Notes on connection between cost & physical Hessians. Stern, Guzman, Martins, Liu, Balasubramanian arXiv 2406.09689

Linear resistor networks trained using double optimization so at minimum in cost & physical landscapes.

Network has N nodes V_α connected by N_e edges k_i

$H_{\alpha\beta}$ is physical Hessian (or graph Laplacian)

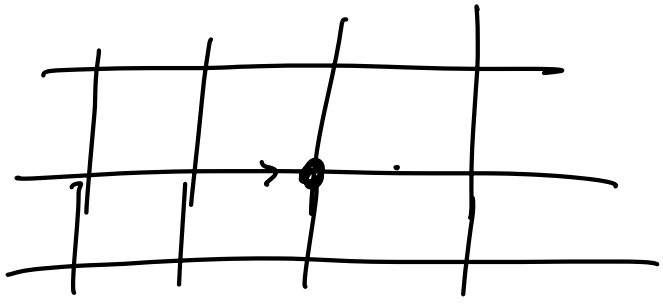
$$H_{\alpha\beta} = \frac{\partial^2 P}{\partial V_\alpha \partial V_\beta}$$

symmetric $N \times N$

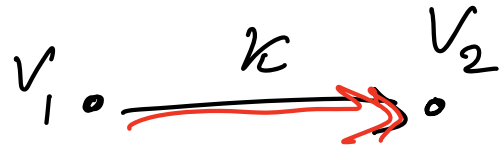
For linear resistors,

$$P = \frac{1}{2} V^T H V$$

where $V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_\alpha \\ \vdots \\ V_N \end{pmatrix}$



recall for single resistor



$$P = \frac{1}{2} K (V_1 - V_2)^2 = \frac{1}{2} K (V_2 - V_1)^2$$

so in this case $H =$

	node 1	node 2
node 1	K	$-K$
node 2	$-K$	K

$$P = \frac{1}{2} \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} K & -K \\ -K & K \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$$

Introduce adjacency matrix Δ_{ia}

where $\Delta_{ia} = -1, 0, 1$ (put arrow on diagram above)

$\Delta =$

	node 1	node 2
edge 1	-1	1

Define $\kappa_{ij} = \kappa_i \delta_{ij}$

Then $\Delta^T \kappa \Delta = \begin{pmatrix} -1 & \\ & 1 \end{pmatrix} (\kappa) \begin{pmatrix} -1 & 1 \end{pmatrix}$

$$= \begin{pmatrix} \kappa & -\kappa \\ -\kappa & \kappa \end{pmatrix} = H$$

So $H_{\alpha\beta} = \Delta^T_{\alpha i} \kappa_{ij} \Delta_j$

Suppose we have N_{input} voltages, $\tilde{V}_{\alpha,in}$ applied to input nodes, specified by vector S_α

Then we will call this the free state (sources applied, outputs left free).

To satisfy Kirchhoff's law,

$P^F = \frac{1}{2} V^T H V - V^T S$ must be minimized.

Why this additional term?

Add a Lagrange multiplier

$$P = \underbrace{\frac{1}{2} V^T H V}_{\text{all other nodes}} + \lambda \cdot (V - \tilde{V})$$

λ: vector of Lagrange multipliers

Clearly, λ has units of current.

Construct an extended Hessian, with N_{input} extra rows & columns except for -1 on the diagonal at each of the input node positions in the matrix.

Let U be augmented voltage vector

$$\begin{pmatrix} V_1 \\ \vdots \\ V_N \\ V_1^{input} \\ \vdots \\ V_{N_{input}}^{input} \end{pmatrix}$$

where $\lambda = \tilde{V}^{input}$

Define the source vector

$$S = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ S_1^{input} = V_1 \\ \vdots \\ S_{N_{input}}^{input} = V_{N_{input}} \end{pmatrix}$$

Then

$$P = \frac{1}{2} U \hat{H} U^T - U^T S$$

The power is minimized if the nodes have the voltages

$$V^F_{(K,I)} = H^{-1}(K)\, S$$

Define the cost function

$$C = \frac{1}{2} \sum_{\substack{\text{tasks} \\ n}}^{n_T} \left(V_\gamma^F - V_\gamma^{\text{desired}} \right)^2$$

