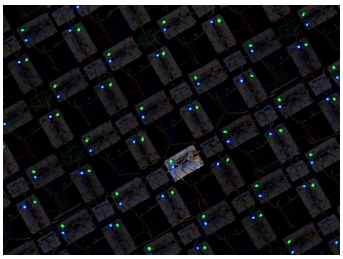
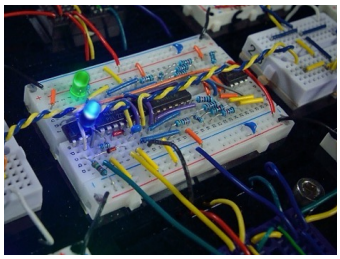
Autonomous Learning Metamaterials

Douglas J. Durian <djdurian@physics.upenn.edu>
University of Pennsylvania, Philadelphia PA

- Physical systems made of many copies of a repeat unit, each of which uses a *local learning rule* to collectively optimize a *global* cost function defining a task to be "computed" physically. Eg "*Contrastive Local Learning Networks*"

  Analog in-memory training for analog in-memory analog computing for control, metamaterials with complex functionality, AI,…

- Boulder School 2024
  - Lecture 1: learning systems and rules
  - Lecture 2: electronic realizations
  - Lecture 3: mechanical realizations

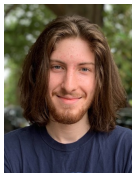



1

ACKNOWLEDGEMENTS

experiment


Sam Dillavou


Jacob Wycoff


Benjamin Beyer


Lauren Altman

theory


Menachem Stern


Andrea Liu

2nd generation


Marc Miskin

Dinesh Jayaraman, Maggie Miller, Shivangi Misra, Tarunyaa Sivakumar, Cynthia Sung

University of Pennsylvania

LRSM

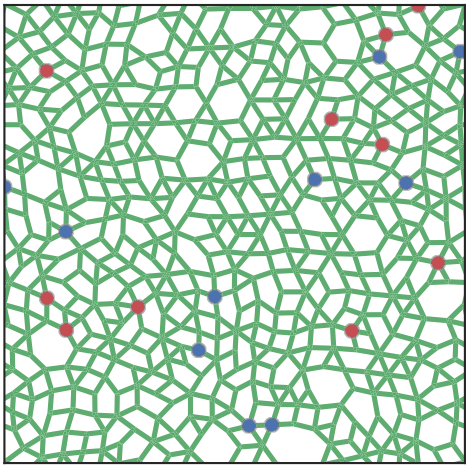
NSF UPenn MRSEC

2

Contrastive Local Learning Networks

- Can we build edges where all the edges independently self-adjust using the local learning rules so that **outputs** evolve to the desired function of **inputs**?

- **inputs**
- **outputs**



Example tasks/functions:
(1) Outputs are at $\{O_j\}$ when inputs are at $\{I_i\}$
(2) Outputs are a desired linear combo of inputs
(3) Classification: e.g inputs are grayscale pixel data and outputs indicate whether the image is a dog or a cat

Bottom-up learning of complex functionality using local rules in a fully recurrent network, giving brain-like advantages over ANNs

3

Top-down mechanical metamaterials

- Tune the repeat elements using global knowledge to learn complex functionality
  - *external memory and computation are inherently required*

Eg. two-motion-task with triangular spring network, as metaphor for airplane wing:

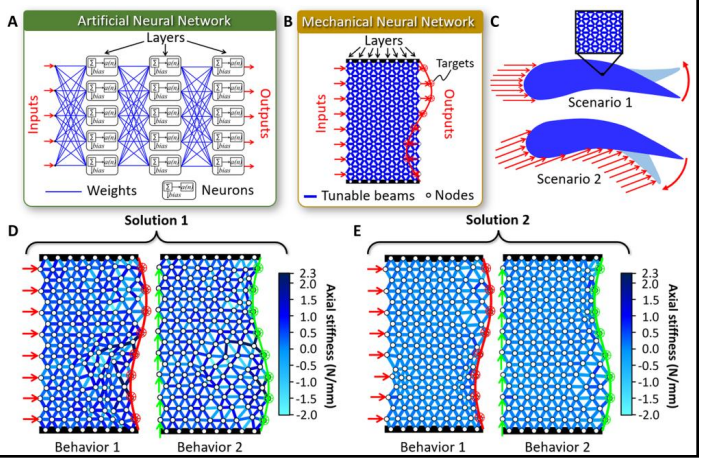
SCIENCE ROBOTICS | RESEARCH ARTICLE

Sci. Robot. **7**, eabq7278 (2022) 19 October 2022

ARTIFICIAL INTELLIGENCE

Mechanical neural networks: Architected materials that learn behaviors

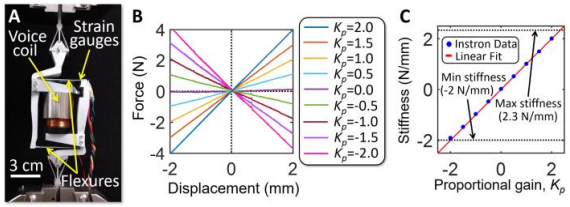
Ryan H. Lee[1†], Erwin A. B. Mulder[2], Jonathan B. Hopkins[1*†]



