PNAS RESEARCH ARTICLE | APPLIED PHYSICAL SCIENCES

Learning to learn by using nonequilibrium training protocols for adaptable materials

Martin J. Falk[a,1], Jiayi Wu[a,1], Ayanna Matthews[b], Vedant Sachdeva[b], Nidhi Pashine[c], Margaret L. Gardel[a,d,e,f], Sidney R. Nagel[a,d], and Arvind Murugan[a,d,2]

Edited by David Weitz, Harvard University, Cambridge, MA; received November 19, 2022; accepted May 25, 2023

Evolution in time-varying environments naturally leads to adaptable biological systems that can easily switch functionalities. Advances in the synthesis of environmentally responsive materials therefore open up the possibility of creating a wide range of synthetic materials which can also be trained for adaptability. We consider high-dimensional inverse problems for materials where any particular functionality can be realized by numerous equivalent choices of design parameters. By periodically switching targets in a given design algorithm, we can teach a material to perform incompatible functionalities with minimal changes in design parameters. We exhibit this learning strategy for adaptability in two simulated settings: elastic networks that are designed to switch deformation modes with minimal bond changes and heteropolymers whose folding pathway selections are controlled by a minimal set of monomer affinities. The resulting designs can reveal physical principles, such as nucleation-controlled folding, that enable such adaptability.

adaptability | materials training | inverse design | disordered materials

Significance

Biological systems are distinguished by "metaproperties" such as robustness or multifunctionality. We show that the metaproperty of adaptability naturally emerges when tunable synthetic materials are trained for different incompatible functions in sequence. By switching training goals, materials localize to special regions of their high-dimensional design spaces, where they have learned to be rapidly adaptive to a changing environment. Examining the resulting materials can reveal physical principles underlying material adaptability. This way of training works on top of existing design methods and can be applied across a wide array of materials, real or simulated. We demonstrate our method in two different simulation contexts: disordered elastic structures and heteropolymers with tunable interactions.

Considered as materials, biological systems are striking in their ability to perform many individually demanding tasks in contexts that can often change over time. This success can be attributed to "metaproperties" like modularity (1–5), robustness (6), plasticity for learning (7), and multifunctionality (8–11). While inverse materials design has sought to optimize specific properties (12–23), less attention has been given to identifying general design strategies for creating materials with metaproperties.

Here, we show how a biologically inspired design method can target one such metaproperty, adaptability. By adaptability, we mean the ability to switch between mutually incompatible functions with minimal changes in design parameters. For example, consider the Poisson's ratio of an elastic network, which is a unique number which characterizes the global deformation of a material in response to small uniaxial strains. An adaptable elastic network could switch from a negative Poisson's ratio to a positive one with minimal network changes, even though a given network can only have one Poisson's ratio. In this example, the mutually incompatible functions are the different Poisson's ratios, and the design parameters are the stiffnesses of the network bonds. A truly adaptable material will be as good as a nonadaptable material at any given function but will require fewer modifications to produce a distinct, incompatible function.

At first glance, the existence of a truly adaptable material seems highly improbable. However, if the design space of the material is high-dimensional, then we should generically expect that there are many distinct choices of design parameters with equivalent performance for a given function (1, 19, 24–28). Our goal is to identify the much more rare subsets of design solutions which both perform the given function and are adaptable.

In our approach, we take existing optimization algorithms for a target function and repeatedly switch the target before optimization is completed for any one function. The partially adapted design parameters for one function are used as initial conditions for optimizing the second function. This intuitively requires the solutions identified in successful periods of training to drift closer to each other in design space with each switch (Fig. 1).

The underlying logic of this approach is that the sets of design parameters which survive the oscillating selection process are required to be similar by construction, even if their yield is lower. The existence of similar design parameter sets then implies that there are shared design characteristics between the solutions, even though the functions they perform are incompatible.

Author contributions: M.J.F., J.W., M.L.G., S.R.N., and A. Murugan designed research; M.J.F., J.W., A. Matthews, V.S., and N.P. performed research; M.J.F., J.W., and A. Matthews analyzed data; and M.J.F., J.W., A. Matthews, S.R.N., and A. Murugan wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]M.J.F. and J.W. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: amurugan@uchicago.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2219558120/-/DCSupplemental.

Published June 26, 2023.

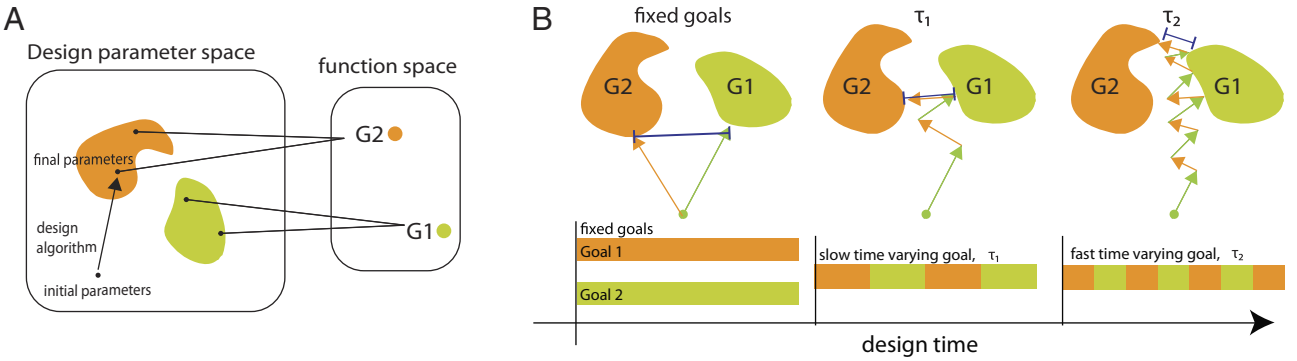

Fig. 1. Evolving design parameters toward different target functionalities over time can select for highly adaptable materials. (*A*) Materials can be characterized by their design parameters (e.g., for an elastic network, the rest length and stiffness of each bond are design parameters). The design parameter-to-function map is often highly degenerate; many choices of design parameters (e.g., orange region) can achieve the same equivalent function (e.g., *G2*). Some parts of the orange region might be close to design parameters (green) that achieve an alternative function *G2*; such regions can often be vanishingly small but correspond to materials that can quickly adapt from exhibiting function *G1* to *G2* with minimal changes. We have drawn regions for each goal as simply connected regions, but the topology may be more complex. (*B*) Optimization for a single, fixed goal either *G1* or *G2* will typically result in nonadaptable materials that are good at function *G1* or *G2* but not adaptable. Switching between the two goals on a fixed timescale τ, even if the parameters are not yet well adapted to the current target goal, by construction selects for parameters that are closer together in parameter space. Faster switching selects for closer parameter sets that might be rarer or might not exist, leading to a trade-off between yield and adaptability.