↑ label

$$= \frac{1}{2} \sum_\gamma c_\gamma^2$$

Double optimization leads to minimum $C = 0$ of C and P. Near minimum, C is described by cost Hessian:

$$H_{ij} = \frac{\partial^2 C}{\partial K_i \partial K_j}$$

$N_e \times N_e$ matrix

For a system with n_T constraints, H_{ij} has n_T high eigenvalues

Sagun, Bottou, Le Cun 1611.07476 (2016)

Consider case of single task (constraint)

$$C = \frac{1}{2} c^2$$

then $$\mathcal{H}_{ij} = \frac{\partial^2 C}{\partial \kappa_i \partial \kappa_j} = \underbrace{\frac{\partial c}{\partial \kappa_i}\frac{\partial c}{\partial \kappa_j}}_{\mathcal{H}^{sat}} + \underbrace{c\,\frac{\partial^2 c}{\partial \kappa_i \partial \kappa_j}}_{0 \;\mathcal{H}^{dyn}}$$

when $c = 0$

assume double optimization has succeeded so $\mathcal{H}^{dyn} = 0$

$$\mathcal{H}^{sat} = \underbrace{g_i g_j^T}_{\text{outer product of } g \text{ w/ itself}}$$

vector

$$g_i = \frac{\partial c}{\partial \kappa_i}$$

Recall that $\mathcal{H}^{sat}$ as 1 high eigenmode for 1 constraint: g_1 is this eigenmode. (show slide again)

Note also that

$$\frac{\partial C}{\partial \kappa_i} = c\,\frac{\partial c}{\partial \kappa_i} = c\,g_i$$

going down gradient ✓ is going in direction of stiff mode!

Now recall that c depends on free state V^F:

$$g_i = \frac{dc}{dV^F} \cdot \frac{dV^F}{dK_i}$$

But $V^F = H^{-1}(K)S$

So $$\frac{dV^F}{dK_i} = -H^{-1}\frac{dH}{dK_i}H^{-1}S$$

$$g_i = \frac{dc_r}{dV_F} H^{-1} \frac{dH}{dK_i} H^{-1} S$$

But $H = A^T K A$

Let $\Delta^{(i)}$ correspond to the vector of nodes in the i^{th} row of $\Delta_{i\alpha}$ so that $\Delta^{(i)}_{\alpha} = \Delta_{i\alpha}$