4

Top-down "airplane wing" example

- Small-scale realization via spring stiffnesses (positive & negative, using control)
  - 21 edges; 2 input nodes & two output nodes



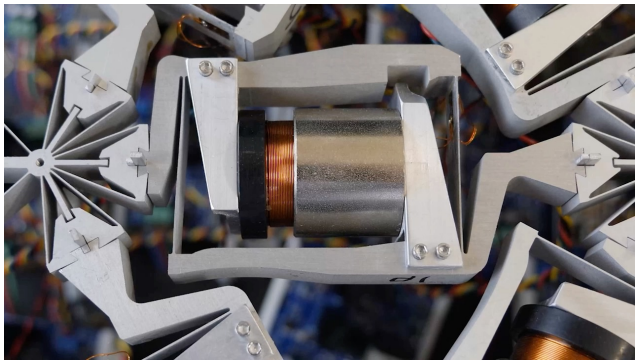


Fig. 2. Tunable beams that use closed-loop control to achieve variable axial stiffnesses. (**A**) Voice coil and strain gauges were used as actuators and sensors to control the axial stiffness of the beam. (**B**) Data collected from an Instron testing machine as it stretches and compresses the tunable beam while it is actively controlled to achieve linear force-displacement responses using different proportional gain values, K_p. (**C**) Plot demonstrating how well the controller's prescribed proportional gain corresponds with the beam's resulting axial stiffness.

5

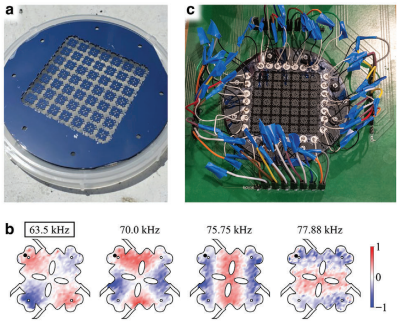
Top-down speech recognition example

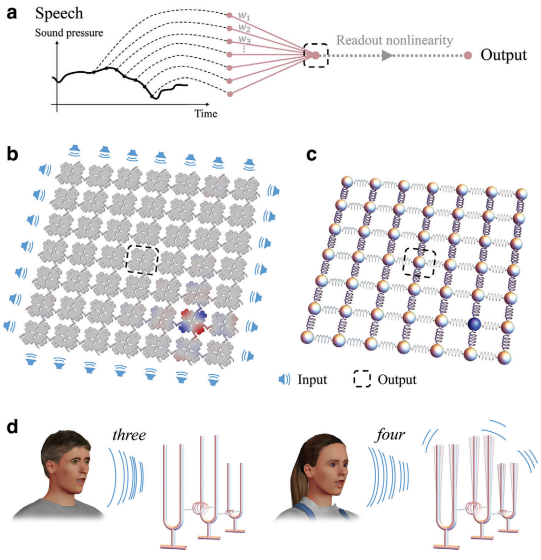
- With a trained mechanical metamaterial

  "springtronics"

In-Sensor Passive Speech Classification with Phononic Metamaterials *Adv. Funct. Mater.* **2024**, *34*, 2311877

Tena Dubček, Daniel Moreno-Garcia, Thomas Haag, Parisa Omidvar, Henrik R. Thomsen, Theodor S. Becker, Lars Gebraad, Christoph Bärlocher, Fredrik Andersson, Sebastian D. Huber, Dirk-Jan van Manen, Luis Guillermo Villanueva, Johan O.A. Robertsson, and Marc Serra-Garcia**





6

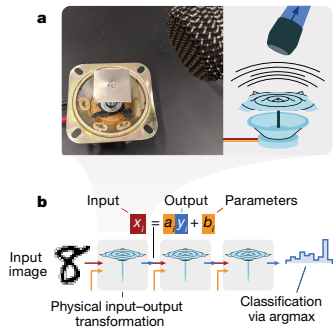
Top-down MNIST examples

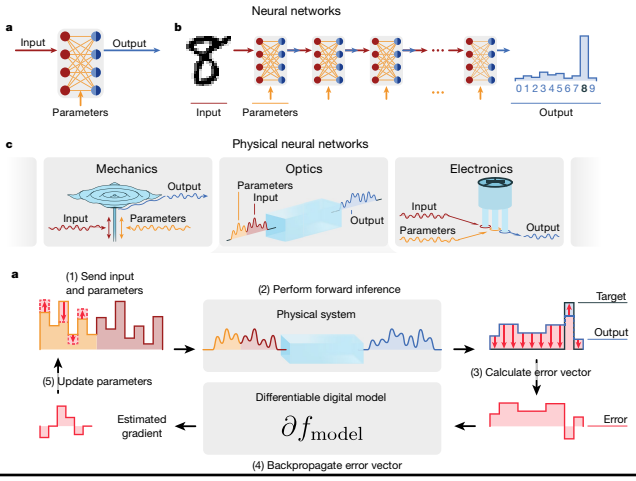
- Different physical "networks" (nonlin oscillators with info encoded in waves)