We note that our approach does not directly optimize a metric of adaptability. Instead, material adaptability arises because of the sequence of selection pressures the material is subject to during design optimization. Our method functions analogously to a software "wrapper": a procedure which coordinates the deployment of existing optimization algorithms. It is therefore compatible with a wide range of existing material design procedures, ranging from fully computer-based (12–14) to fully in situ (29, 30). Our work extends intuition developed earlier on modularity (1, 2) in biological contexts to canonical synthetic materials platforms. However, in these prior works, the tasks under selective pressure were chosen to be modular. The resulting system reflected the modules specified by the selective environment: e.g., with logic circuits (1), the environment switched between selecting for computing an AND operation between two subgoals, or an OR operation between those same two subgoals. Consequently, the resulting logic circuits developed modules for computing subgoals which could be quickly recombined to achieve AND or OR with minimal changes.

In contrast, in many problems relevant to materials, the different goals or functions required may not have any obvious modular structure. For example, consider two goals *G1*, *G2* representing a material with different Poisson's ratios, an elastic network with incompatible long-range motions, or a polymer folding into two distinct structures with no common substructures. In this work, we focus on such arbitrary goals that are not chosen to be modular in any obvious sense. We will nevertheless use the alternating selection paradigm of prior biological works; we find that such design protocols can reveal adaptable organization of materials that can be rationalized in retrospect, even for goals not organized in any obvious modular manner.

We demonstrate the utility of this method in the context of three simulated systems—a) elastic networks with locally tunable moduli being trained to exhibit allostery, i.e., targeted long-range coupling between local deformations, b) elastic networks undergoing irreversible connectivity changes for targeted Poisson's ratio, and c) self-assembling heteropolymers with monomer interactions tuned for folding into distinct structures. In each system, rapidly oscillating training goals allow us to find design parameters which can switch between mutually exclusive functions with minimal parameter changes. In the self-assembly case, we gain physical insight into the origin of adaptability, as selecting for adaptability localizes parameter changes to interaction units which control kinetic barriers in the folding landscape. Similarly, in the elastic networks, we find that adaptability arises from a coherent displacement unit which is easily shifted to perform opposing allosteric motions. Thus, our work suggests a broad strategy to identify physical mechanisms that enable adaptability in materials with arbitrary underlying physics.

Results

Elastic Networks. In the context of mechanical materials, we first focus on allosteric response, that is, the ability of an elastic network to exhibit a desired strain at a distant target site if strained at a specific source site. Such allosteric responses have been created through multiple methods (2, 31–35). Typically, there are multiple design parameters that give rise to an allosteric response, which means that the design space is degenerate with respect to that particular allosteric motion. Our goal is to search through this degenerate design space for the smaller set of parameters that can perform a specific, different, incompatible allosteric response with minimal adjustment.

We use a 2-d mass-spring network to model allostery in a mechanical system. Specifically, we simulate a 22-node hexagonal lattice with fixed boundary conditions. While the geometry and rest lengths of all springs are fixed, the collection of 83 spring constants $K = \{k_i\}_{i=1}^{83}$ are design parameters that can be tuned to get specific allosteric responses. The two motions we train for, *G1* and *G2*, correspond to two opposite responses for the same input at the same source site (Fig. 2*A*). We first reproduce such single-function design by starting from many random sets of springs constants K and running a gradient descent procedure on a cost function related to the dynamical matrix.

Our cost function rewards the softest mode transmitting strain from source to target while also rewarding a large energy gap between such a soft mode and the next mode; see *SI Appendix* for further details. Other works have created such allosteric materials using a range of different cost functions and optimization procedures (31, 32). Our procedure, while different in detail from previous work, nevertheless consistently produces networks with the desired allosteric response.

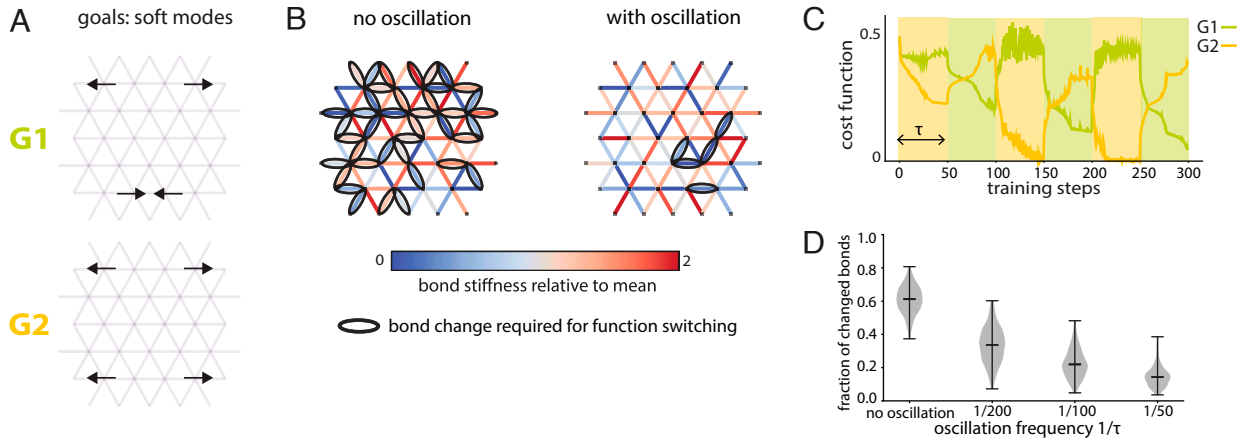

Fig. 2. Oscillatory training protocols generate adaptable solutions for allostery in elastic networks. (*A*) We perform training on elastic networks with tunable bond stiffnesses. Goals *G*1 and *G*2 seek extensile and contractile strains respectively at a pair of target nodes in response to strain at a pair of input nodes. Target nodes, applied strain, and input nodes are the same for *G*1 and *G*2. (*B*) Each successful run of training generates a pair of elastic networks; one performs *G*1, and the other performs *G*2. We show the network which performs *G*1, bonds colored by relative stiffness (normalized to network bond stiffness mean). Relative bond stiffnesses which change by more than 0.1 to switch to *G*2 circled in black. Network examples from training without oscillations (*Left*) and with oscillations (*Right*). (*C*) Performance on each goal *G*1, *G*2 quantified by a cost function. Lower cost function indicates improved performance. Training drives cost function of on-target goal lower during each period. Background color panels indicate on-target goal. τ indicates training steps per period of goal oscillation. (*D*) Faster oscillation (smaller τ) during training gives networks with higher adaptability (defined as fraction of all relative bond stiffnesses which change beyond 0.1 threshold when switching between *G*1 and *G*2). Violin plots show distribution of changed bond fraction over successfully trained network pairs; black lines indicate minimum, mean, and maximum values (n = 500 for all distributions).

Crucially, we find that there are numerous choices of design parameters K—here, bond stiffnesses—that can separately perform each of the goals *G*1 and *G*2. One of these networks for *G*1 is shown in Fig. 2 *B*, *Left*; the others are different in the choice of K but are equivalent in terms of performance at *G*1.

To leverage this degeneracy in design parameters for adaptability, we studied a family of algorithms in which the target goal is switched periodically between *G*1 and *G*2 during design optimization at different timescales τ, where τ is the number of optimization steps per period. In this way, design parameters partially optimized for one goal, say *G*1, are used as initial conditions for the next period of design that targets *G*2. Following our intuition in Fig. 1, any two sets of design parameters produced consecutively in this way are likely to be similar.