Then $$\frac{dH}{dr_{ij}} = \Delta^{(i)T} \Delta^{(j)}$$

And $$g_i = -\frac{dC}{dV_F} H^{-1} \Delta^{(i)T} \Delta^{(i)} H^{-1} S$$

So $$\mathcal{H}^{sat}_{ij} = g_i \cdot g_j \sim (H^{-1})^4$$

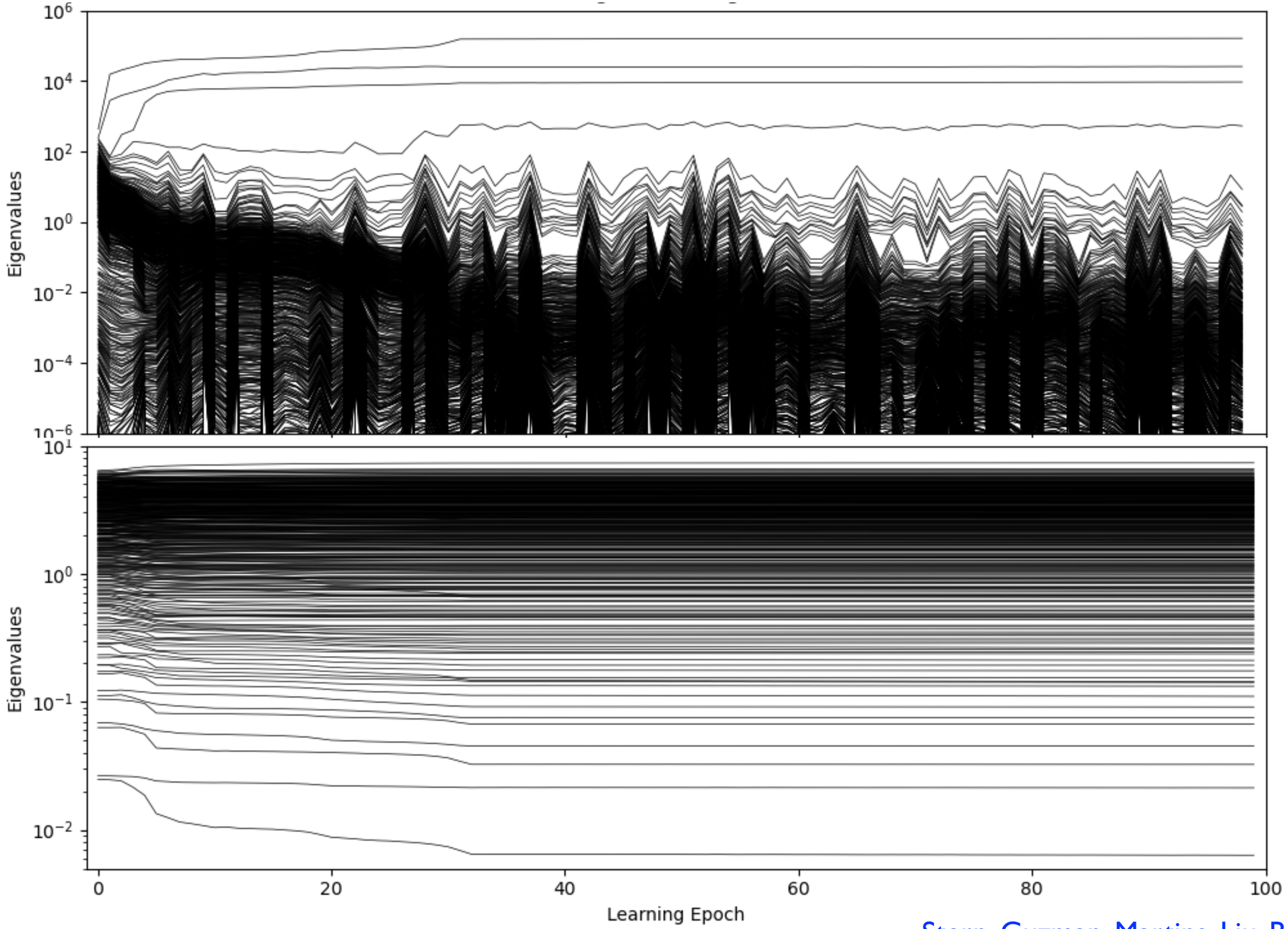
curvature around min. in cost landscape is related to curvatures around min in physical landscape! The 2 become coupled via double optimization process.

Going further, we can show that

$\mathcal{H}^{dyn} \sim \sqrt{C}\, H^{-3} \to 0$ when $C \to 0$.