Deep physical neural networks trained with backpropagation

Logan G. Wright, Tatsuhiro Onodera, Martin M. Stein, Tianyu Wang, Darren T. Schachter, Zoey Hu & Peter L. McMahon

Nature **601**, 549–555 (2022)





7

Bottom-up mechanical metamaterials

- Tune the repeat elements using local learning rules, e.g coupled learning
  - eliminate inherent need for external memory or CPU…
  - eliminate "reality gap" between physical system and in-silico model used for training…

8

Coupled Learning for Spring Networks

- Elastic energy of a spring is $u = \frac{1}{2}\, k(x\text{-}L)^2$ → $\{k,L\}$ are learning parameters

- Contrast function $U^{Clamped} - U^{Free}$ is physically positive

  Drive it to zero by gradient descent on either spring constants or rest lengths

  For motion tasks:

$$\dot{k}_j = -\frac{\partial}{\partial k_j}\sum \frac{1}{2}k_i\left(x_i^C - L_i\right)^2 - \frac{1}{2}k_i\left(x_i^F - L_i\right)^2 = \left[L_j - (x_j^C + x_j^F)/2\right]\left(x_j^C - x_j^F\right)$$

$$\dot{L}_j = -\frac{\partial}{\partial L_j}\sum \frac{1}{2}k_i\left(x_i^C - L_i\right)^2 - \frac{1}{2}k_i\left(x_i^F - L_i\right)^2 = k_j\left(x_j^C - x_j^F\right)$$

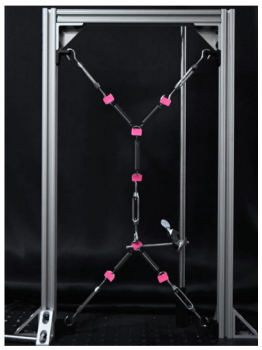
9

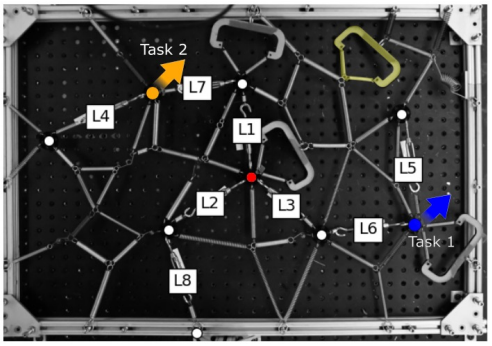
Mechanical Learning Metamaterials

- First attempt – motion tasks

  edge = adjustable turnbuckle in series with spring (under tension)

  hand-adjust edges using local rules:





Lauren Altman

[Altman, Stern, Liu, Durian, arXiv:2311.00170]

10

Coupled learning rule for turnbuckles

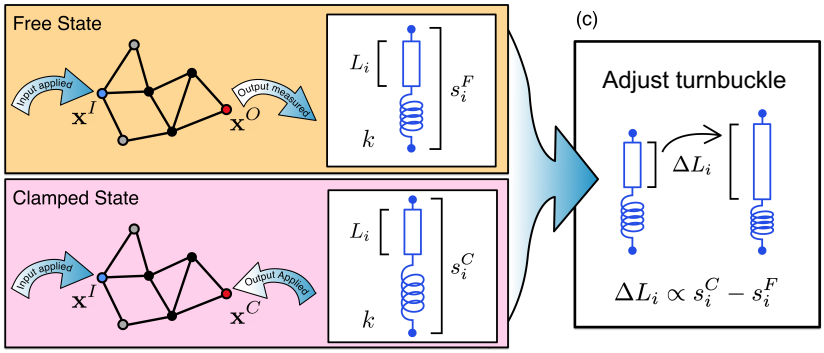
- Edge energy is set by [(node-node distance, s) – (rest + turnbuckle lengths)]^2

$$u_i = \frac{1}{2}k\left[s_i - (l + L_i)\right]^2$$

- Do gradient descent on (clamped energy – free energy) contrast function:

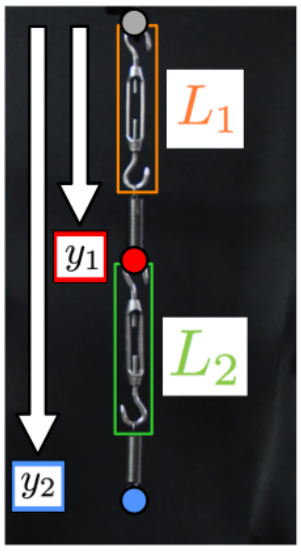
$$\begin{aligned}\frac{dL_i}{dt} &\propto -\frac{\partial}{\partial L_i}\left[\sum u_j^C - u_j^F\right] \\ &= -\frac{\partial}{\partial L_i}(u_i^C - u_i^F) \\ &= k(s_i^C - s_i^F)\end{aligned}$$

$\Delta L_i = \alpha(s_i^C - s_i^F)$ is a ***local*** update rule!