The results of switching at different frequencies are shown in Fig. 2 *B*–*D*. We find that such oscillatory design has two distinct phases. Initially, the design parameters are not good at either goal *G*1, *G*2 (Fig. 2*C*). After this phase, we find one of two outcomes: a) success, i.e., convergence to a limit cycle between a pair of design parameters (bond stiffnesses) K_1 and K_2 that are good at *G*1 and *G*2, respectively (Fig. 2*C*) or b) failure, i.e., convergence to design parameters that are good at only one or neither property.

In successful cases, we can measure adaptability of the pair K_1, K_2 as the number of bonds that need to change their elastic constant k_{ij} by more than 0.1 when normalized by the mean stiffness of the network.

We repeat simulations for 500 random initial assignments of bond stiffness for each of 4 different oscillation timescales τ (Fig. 2*D*). We see a substantial and systematic increase in adaptability with frequency of switching, when restricting to successful runs.

Intuitively, for more rapid switching times, when the process does converge on a limit cycle, the pair K_1, K_2 are closer, as they must be since there is less design time to get from one to another. In the limit of extremely rapid switching times, the differences between the two solutions will vanish, as there is no design time to make changes to each solution.

For example, using an oscillation timescale $\tau = 50$, we identify networks that can switch function by changing as few as 5 bonds (Fig. 2 *B*, *Right*). In contrast, networks obtained by optimizing for *G*1 or *G*2 alone typically differ significantly in 50 bonds (Fig. 2, *B*, *Left* and *D*).

Physical interpretation. To better understand how adaptability arises under oscillatory training, we optimized several larger networks. While in small networks, it was more difficult to identify physical principles in the optimized networks, in larger networks, the mechanical signatures of allostery were visually clear. In one example network, we found that oscillatory training produced a section of the network which moved coherently. This section is diverted with just a small number of bond stiffness changes, shifting from an in-phase to an out-of-phase motion. Surprisingly, training for only one motion at a time also produced networks with coherent motions. However, the coherent motions selected for in the nonoscillatory training differed between the *G*1 and *G*2 goals (*SI Appendix*, Fig. 1*A*). We quantified this effect through the overlap of displacement patterns in the lowest energy modes, finding that patterns between adaptable pairs had a significantly higher overlap compared to nonadaptable pairs' overlap (*SI Appendix*, Fig. 1*B*).

Local learning rules. We showed that adaptable allosteric response can be created in elastic networks by alternating training for two incompatible motions (Fig. 2). This training relied on the optimization of a global cost function under gradient descent. However, recent examples of elastic network training aim to change bulk elastic moduli with algorithms which use local information as input (36–38) and modify the network in an irreversible fashion (39, 40). When training for adaptability in mechanical allostery, we implicitly assumed that the dynamics of training would allow for returns to previously visited regions of design parameter space. It is not clear that we can train for adaptability without the ability to move unrestricted through design parameter space. Here, we show that we can extend our oscillatory training framework to the task of developing bulk elastic response even with irreversible local update rules, at the cost of decreased yield.

Our two target goals $G1$ and $G2$ now correspond to having Poisson's ratios of > 0.75 and < -0.75, respectively (Fig. 3*A*). We will use the same notation as in the previous section—$G1$ and $G2$—to refer to these goals. The Poisson's ratio ν of a material describes its bulk deformation response to a uniaxial strain. If the applied strain is compressive, a negative ν indicates that a network will contract along the axis orthogonal to the strain, while a positive ν indicates that a network will expand along the orthogonal axis. Note that for isotropic materials in two dimensions, ν is constrained to be within $[-1, 1]$, but here, we consider materials which may become anisotropic as they are trained.

Our algorithm for training elastic networks to perform $G1$ and $G2$ proceeds by the irreversible removal of bonds based on local information. Our design parameters are the presence or absence of a bond in the network, but we do not allow bond additions. We initialize our 500 training simulations with 2D mass-spring networks of approximately 200 nodes obtained from positionally disordered jammed packings. We simulate these networks under periodic boundary conditions. During training, we enforce a deformation on the network, measure the strain in each bond, and then remove the bond which experiences the most strain. When we train for positive ν, we compress the network along the y-axis and stretch it along the x-axis. Analogously, we train for negative ν by compressing the network along the y-axis while also compressing it along the x-axis. During oscillatory training, we alternate which of these deformations is applied. See *SI Appendix* for further detail.

Despite the differences between the goals and algorithms considered here compared to those used in the design of adaptable allostery, we find that a similar picture of adaptable mechanical design under oscillatory training emerges (Fig. 3 *B*–*D*).

Even when trained from the same initial network, training for $G1$ or $G2$ separately produces pairs of networks with many differences in their bond removals (Fig. 3 *B*, *Left*). In contrast, when we oscillate which deformation is applied every 20 bond removals, we find that the difference between such network pairs is 50% to 70% lower than the difference in network pairs trained without oscillation (Fig. 3 *B*, *Right* and *D*).

However, in an ensemble of networks undergoing oscillatory training, 82% experience mechanical failure before the training ends, due to the irreversible nature of bond removals. Of those that survive training, 29% are able to rapidly switch between $G1$ and $G2$, with an overall yield of 5% (Fig. 3*C*). When comparing an ensemble of networks trained with oscillation to an ensemble of networks trained without, we find quantitative evidence that adaptability increases with oscillatory training, at the cost of lower yield (Fig. 3*D*). This trade-off between adaptability and yield is also observed in our allostery training, but the irreversible training algorithm lowers the overall scale of the yields for our Poisson's ratio training in comparison to the allosteric case.

Heteropolymer Folding. Having demonstrated our method's success in designing adaptable mechanical networks, we turn to another paradigmatic class of tunable synthetic materials. Programmable self-assembly of single target structures has been explored in many systems, ranging from colloids to proteins and DNA. Across these diverse systems, a similar set of design parameters are tuned to target assembly of a desired structure. Typically, these parameters include a matrix of binding affinities between building blocks, in addition to global parameters like temperature and concentrations. We refer to the matrix of binding affinities as the affinity matrix.

In most approaches to self-assembly (41, 42), the affinity matrix closely resembles the contact matrix of the building blocks in the desired structure Γ. That is, particles in contact in Γ should typically have stronger binding affinities compared to particles not in contact in Γ, thereby energetically stabilizing the structure relative to other configurations of the same particles.

As a result, design parameters optimal for assembling a structure Γ_a would not be good at assembling an unrelated structure Γ_b with high yield. This makes adaptability in self-assembly seem difficult from the outset. The stochastic nature of self-assembly provides an additional complication compared to elastic networks.