Evolution of Eigenvalues with Learning of Linear Regression

Learning Hessian H_L



Physical Hessian H_P

Highest eigenvalues increase

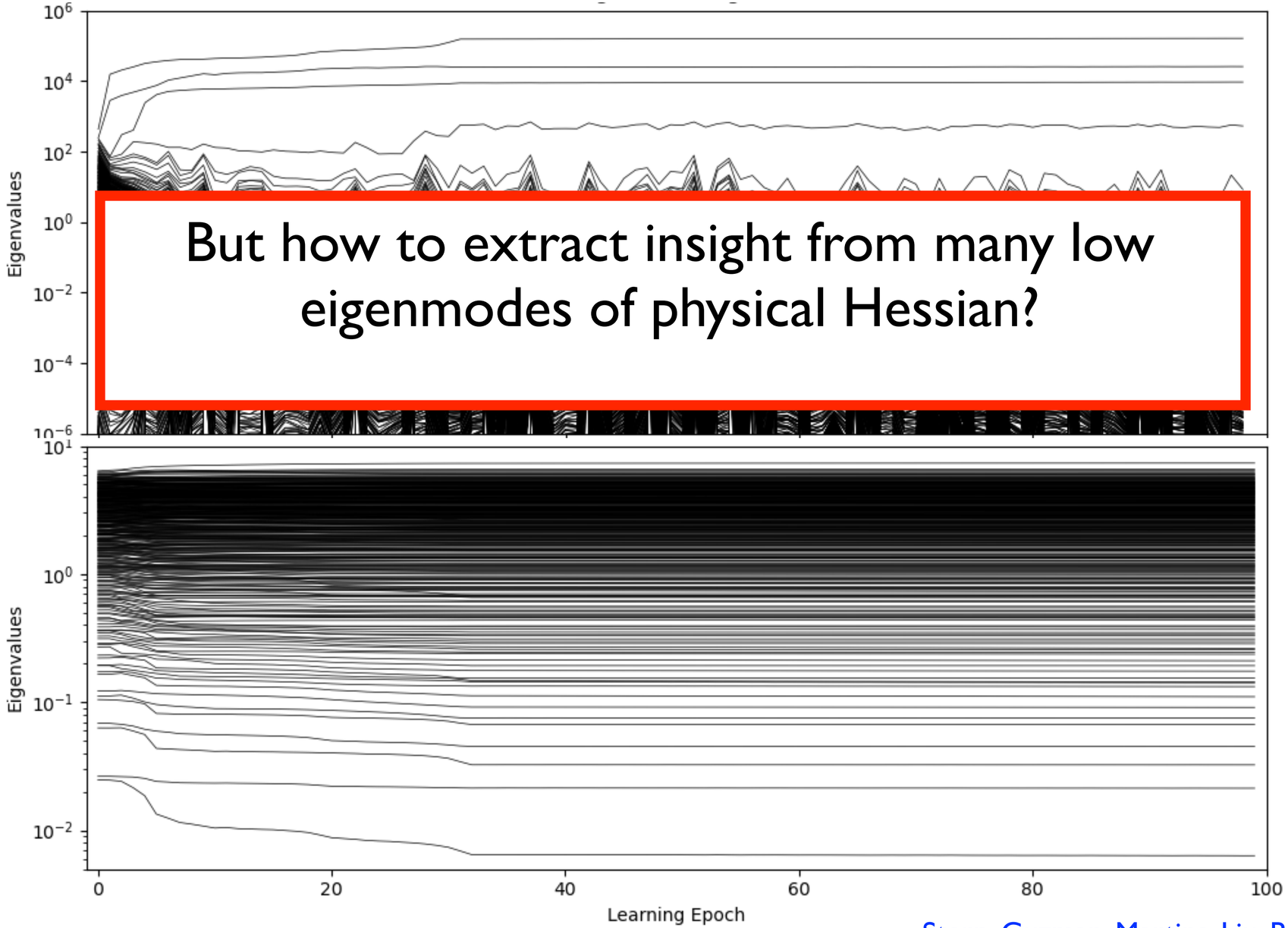
Lowest eigenvalues decrease

$$H_L \sim \left[H_P^{-1}\right]^4$$

Stern, Guzman, Martins, Liu, Balasubramanian arXiv (2024)

Evolution of Eigenvalues with Learning of Linear Regression

Learning Hessian H_L



Physical Hessian H_P

Highest eigenvalues increase

Lowest eigenvalues decrease

$$H_L \sim \left[H_P^{-1}\right]^4$$

Stern, Guzman, Martins, Liu, Balasubramanian arXiv (2024)

Maximally Localized Basis

- Start with low eigenvectors of physical Hessian $\mathcal{B} = \{\Psi_1, \Psi_2, \Psi_3, ...\}$
- Construct spreading functional that measures spatial extent of eigenvectors