Example-1: Motion divider

- Identical springs in series: move free end (y_2) and learn a desired motion of middle node (y_1) by adjusting turnbuckle lengths {L_1, L_2}



- Range of motion tasks that can be learned:

Force balance: $k(y_1 - l - L_1) = k(y_2 - y_1 - l - L_2)$

Solution: $y_1 = \frac{1}{2}y_2 + a$ where $a = \frac{1}{2}(L_1 - L_2)$

- Overparameterized: {L_1, L_2} can evolve to a multitude of of final values giving the desired value of a

Different "a" tasks and learning rates

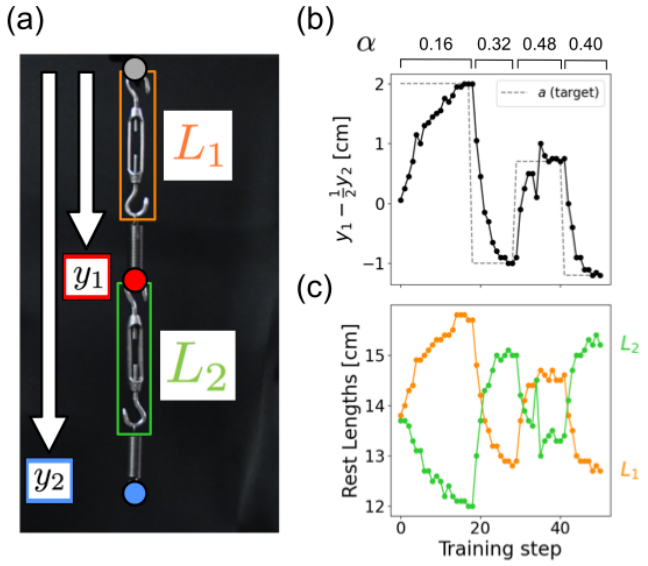
- Task

  $y_1 = \frac{1}{2} y_2 + a$ where $a = \frac{1}{2}(L_1 - L_2)$

- Update rule

  $\Delta L_i = \alpha(s_i^C - s_i^F)$

  – Converges to desired task in all four cases!



Learning Dynamics of $\Delta L_i = \alpha(s_i^C - s_i^F)$?

- Clamping $y_1^C = \eta y_1^D + (1-\eta) y_1^F$
- Gives $\Delta L_1 = \alpha\eta\left(a - \frac{L_1 - L_2}{2}\right)$

  $\Delta L_2 = -\Delta L_1$

  - Thus, we take full η=1 nudge in experiments

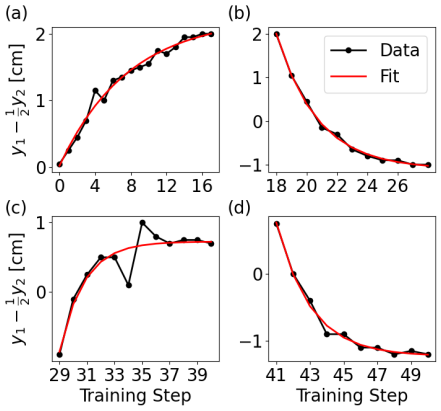
- Turn this into ODEs and solve:

$$L_1(t) = e^{-\nu t}\left(\frac{L_1^0}{2} - \frac{L_2^0}{2} - a\right) + a + \frac{L_1^0}{2} + \frac{L_2^0}{2}$$

$$L_2(t) = e^{-\nu t}\left(a - \frac{L_1^0}{2} + \frac{L_2^0}{2}\right) - a + \frac{L_1^0}{2} + \frac{L_2^0}{2}$$

where νt = α x number of training steps

→ exponential convergence, matches data



Trial	True a [cm]	Fit a [cm]	True α	Fit α
(a)	2.00	2.25	0.16	0.13
(b)	-1.00	-1.11	0.32	0.37
(c)	0.70	0.73	0.48	0.56
(d)	-1.20	-1.23	0.40	0.49

Example-2: Symmetrization

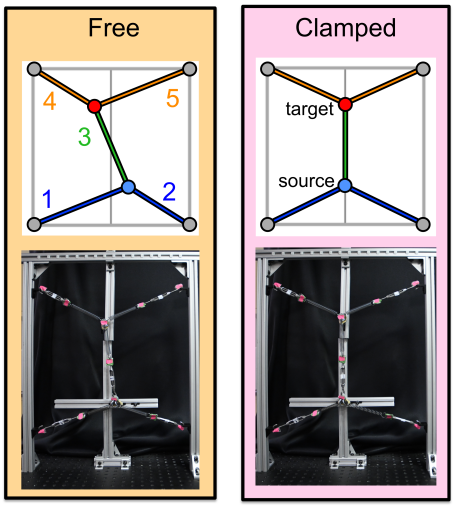
- Train to become left-right symmetrical by adjusting the five turnbuckle lengths.