To test whether we can design a self-assembling system to be adaptable, we built a simulation of 2-d heteropolymer folding using the HOOMD-blue software (43). Specifically, we consider a polymer of 13 monomers, each of which is bonded to the next with harmonic springs. A harmonic bending energy is present to stabilize the fully extended polymer state with a persistence length of 5.5. Each monomer interacts with all non-neighbor

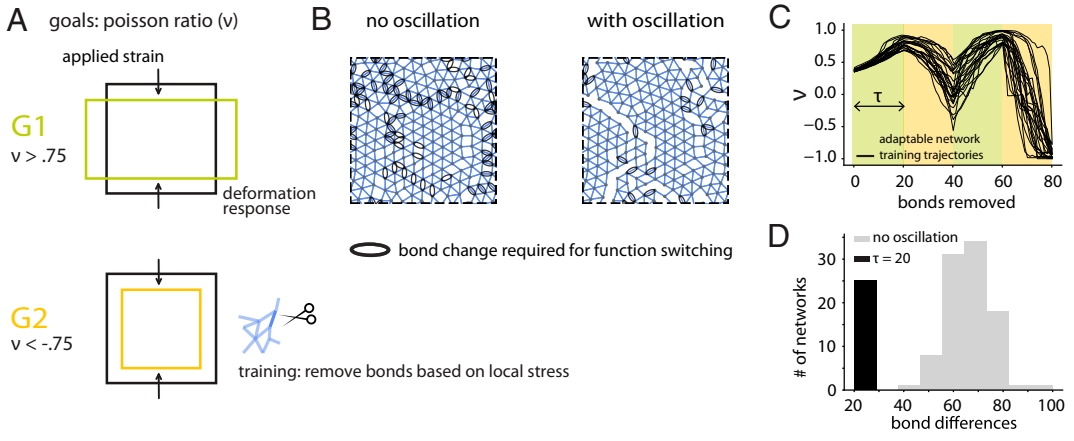

Fig. 3. Oscillatory training protocols generate networks with adaptable Poisson's ratios. (*A*) 2D elastic networks with disordered geometry are trained for Poisson's ratios $\nu > 0.75$ ($G1$) or $\nu < -0.75$ ($G2$). During training, bonds are irreversibly removed based on local stress in response to applied strain. (*B*) Each successful run of training generates a pair of elastic networks; one network performs $G1$, and the other performs $G2$. We show the network which performs $G1$. Bond changes required for switching to $G2$ are circled in black. Network examples from training without oscillations (*Left*) and with oscillations (*Right*). (*C*) An ensemble of 500 networks undergoing oscillatory training with bond removal results in successful switching between $G1$ and $G2$ in 25 networks. Background color panels indicate on-target goal. τ indicates bonds removed per period of goal oscillation. (*D*) Successful networks trained with oscillation ($n = 25$) have a bond difference of 20 bonds. Networks trained without oscillation ($n = 94$) showed bond differences ranging from 40 to 70 bonds.

monomers through an attractive Morse potential with a tunable affinity. The affinities have a maximal value of 6 kT and a minimal value of 0 kT. As such, the design space has dimension 66, equal to the number of lower-triangular entries in a 13 × 13 affinity matrix without including the first off-diagonal; see *SI Appendix* for further simulation details.

Our task is to select affinity matrices which can readily switch between two distinct, mutually exclusive design goals (Fig. 4*A*). Our first goal, $G1$, is to produce polymers that fold from a fully extended initial condition into a spiral with a winding number exceeding that of 1.0 around the monomer at the head of the polymer. As self-assembly is stochastic, we further specify that this must occur with greater than 70% probability within 500 units of simulation time. $G2$ is defined analogously, with the difference being that the target structure is now an antispiral, which has a winding number exceeding that of 1.0 measured relative to the monomer at the tail of the polymer. We use the same notation as in previous sections—$G1$ and $G2$—to refer to these goals.

We optimize the yield of a given target structure over the 66 design parameters using the covariance matrix adaptation evolutionary strategy (CMA-ES) (44, 45) that simulates an evolving population of design parameters. The loss function for our implementation of CMA-ES is the negative of the yield, with a floor set by the minimum 70% yield required for successfully achieving either $G1$ or $G2$; see Supplemental Information for further parameter choice details.

We perform optimization with two training protocols: 1. "no-oscillation" training, where $G1$ and $G2$ are optimized individually, and 2. "with-oscillation" training, where we switch between $G1$ and $G2$ with a period of 5 training steps. When we successfully perform with-oscillation, we see that the maximal yield of each goal increases in an alternating fashion with each successive training period (Fig. 4*B*). This suggests that oscillating training is converging to affinity matrix solutions which can easily switch between $G1$ and $G2$. Through this procedure, we collect a set of affinity matrix pairs A_1, A_2. Additionally, we verify that the adaptability of our pairs A_1, A_2 does not come at the expense of significant performance degradation on the individual goals they are trained for (*SI Appendix*, Fig. 2). We collect an analogous set of pairs for no-oscillation training simply by running converged optimizations from the same initial conditions.

To characterize the distribution of successful affinity matrix pairs, we computed the average difference between a matrix that achieved $G1$ and its corresponding $G2$ partner (Fig. 4 *C*, *Top row*), focusing only on entries that changed substantially (i.e., by more than 1 kT). The resulting average difference matrix from the no-oscillation training shows many more changed entries than the matrices from with-oscillation training. This visually suggests that oscillatory training identifies more adaptable regions of design space, where the affinity matrices which achieve $G1$ are closer to those which achieve $G2$. The average affinity matrix for each goal supports the same conclusion (Fig. 4 *C*, *Bottom row*). Note that we have shifted the monomer numbering when plotting affinity matrices, for ease of visual comparison.

To quantitatively confirm these visual conclusions, we compute the fraction of matrix entries that change by more than 2 kT between each A_1, A_2 affinity matrix pair (Fig. 4*D*). As expected, the distribution of this metric across all such pairs is substantially higher for no-oscillation training than for with-oscillation training.

Physical interpretation. The adaptability of self-assembly found here is surprising at first glance. The average adaptable pair of affinity matrices A_1, A_2 resemble each other for the majority of elements (Fig. 4 *C*, *Right*), yet fold into incompatible configurations with high yield.

To understand the physical design principles underlying such adaptability, we estimated aspects of the folding energy landscape for affinity matrices A found through oscillatory and nonoscillatory training; we computed the energies of folded configurations with different winding number (*SI Appendix*). We find that on-target structures are similarly stabilized by both oscillatory and nonoscillatory training, as suggested by the cartoon in Fig. 5*A*.

However, the two training protocols differ in how they treat off-target structures. With nonoscillatory training, the off-target structures are relatively high in energy since training is only ever shown the on-target structure (Fig. 5*B*). Consequently, the affinity matrix requires extensive changes to assemble the off-target structure. With oscillatory training, both on-target and off-target states are low-energy states. Relative to the no-oscillation off-target distribution, the with-oscillation off-target distribution is lower by ∼ 10kT (Fig. 5*B*).