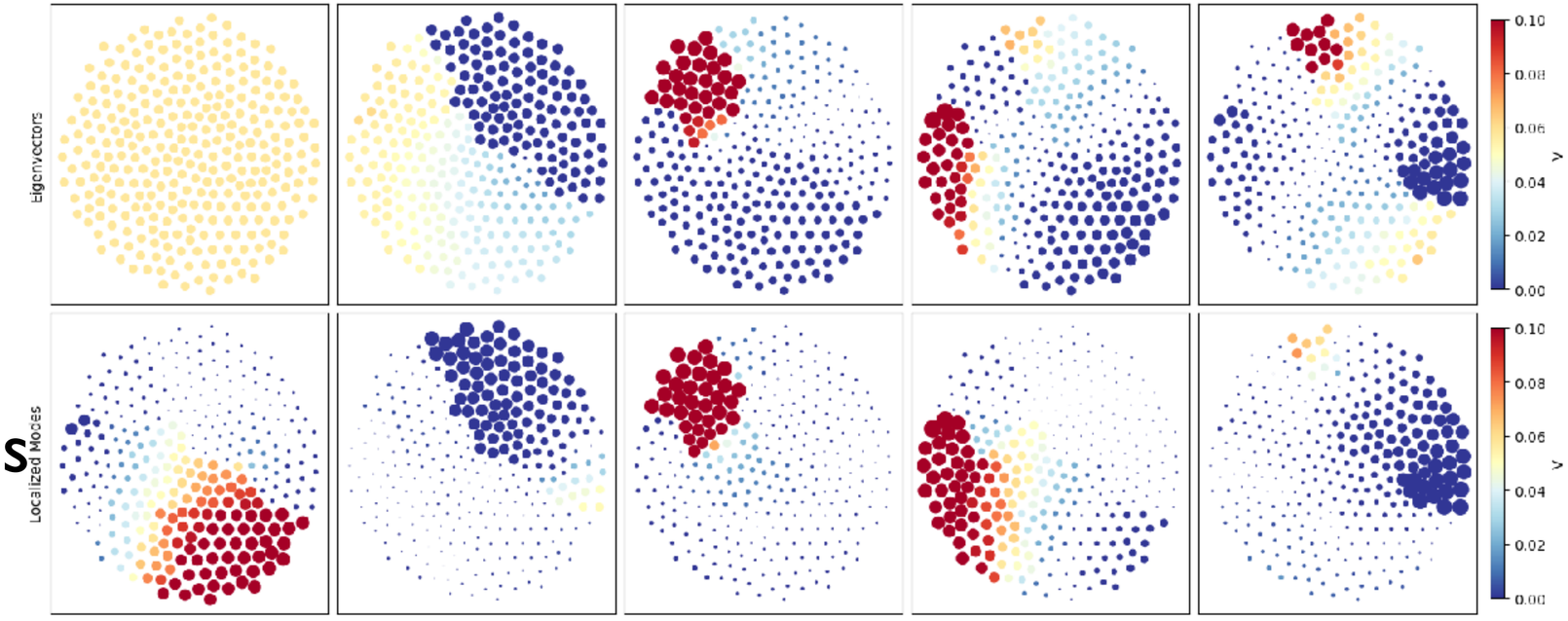
$$\Omega[\mathcal{B}] = \sum_i \left(\Psi_i^{\mathrm{T}} \mathcal{R}^2 \Psi_i - (\Psi_i^{\mathrm{T}} \mathcal{R} \Psi_i)^2 \right)$$

- Minimize functional
- Minimum of functional gives new basis vectors that span same space