  Two-input / one output: Tug source to center line at different heights and ask target to be on centerline

  $x_{\text{target}} = x_{\text{source}} = 0$ for *any* y_{source} value.

  Multitude of solutions that could be learned:

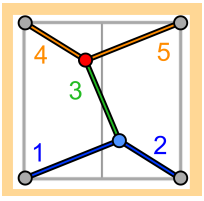
  $\{L_1=L_2$ & $L_4=L_5\}$; note that L_3 plays no role



15

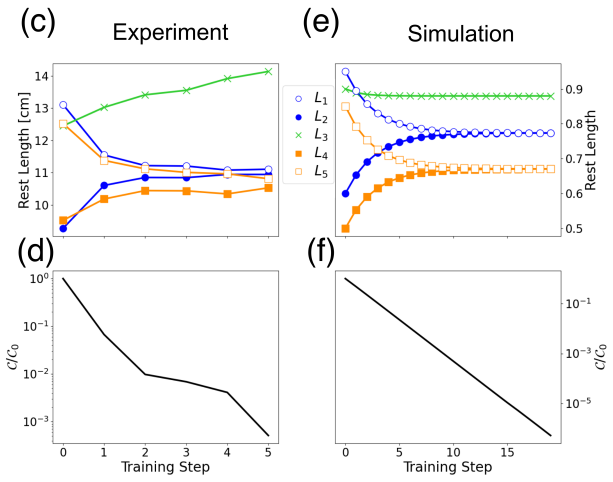
Learning dynamics

- Achieves $\{L_1=L_2$ & $L_4=L_5\}$ quickly while contrast $C=U_{clamped}-U_{free}$ decreases exponentially



- Note that L_3 becomes constant in simulation, but drifts up in experiment

  {toward lower total energy}

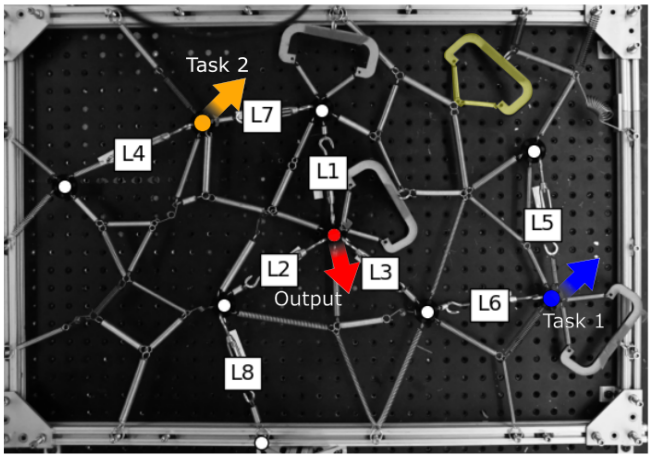


$\eta = 1,\ \alpha = 1$ $\eta = 1,\ \alpha = 0.4$

16

Example-3: random nonlinear network

- 39 nodes and 59 edges, 8 of which have adjustable turnbuckles
  - A variety of stiffnesses and rest lengths; some are nonlinear; some are torsion; some go slack and buckle (to avoid compression). Would be quite hard to simulate, and maybe not possible to train *in silico*
- Train for *two* simultaneous tasks

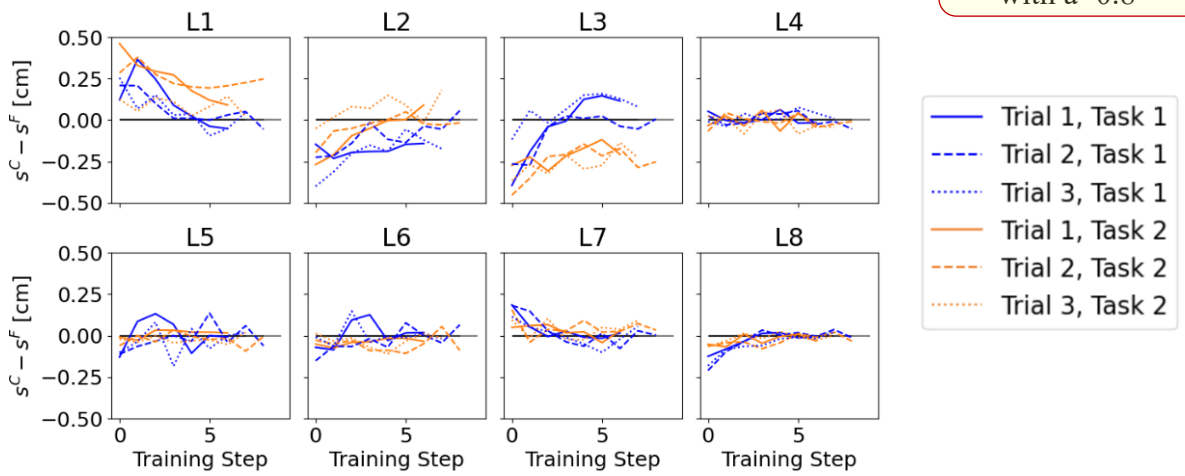


17

Learning dynamics

- Trials starting from three different initial conditions:

$$\Delta L_i = \alpha(s_i^C - s_i^F)$$
with a=0.8



18

What about…

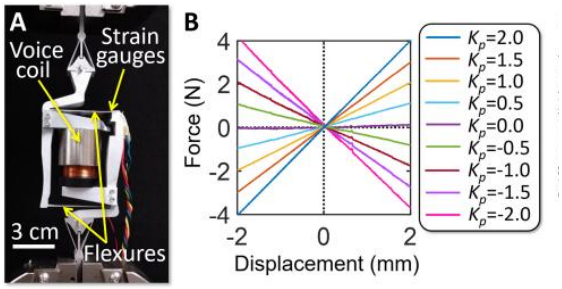
- Autonomous updates?
  - Need *in situ* mechanisms to sense change in edge length in free vs clamped conditions and to adjust learning DOF accordingly
- Force tasks?
  - Need ability to tune the spring stiffness, since $u = F^2/(2k)$ is independent of rest length

19

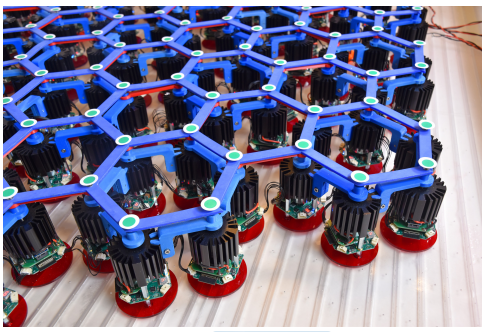
Variable-stiffness springs

- Perhaps-surprisingly hard to make, and harder to miniaturize!

  two examples, both under active control:



Sci. Robot. **7**, eabq7278 (2022) 19 October 2022



arXiv:2108.08837 [pdf, other] cond-mat.soft

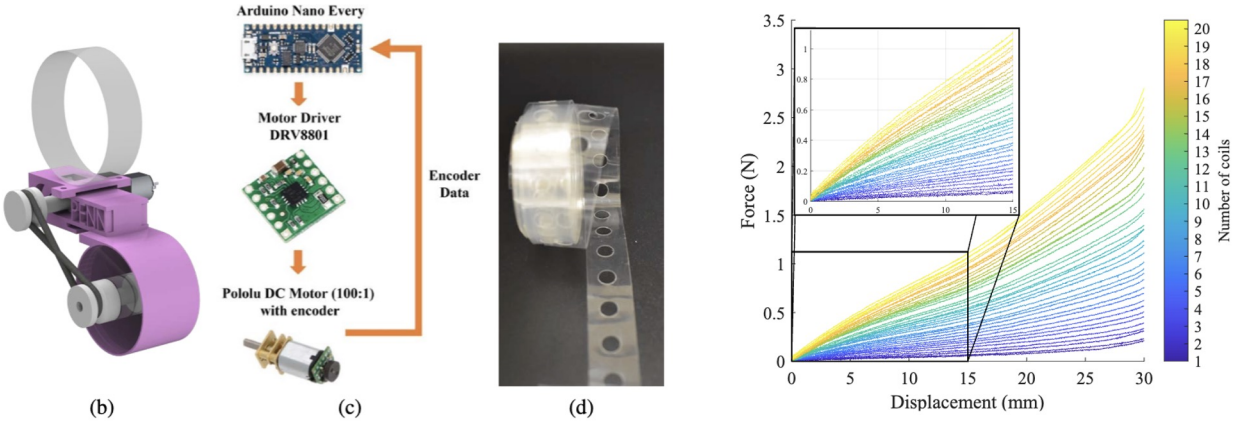
Limit cycles turn active matter into robots

Authors: Martin Brandenbourger, Colin Scheibner, Jonas Veenstra, Vincenzo Vitelli, Corentin Coulais

20

Variable-stiffness springs

- The trick of Shivangi Misra, in Cynthia Sung's lab at UPenn Mech Eng:
  - vary the thickness of a coil of plastic strip, for a robotics application