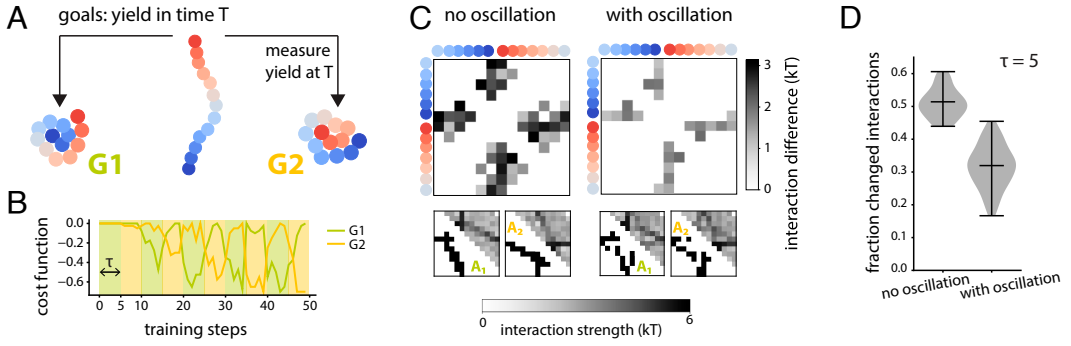

Fig. 4. Oscillatory training protocols generate adaptable solutions for heteropolymer folding. (*A*) We train the monomer interaction matrix for a heteropolymer of length 13 in order to target different folded structures—a clockwise spiral $G1$ and a counterclockwise spiral $G2$—in finite time, starting from an unfolded state. Monomers are colored according to position. (*B*) Performance on each goal $G1$, $G2$ is quantified by a cost function. Lower cost function indicates improved performance. Training drives cost function of on-target goal lower during each period. Background color panels indicate on-target goal. τ indicates training steps per period of goal oscillation. (*C*) Trained interaction matrices that target a spiral $G1$ and an antispiral $G2$: (*Bottom*) matrices A_1, A_2; the upper triangle is the matrix, and the lower triangle shows upper quartile interaction values. (*Top*) matrix difference $|A_1 - A_2|$. Both top and bottom panels are averaged over no-oscillation ($n = 40$) and with-oscillation ($n = 62$) training runs. Any matrix elements < 1 kT are visualized in white. Polymer ends are positioned at the center of interaction matrices. (*D*) Fraction of interactions which change by > 2 kT to switch between $G1$, $G2$. Violin plots show distribution over no-oscillation training pairs ($n = 40$) and with-oscillation pairs ($n = 62$). Lines indicate minimum, mean, and maximum values.

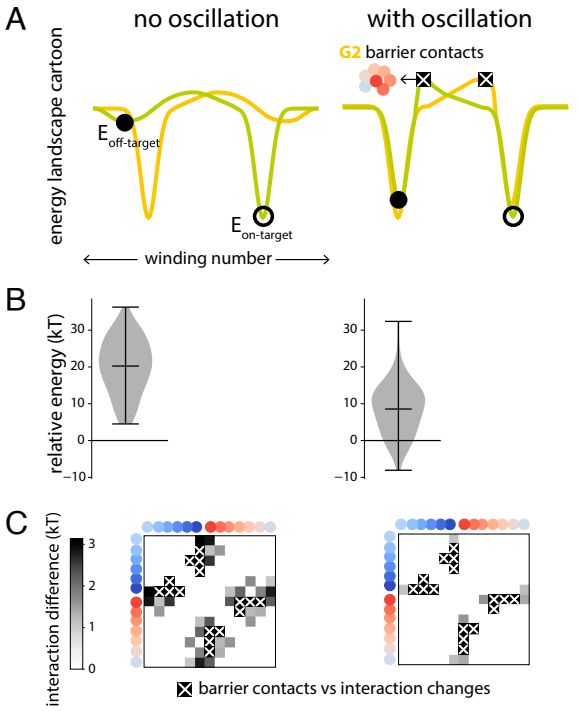

Fig. 5. Adaptability in polymer folding relies on localizing interaction changes to nucleation barriers. (*A*) When trained without oscillations, energy landscapes (cartoons) targeting *G*1 (green) and *G*2 (orange) have deep energy minima at their on-target *G*1 and *G*2 (black open circle) but high energy at off-target *G*2 and *G*1 (black solid circle), respectively. With oscillatory training, discrimination is now kinetic; both on- and off-target energies remain low, but the landscape develops kinetic barriers to folding (white cross black squares). Kinetic barriers enable adaptability since folding can changed by changing a small number of contacts in the nucleation seed. (*B*) Off-target energy distributions for no-oscillation (*Left*) and with-oscillation (*Right*) training (data from 40 and 62 simulation runs, respectively); energy is relative to the mean of the no-oscillation on-target distribution. The off-target distribution is lower in energy than in with-oscillation training. Violin plot lines indicate minima, means, and maxima. (*C*) As in Fig. 4*C*, interaction matrix differences between *G*1 and *G*2, for no-oscillation (*Left*) and with-oscillation (*Right*) training. Barrier contacts are overlaid as white cross black squares; oscillatory training localizes interaction matrix changes to the barrier contacts. Monomers are colored according to position.

Despite such energetic stabilization of off-target structures, oscillatory training results in robust on-target folding by exploiting kinetics. Folding is controlled by a nucleation barrier that is higher for the off-target structure than for the on-target structure (*SI Appendix*, Fig. 3). Using the estimated energy landscapes, we can identify "barrier contacts" that need to form in a partially folded nucleation seed before subsequent downhill folding to completion. Oscillatory training localizes the few changed affinities to those involved in forming barrier contacts (black in Fig. 5*C*).

Thus, our time-varying algorithm points at a physical principle for adaptive self-assembly of independent validity. Kinetic yield is controlled by partially folded early intermediate structures that correspond to nucleation barriers. These barriers can be lowered in energy or conversely destabilized by relatively few changes to the affinity matrix, resulting in the spiral or antispiral with high selectivity. Similar principles might apply more broadly to proteins and ribozymes where partly folded configurations, en route to fully folded configurations can be destabilized; indeed, such mechanisms might operate in experimentally characterized adaptable proteins and ribozymes where a single mutation can switch the polymer between distinct structures and thus function (46, 47).

General Trade-Offs and Design Principles. Across the diversity of optimization methods and material physics, we observed some common features to the emergence of adaptability.

First, there is a trade-off between the yield of the training and the adaptability of the converged solutions. In each context, both the yield and the adaptability are controlled by the frequency of oscillation. Increasing oscillation frequency increases the adaptability of converged solution pairs. At the same time, increasing oscillation frequency decreases the yield of the procedure (Fig. 6). This trade-off between yield and adaptability is fundamental to our proposed scheme; we produce adaptable pairs by selecting for those solutions which have survived a transition between *G*1 and *G*2 in one period of training. As the training period shortens, this selection becomes a more stringent requirement, and no such adaptable solutions might exist in the region of parameter space being explored by the random initialization. Thus, for any given problem, the selection-based nature of the optimization means that increasing yield requires giving up adaptability, and increasing adaptability might mean sacrificing yield.

Additionally, irreversible training rules lower the overall scale of the yield and adaptability trade-off. For instance, 98% of the failures in the Poisson's ratio training for single targets were due to mechanical failure, with an overall yield of 20%. These mechanical failures were a natural consequence of irreversible bond removal training; bonds can never be replaced and thus networks eventually form system-spanning cracks. By contrast, yields in systems where we allowed reversible strengthening and weakening of interactions had higher yields (Fig. 6 *A* and *C*).