$$\mathcal{B}' = \{\Psi'_1, \Psi'_2, \Psi'_3, ...\}$$

Lowest eigenvectors

Maximally localized basis



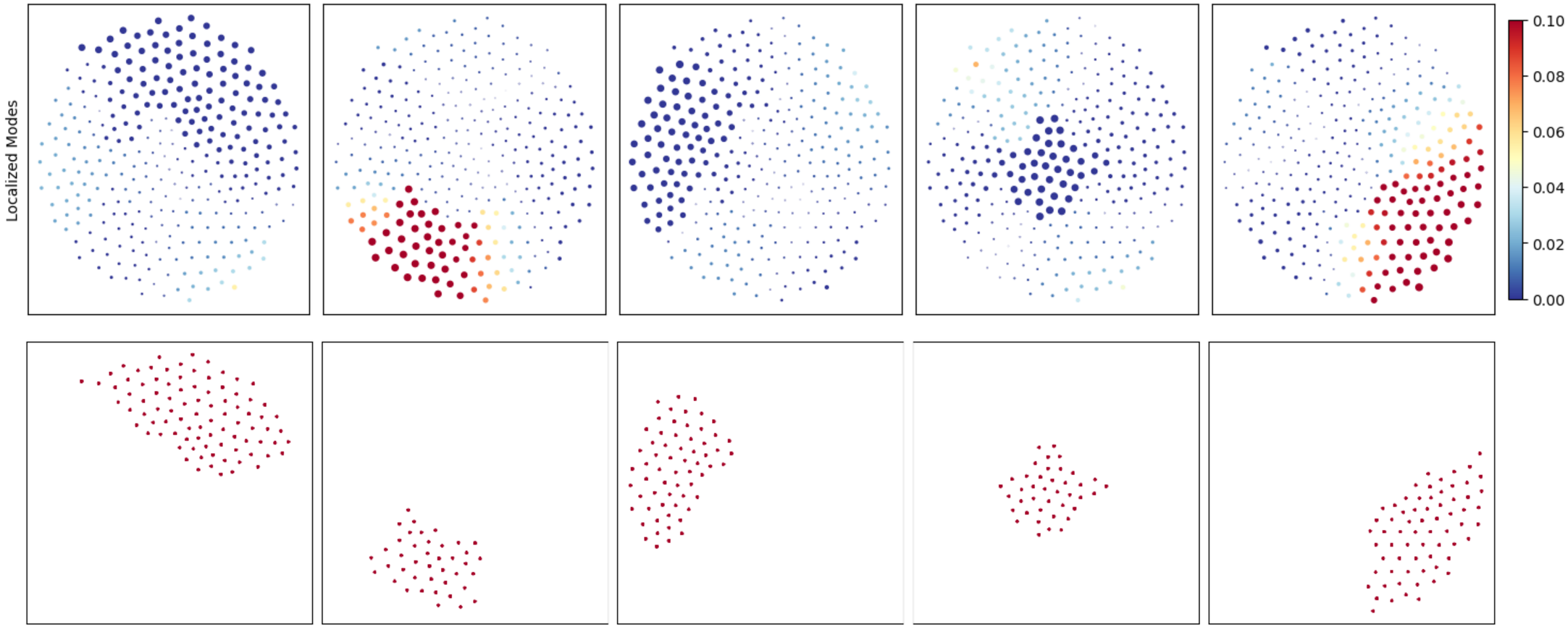
Felipe Martins, Marcelo Guzman

Getting Insight from Maximally Localized Basis

Binarize max localized modes

Construct boundaries between regions of different V

Combine boundaries together

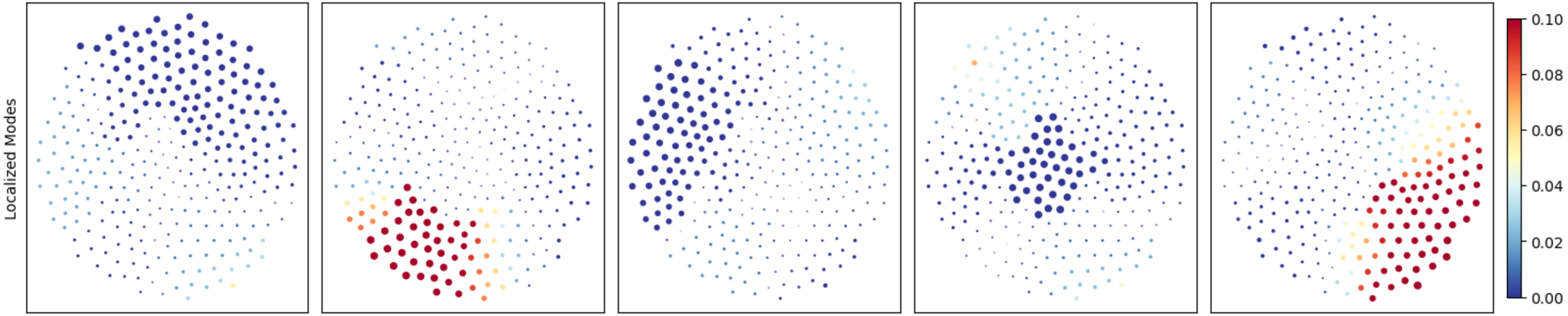