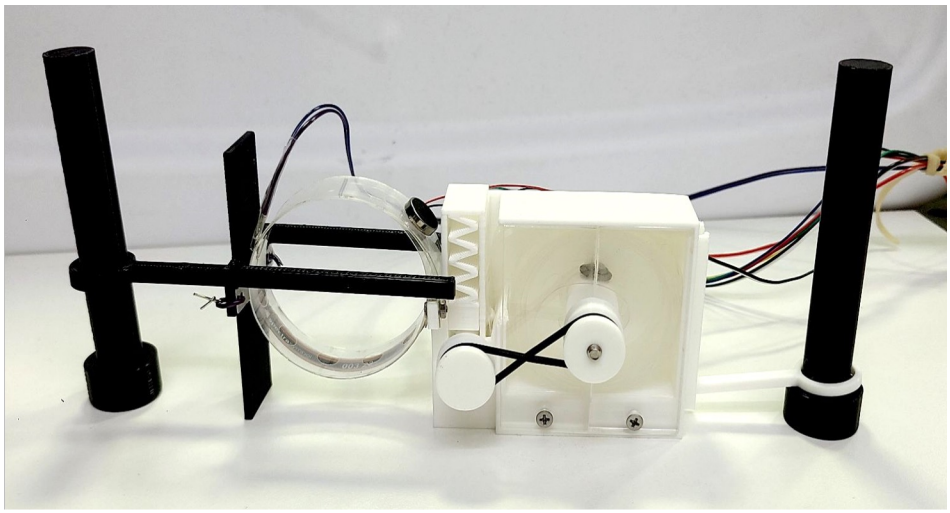


Misra, S., Mitchell, M., Chen, R., & Sung, C. (2023). Design and Control of a Tunable-Stiffness Coiled-Spring Actuator.

21

Automation – in progress

- One edge = piezo strain gauge on coil of variable thickness

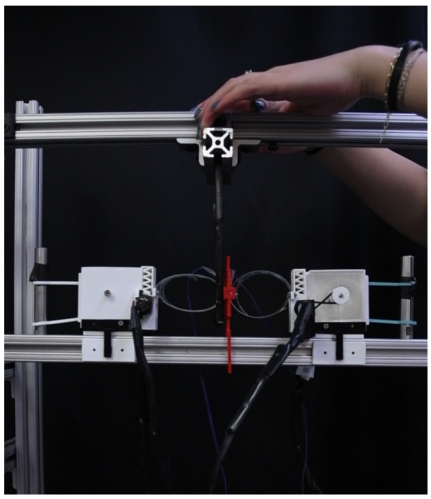


[Lauren Altman, Maggie Miller, Shivangi Misra, Cynthia Sung, DJD (in progress)]

22

Motion Divider with Tunable Stiffneses

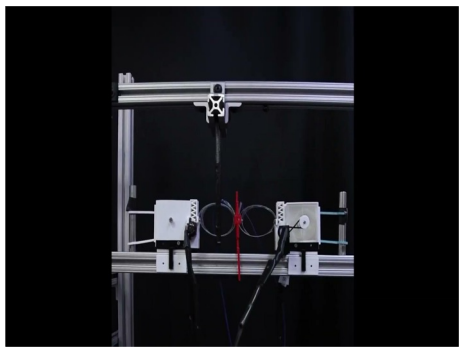
Manual clamping:



Binary update rule:

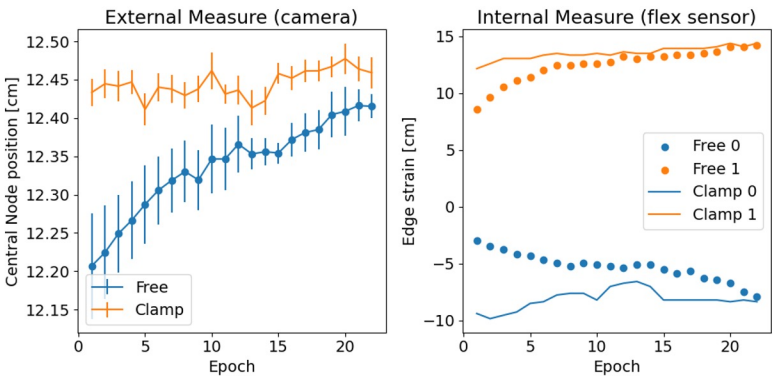
$$\Delta k_i = \begin{cases} +k_{\mathrm{HL}}, & s_i^C > s_i^F \\ -k_{\mathrm{HL}}, & s_i^C < s_i^F \end{cases}$$

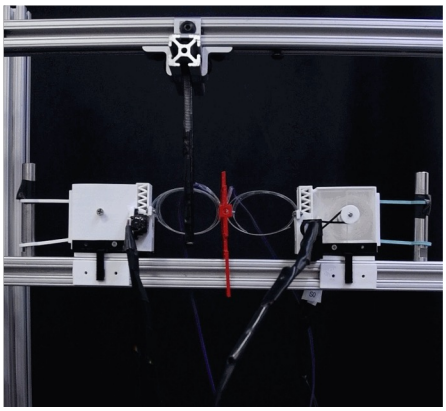
Supervisor performs clamping and global clock (switch learning mode)