Second, the dimensionality of the design space influences multiple aspects of the oscillatory training process. A system with more tunable disorder (and hence a higher-dimensional design space) will host more degenerate solutions for a given goal; increased dimensionality is likely to increase the probability of single-target training success. By the same logic, we might expect that systems with more design parameters are more amenable to producing adaptable solutions. To test this intuition, we performed training of smaller elastic networks, with fewer design parameters. We found that the yield in these smaller systems was lower than those at the comparable oscillation frequency in

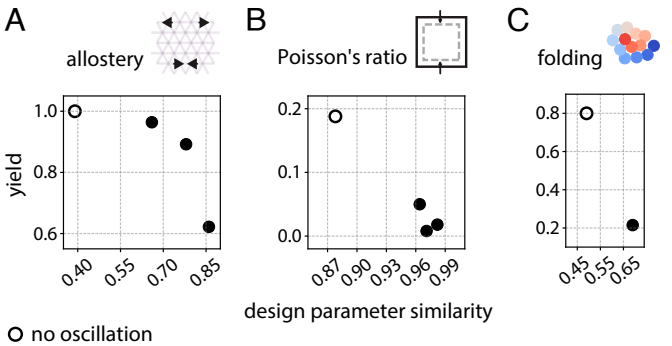

Fig. 6. A trade-off between adaptability and yield. Fraction of optimization trajectories which successfully converge to solutions when trained for incompatible functions (yield). Yield is plotted against design parameter similarity, the average fraction of parameters which changed when switching between incompatible functions. Training for adaptable (*A*) allosteric modes and (*B*) Poisson's ratios in elastic networks and (*C*) folded structure of heteropolymers. Total number of training runs for each task and frequency condition: allostery (all n = 500), Poisson's ratio (all n = 500), and folding (no oscillation n = 50, with oscillation n=200). τ is the length of each training episode, as shown in Figs. 2*C*, 3*C*, and 4*B*. Training in (*A*) and (*C*) allows for bidirectional changes in parameters during training. In (*B*), training only allows for weakening or removal of bonds, leading to mechanical failure more often, and lower yields.

the larger systems (*SI Appendix*, Fig. 4). For allostery, we also observed that the fraction of changed bonds increased relative to the larger systems (*SI Appendix*, Fig. 4*B*).

Finally, we observe that adaptable solutions are potentially less robust to perturbation. In particular, we investigated the effects of pruning bonds in elastic networks trained to have Poisson's ratio $\nu > 0.75$. We found that single bond removals rarely had a substantial effect on control networks trained with nonoscillatory protocols. While the majority of bonds removed had a similarly small effect in adaptably trained networks, a small number of bond removals had an impact on ν greater than the range found in the controls (*SI Appendix*, Fig. 5*A*). Similarly, we found that adaptably trained networks had on average $1-2$ bonds which could destroy the $\nu > 0.75$ functionality, while control networks had none (*SI Appendix*, Fig. 5*B*). The increased sensitivity of adaptably trained networks is consistent with an intuitive picture of our optimization scheme, where oscillatory training localizes systems to the boundaries of high-dimensional degenerate functionality sets. Nevertheless, 99% of bond removal perturbations do not affect the ability of the system to perform its function.

Discussion

Biological materials differ from synthetic materials in not just their physics and composition but also in the history of their development. In fact, the way biological materials are arrived at, through a process of incremental evolution in a sequence of historic environments (48, 49), is critical in understanding why they function differently from synthetic materials. While synthetic materials can sometimes rival or even exceed specific functionalities of natural systems, these synthetic systems are lagging precisely in metaproperties like adaptability, robustness, and ability to acquire unique functions on the fly.

Here, we have shown how one such property, adaptability, can arise without any direct optimization. Instead, we find adaptable materials by applying a time-varying sequence of selection pressures during design.

This adaptability comes through the spontaneous formation of identifiable physical units, despite the fact that our goals and systems were not explicitly modular in form. In contrast to prior work (1, 2) which used explicitly modular goals, the physical systems in our work must identify implicit "modules," e.g., reusable nucleation-based pathways or mechanical substructures, that can be used for the multiple nonmodular functions we require. In this way, our work can be seen as building on those ideas to identify physical principles specific to the physics of systems studied (e.g., nucleation for polymer folding) that allow for adaptability.

Our proposed method has wide applicability, in that it functions as a wrapper around preexisting design programs, and can be applied without in-depth knowledge of a system.

In fact, the framework outlined here can be extended to greater than two functionalities. When such multifunctional training converges, it will by construction identify adaptable solutions which can easily toggle between functions in sequence. However, the yield curve as a function of oscillation frequency is likely to decrease rapidly.

As an optimization method for high-dimensional cost functions, there are also potential connections to time-varying training frameworks commonly employed in machine learning. For instance, the study of catastrophic forgetting (50) has yielded methods for the robust sequential training of multiple tasks. Similarly, curriculum learning (51) and dynamical loss functions (52) have been used to improve generalization. Our work suggests measuring whether time-varying machine learning frameworks update weights parsimoniously when successfully retraining for different tasks.

Our method can also help reveal system-specific physical insights that can be exploited without further need for our method. For example, in many of the current platforms for self-assembly, the yield is frequently governed by kinetics rather than equilibrium free energies (53–55). Our simulations of heteropolymer self-assembly revealed a broadly relevant design principle for such systems—nucleation barriers in energy landscapes can be leveraged to create adaptability in self-assembly.

Similarly, in the context of elastic networks, we identified that coherent motions which link two allosteric sites can be easily diverted in order to achieve incompatible goals. These insights can now be used to guide design without need for blind numeric optimization, both in synthetic systems like colloids (54) and DNA nanotechnology (56), but also in natural systems like proteins (55).

One key condition for our method's success is the existence "neutral variation" in the goals under consideration (47, 57, 58); there must exist changes in design parameters that have no cost in terms of the current target functionality but that help adapt to new functionality. The systems studied here have such degeneracy; for example, many elastic networks with different bond stiffnesses showed the same desired allosteric response. In fact, degeneracy is generically expected whenever systems are disordered, with the number of design parameters often being extensive in the size of the system. In biological examples, such genotype-to-phenotype maps are often redundant when the space of genotypes (or design parameters) is larger dimensional than the space of phenotypes (exhibited properties). We do not expect as much success in systems such as self-assembled crystals with only 1 or 2 species or regular lattices of elastic elements.

Materials and Methods

We briefly review the methods employed for each of the three tasks considered in the main text. For a more detailed description, please refer to the appropriate section in *SI Appendix*.

Training for Mechanical Allostery with Gradient Descent. To train for mechanical allostery, we compute the dynamical matrix of a 22-node hexagonal lattice with fixed boundary nodes and an arbitrary set of bond stiffnesses. From the dynamical matrix, we can construct a cost function which encourages overlap between the lowest energy mode and the desired allosteric behavior, in addition to encouraging a gap between the lowest and second-lowest energy modes. To train for a given mechanical allostery behavior, we use automatic differentiation to perform gradient descent on the cost function with respect to the bond stiffnesses.

