. R. Soc. Interface **14**, 20170275 (2017).

[86] S. M. Scheiner, Genetics and evolution of phenotypic plasticity, Annu. Rev. Ecol. Syst. **24**, 35 (1993).

[87] D. J. Begun, A. K. Holloway, K. Stevens, L. W. Hillier, Y.-P. Poh, M. W. Hahn, P. M. Nista, C. D. Jones, A. D. Kern, C. N. Dewey, L. Pachter, E. Myers, and C. H. Langley, Population genomics: Whole-genome analysis of polymorphism and divergence in Drosophila simulans, PLoS Biol. **5**, e310 (2007).

[88] H. A. Orr, Fitness and its role in evolutionary genetics, Nat. Rev. Genet. **10**, 531 (2009).

[89] L. Keren, J. Hausser, M. Lotan-Pompan, I. V. Slutskin, H. Alisar, S. Kaminski, A. Weinberger, U. Alon, R. Milo, and E. Segal, Massively parallel interrogation of the effects of gene expression levels on fitness, Cell **166**, 1282 (2016).

[90] S.-A. A. Chen, A. F. Kern, R. M. L. Ang, Y. Xie, and H. B. Fraser, Gene-by-environment interactions are pervasive among natural genetic variants, Cell Genom. **3**, 100273 (2023).

[91] S. P. Carroll, A. P. Hendry, D. N. Reznick, and C. W. Fox, Evolution on ecological time-scales, Funct. Ecol. **21**, 387 (2007).

[92] R. S. Staron, M. J. Leonardi, D. L. Karapondo, E. S. Malicky, J. E. Falkel, F. C. Hagerman, and R. S. Hikida, Strength and skeletal muscle adaptations in heavy-resistance-trained women after detraining and retraining, J. Appl. Physiol. **70**, 631 (1991).

[93] L. M. Schwartz, Muscle nuclei remember to cheat death, J. Physiol. **591**, 6133 (2013).

[94] W.-C. Ho, D. Li, Q. Zhu, and J. Zhang, Phenotypic plasticity as a long-term memory easing readaptations to ancestral environments, Sci. Adv. **6**, eaba3388 (2020).

[95] S. S. Schoenholz and E. D. Cubuk, in *Proceedings of the 34th Conference on Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran, Red Hook, 2020), pp. 11428–11441.

[96] C. P. Goodrich, S. Dagois-Bohy, B. P. Tighe, M. van Hecke, A. J. Liu, and S. R. Nagel, Jamming in finite systems: Stability, anisotropy, fluctuations, and scaling, Phys. Rev. E **90**, 022138 (2014).

[97] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python+NumPy programs, GitHub, San Francisco, 2018, http://github.com/jax-ml/jax.

[98] M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert, in *Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, 2022*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran, Red Hook, 2022), pp. 5230–5242.

[99] DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, *et al.*, The DeepMind JAX ecosystem, GitHub, San Francisco, 2020, https://github.com/google-deepmind/optax.