23

Motion Divider with Tunable Stiffneses



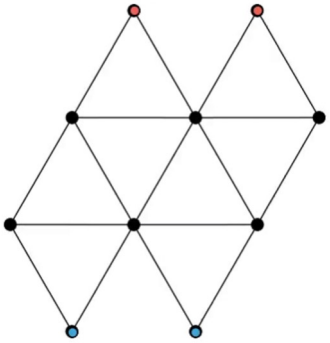


Free states

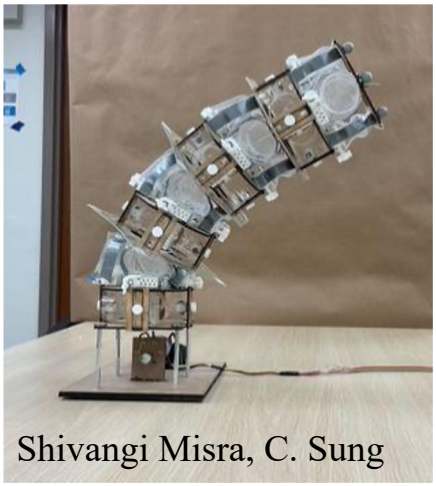
24

More-complex tasks – in progress

Robotic Gripper



Robotic Actuator Tower



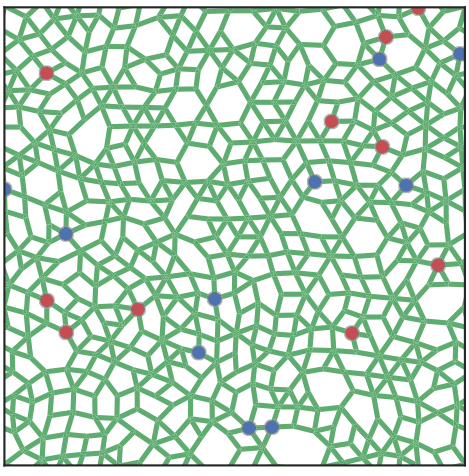
Others?

- Dynamical Behavior
- Material properties

Contrastive Local Learning Networks

- Yes, we can build edges where the edges independently self-adjust using coupled learning so that **outputs** evolve to the desired function of **inputs** -- electrically and mechanically!

- **inputs**
- **outputs**



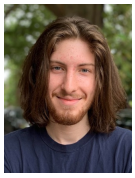
Example tasks/functions:
(1) Outputs are at $\{O_j\}$ when inputs are at $\{I_i\}$
(2) Outputs are a desired linear combo of inputs
(3) Classification: e.g inputs are grayscale pixel data and outputs indicate whether the image is a dog or a cat

Bottom-up learning of complex functionality using local rules in a fully recurrent network, giving brain-like advantages over ANNs

ACKNOWLEDGEMENTS

experiment


Sam Dillavou


Jacob Wycoff


Benjamin Beyer


Lauren Altman

theory


Menachem Stern


Andrea Liu

2nd generation


Marc Miskin

Dinesh Jayaraman, Maggie Miller, Shivangi Misra, Tarunyaa Sivakumar, Cynthia Sung

University of Pennsylvania

LRSM

NSF UPenn MRSEC

27

References

Reviews:

[1] M. Stern and A. Murugan, *Learning Without Neurons in Physical Systems*, Annu. Rev. Condens. Matter Phys. **14**, 417 (2023).

[2] B. Scellier, M. Ernoult, J. Kendall, and S. Kumar, *Energy-Based Learning Algorithms for Analog Computing: A Comparative Study*, NeurIPS (2023).

[3] A. Momeni et al., *Training of Physical Neural Networks*, arXiv:2406.03372.

Specific local learning rules:

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

[2] 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).

[3] 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).

[4] 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. **120**, e2219558120 (2023).

[5] V. R. Anisetti, A. Kandala, B. Scellier, and J. M. Schwarz, *Frequency Propagation: Multimechanism Learning in Nonlinear Physical Networks*, Neural Comput. **36**, 596 (2024).

Electronic realizations of coupled learning:

[1] S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, *Demonstration of Decentralized Physics-Driven Learning*, Phys. Rev. Appl. **18**, 014040 (2022).

[2] J. F. Wycoff, S. Dillavou, M. Stern, A. J. Liu, and D. J. Durian, *Desynchronous Learning in a Physics-Driven Learning Network*, J. Chem. Phys. **156**, 144903 (2022).

[3] M. Stern, S. Dillavou, D. Jayaraman, D. J. Durian, and A. J. Liu, *Training Self-Learning Circuits for Power-Efficient Solutions*, APL Mach. Learn. **2,** 016114 (2024).

[4] 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. **121**, e2319718121 (2024).

Mechanical realizations of coupled learning:

[1] L. Altman, M. Stern, A. J. Liu, D. J. Durian, *Experimental Demonstration of Coupled Learning in Elastic Networks*, arXiv:2311.00170

28