Training for Poisson Ratios with Local Bond Pruning Rules. To train for Poisson ratio ν, we compute the most strained bond when a deformation is applied to a elastic network with uniform stiffness and disordered geometry. The network is generated from jammed packings and consists of approximately 200 nodes. To train for positive ν, we apply a shear deformation. For negative ν, we apply a compressive deformation. In both cases, the most strained bond under deformation is removed. We use the rigidpy library (59) to apply the deformations.

Training for Heteropolymer Folding with CMA-ES. To train for heteropolymer folding path selection, we use a Langevin dynamics simulation implemented via HOOMD-blue (43). We simulate a 13-monomer heteropolymer held together

by harmonic springs. The polymer experiences thermal fluctuations, bending stiffness, excluded volume forces, and forces arising from the intermonomer Morse potentials which are set via an affinity matrix with 66 independent entries. The cost function value attached to a given affinity matrix is given by the fraction of simulation runs in which the polymer has exceeded a threshold winding number at a given simulation time. We use a covariance matrix adaptation evolution strategy (CMA-ES), implemented through the python package pycma (60), in order to search for optimal affinity matrices.

Data, Materials, and Software Availability. Code for each set of tasks has been deposited in Github (allostery (61): https://github.com/jiayiwus1x/build-soft-modes-of-networks; Poisson's ratio (62): https://github.com/AyannaMatthews/AdaptableTraining-Auxetics; heteropolymers (63): https://github.com/falkma/AdaptableTraining-Heteropolymers). Data to reproduce main figure results can be found on Zenodo (https://doi.org/10.5281/zenodo.8019474)(64). Any additional codes and datasets are available from the authors upon request.

ACKNOWLEDGMENTS. We thank Varda Hagh, Kabir Husain, Heinrich Jaeger, Ilya Nemenman, Rama Ranganathan, and Riccardo Ravasio for discussions. This work was primarily supported by the University of Chicago Materials Research Science and Engineering Center, which is funded by the NSF under award number DMR-2011854. M.L.G. acknowledges support from NSF Grant DMR-2215605. A. Matthews acknowledges support from the University of Chicago Biophysics Training Grant. A. Murugan acknowledges support from the Simons Foundation. S.R.N. acknowledges support from DOE Basic Energy Sciences Grant DE-SC0020972. This work was completed in part with resources provided by the University of Chicago's Research Computing Center.

Author affiliations: [a]Department of Physics, The University of Chicago, Chicago, IL 60637; [b]Graduate Program in Biophysical Sciences, The University of Chicago, Chicago, IL 60637; [c]School of Engineering and Applied Science, Yale University, New Haven, CT 06511; [d]James Franck Institute, The University of Chicago, Chicago, IL 60637; [e]Institute for Biophysical Dynamics, The University of Chicago, Chicago, IL 60637; and [f]Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637

1. N. Kashtan, U. Alon, Spontaneous evolution of modularity and network motifs. *Proc. Natl. Acad. Sci.* **102**, 13773–13778 (2005).
2. M. Hemery, O. Rivoire, Evolution of sparsity and modularity in a model of protein allostery. *Phys. Rev. E* **91**, 042704. (2015).
3. M. Parter, N. Kashtan, U. Alon, Facilitated variation: how evolution learns from past environments to generalize to new environments. *PLoS Comput. Biol.* **4**, e1000206. (2008).
4. N. Kashtan, E. Noor, U. Alon, Varying environments can speed up evolution. *Proc. Natl. Acad. Sci.* **104**, 13711–13716 (2007).
5. H. Lipson, J. B. Pollack, N. P. Suh, On the origin of modular variation. *Evolution* **56**, 1549–1556 (2002).
6. A. Wagner, Robustness and evolvability: A paradox resolved. *Proc. R. Soc. B: Biol. Sci.* **275**, 91–100 (2008).
7. A. Gupta, S. Savarese, S. Ganguli, L. Fei-Fei, Embodied intelligence via learning and evolution. *Nat. Commun.* **12**, 1–12 (2021).
8. C. X. Du, G. van Anders, J. Dshemuchadse, P. M. Dodd, S. C. Glotzer, Inverse design of compression-induced solid-solid transitions in colloids. *Mol. Simul.* **46**, 1037–1044 (2020).
9. J. W. Rocks, H. Ronellenfitsch, A. J. Liu, S. R. Nagel, E. Katifori, Limits of multifunctionality in tunable networks. *Proc. Natl. Acad. Sci.* **116**, 2506–2511 (2019).
10. V. Sachdeva, K. Husain, J. Sheng, S. Wang, A. Murugan, Tuning environmental timescales to evolve and maintain generalists. *Proc. Natl. Acad. Sci.* **117**, 12693–12699 (2020).
11. A. Murugan *et al.*, Roadmap on biology in time varying environments. *Phys. Biol.* **18** (2021).
12. M. Rechtsman, F. Stillinger, S. Torquato, Designed interaction potentials via inverse methods for self-assembly. *Phys. Rev. E* **73**, 011406. (2006).
13. M. C. Rechtsman, F. H. Stillinger, S. Torquato, Optimized interactions for targeted self-assembly: Application to a honeycomb lattice. *Phys. Rev. Lett.* **95**, 228301. (2005).
14. G. M. Coli, E. Boattini, L. Filion, M. Dijkstra, Inverse design of soft materials via a deep learning–based evolutionary strategy. *Sci. Adv.* **8**, eabj6731 (2022).
15. M. C. Rechtsman, F. H. Stillinger, S. Torquato, Self-assembly of the simple cubic lattice with an isotropic potential. *Phys. Rev. E* **74**, 021404. (2006).
16. S. Torquato, Inverse optimization techniques for targeted self-assembly. *Soft Matter* **5**, 1157–1173 (2009).
17. Z. Zeravcic, V. N. Manoharan, M. P. Brenner, Size limits of self-assembled colloidal structures made using specific interactions. *Proc. Natl. Acad. Sci.* **111**, 15918–15923 (2014).
18. A. Zunger, Inverse design in search of materials with target functionalities. *Nat. Rev. Chem.* **2**, 1–16 (2018).
19. A. Murugan, H. M. Jaeger, Bioinspired nonequilibrium search for novel materials. *MRS Bull.* **44**, 96–105 (2019).
20. H. Lipson, J. B. Pollack, Automatic design and manufacture of robotic lifeforms. *Nature* **406**, 974–978 (2000).
21. E. Bianchi, G. Doppelbauer, L. Filion, M. Dijkstra, G. Kahl, Predicting patchy particle crystals: variable box shape simulations and evolutionary algorithms. *J. Chem. Phys.* **136**, 214102 (2012).
22. M. Engel, P. F. Damasceno, C. L. Phillips, S. C. Glotzer, Computational self-assembly of a one-component icosahedral quasicrystal. *Nat. Mater.* **14**, 109–116 (2015).
23. Z. Zhang, A. S. Keys, T. Chen, S. C. Glotzer, Self-assembly of patchy particles into diamond structures through molecular mimicry. *Langmuir* **21**, 11547–11551 (2005).
24. S. F. Greenbury, A. A. Louis, S. E. Ahnert, *The structure of genotype-phenotype maps makes fitness landscapes navigable* (Nat. Ecol, Evol, 2022).
25. E. van Nimwegen, Influenza escapes immunity along neutral networks. *Science* **314**, 1884–1886 (2006).
26. M. Ebner, M. Shackleton, R. Shipman, How neutral networks influence evolvability. *Complexity* **7**, 19–33 (2001).
27. J. W. Rocks, A. J. Liu, E. Katifori, Revealing structure-function relationships in functional flow networks via persistent homology. *Phys. Rev. Res.* **2**, 033234. (2020).
28. J. W. Rocks, A. J. Liu, E. Katifori, Hidden topological structure of flow network functionality. *Phys. Rev. Lett.* **126**, 028102. (2021).
29. N. Pashine, D. Hexner, A. J. Liu, S. R. Nagel, Directed aging, memory, and nature's greed. *Sci. Adv.* **5**, eaax4215 (2019).
30. N. Pashine, Local rules for fabricating allosteric networks. *Phys. Rev. Mater.* **5**, 065607. (2021).
31. J. W. Rocks *et al.*, Designing allostery-inspired response in mechanical networks. *Proc. Natl. Acad. Sci.* **114**, 2520–2525 (2017).
32. L. Yan, R. Ravasio, C. Brito, M. Wyart, Architecture and coevolution of allosteric materials. *Proc. Natl. Acad. Sci.* **114**, 2526–2531 (2017).
33. L. Yan, R. Ravasio, C. Brito, M. Wyart, Principles for optimal cooperativity in allosteric materials. *Biophys. J.* **114**, 2787–2798 (2018).
34. T. Tlusty, A. Libchaber, J. P. Eckmann, Physical model of the genotype-to-phenotype map of proteins. *Phys. Rev. X* **7**, 021037. (2017).
35. H. Flechsig, Design of elastic networks with evolutionary optimized long-range communication as mechanical models of allosteric proteins. *Biophys. J.* **113**, 558–571 (2017).
36. M. Stern, M. B. Pinson, A. Murugan, Continual learning of multiple memories in mechanical networks. *Phys. Rev. X* **10**, 031044. (2020).
37. M. Stern, C. Arinze, L. Perez, S. E. Palmer, A. Murugan, Supervised learning through physical changes in a mechanical system. *Proc. Natl. Acad. Sci.* **117**, 14843–14850 (2020).
38. M. Stern, D. Hexner, J. W. Rocks, A. J. Liu, Supervised learning in physical networks: From machine learning to learning machines. *Phys. Rev. X* **11**, 021045. (2021).
39. C. P. Goodrich, A. J. Liu, S. R. Nagel, The principle of independent bond-level response: Tuning by pruning to exploit disorder for global behavior. *Phys. Rev. Lett.* **114**, 225501. (2015).
40. D. R. Reid *et al.*, Auxetic metamaterials from disordered networks. *Proc. Natl. Acad. Sci.* **115**, E1384–E1390 (2018).
41. N. Go, Theoretical studies of protein folding. *Annu. Rev. Biophys. Bioeng.* **12**, 183–210 (1983).
42. V. S. Pande, A. Y. Grosberg, T. Tanaka, Heteropolymer freezing and design: towards physical models of protein folding. *Rev. Mod. Phys.* **72**, 259 (2000).
43. J. A. Anderson, J. Glaser, S. C. Glotzer, Hoomd-blue: A python package for high-performance molecular dynamics and hard particle monte carlo simulations. *Comput. Mater. Sci.* **173**, 109363. (2020).
44. N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation in *Proceedings of IEEE international conference on evolutionary computation*. (IEEE), pp. 312–317 (1996).
45. N. Hansen, The cma evolution strategy: A comparing review. *Towards New Evol. Comput.* pp. 75–102 (2006).
46. A. S. Raman, K. I. White, R. Ranganathan, Origins of allostery and evolvability in proteins: A case study. *Cell* **166**, 468–480 (2016).
47. E. A. Schultes, D. P. Bartel, One sequence, two ribozymes: Implications for the emergence of new ribozyme folds. *Science* **289**, 448–452 (2000).
48. G. P. Wagner, L. Altenberg, PERSPECTIVE: COMPLEX ADAPTATIONS AND THE EVOLUTION OF EVOLVABILITY. *Evolution* **50**, 967–976 (1996).
49. J. M. Smith, Natural selection and the concept of a protein space. *Nature* **225**, 563–564 (1970).
50. J. Kirkpatrick *et al.*, Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**, 3521–3526 (2017).
51. Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning in *Proceedings of the 26th annual international conference on machine learning*. pp. 41–48 (2009).
52. M. Ruiz-Garcia, G. Zhang, S. S. Schoenholz, A. J. Liu, Tilting the playing field: Dynamical loss functions for machine learning in *International Conference on Machine Learning*. (PMLR), pp. 9157–9167 (2021).
53. W. M. Jacobs, A. Reinhardt, D. Frenkel, Rational design of self-assembly pathways for complex multicomponent structures. *Proc. Natl. Acad. Sci.* **112**, 6313–6318 (2015).
54. A. Hensley, W. M. Jacobs, W. B. Rogers, Self-assembly of photonic crystals by controlling the nucleation and growth of dna-coated colloids. *Proc. Natl. Acad. Sci.* **119** (2022).
55. A. Bitran, W. M. Jacobs, X. Zhai, E. Shakhnovich, Cotranslational folding allows misfolding-prone proteins to circumvent deep kinetic traps. *Proc. Natl. Acad. Sci.* **117**, 1485–1495 (2020).
56. K. E. Dunn *et al.*, Guiding the folding pathway of DNA origami. *Nature* **525**, 82–86 (2015).
57. W. Gruner *et al.*, Analysis of RNA sequence structure maps by exhaustive enumeration. II. structures of neutral networks and shape space covering. *Monatsh. Chem.* **127**, 375–389 (1996).
58. W. Grüner *et al.*, Analysis of RNA sequence structure maps by exhaustive enumeration i. neutral networks. *Monatsh. Chem.* **127**, 355–374 (1996).
59. V. F. Hagh, M. Sadjadi, rigidpy: Rigidity analysis in python. *Comput. Phys. Commun.* **275**, 108306. (2022).
60. N. Hansen, Y. Akimoto, P. Baudis, Cma-es/pycma: r3. 0.3 (2020).

61. J. Wu, build-soft-modes-of-networks. Github. https://github.com/jiayiwus1x/build-soft-modes-of-networks. Deposited 18 October 2019.
62. A. Matthews, AdaptableTraining-Auxetics. Github. https://github.com/AyannaMatthews/AdaptableTraining-Auxetics. Deposited 7 June 2023.
63. M. J. Falk, AdaptableTraining-Heteropolymers. Github. https://github.com/falkma/AdaptableTraining-Heteropolymers. Deposited 9 June 2023.
64. M. J. Falk, J. Wu, A. Matthews, AdaptableTraining-DataAndCode. Zenodo. https://doi.org/10.5281/zenodo.8019474. Deposited 9 June 2023.