Getting Insight from Maximally Localized Basis

Binarize max localized modes

Construct boundaries between regions of different V

Combine boundaries together

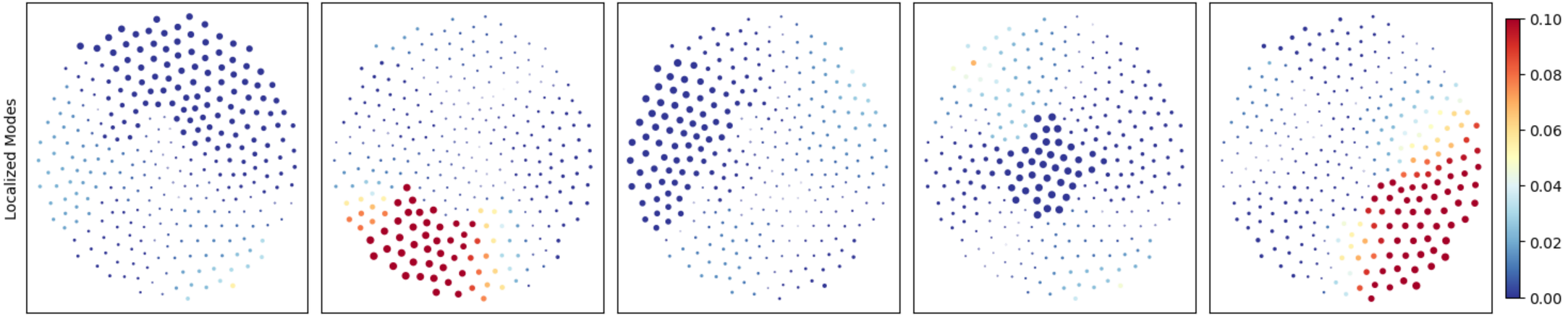


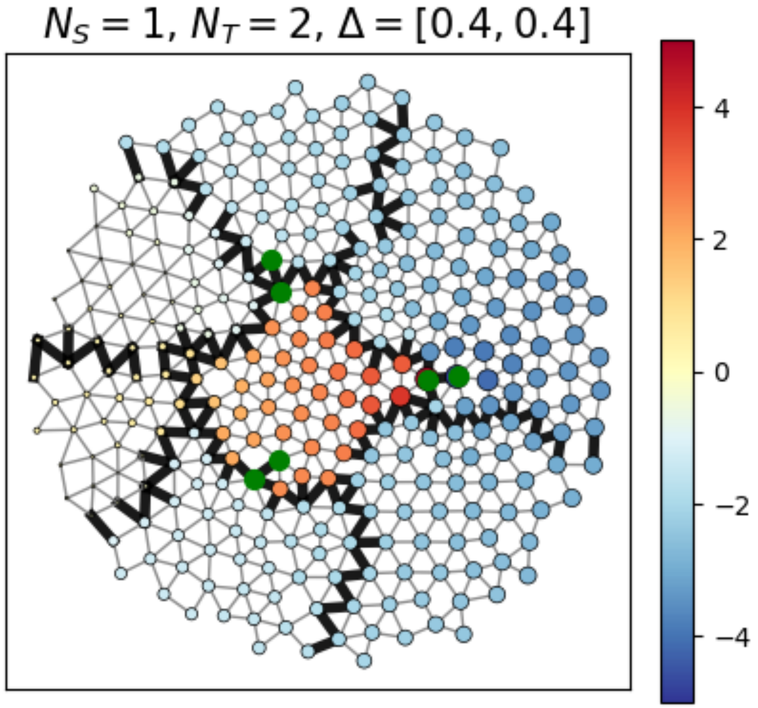
Getting Insight from Maximally Localized Basis

Binarize max localized modes

Construct boundaries between regions of different V

Combine boundaries together

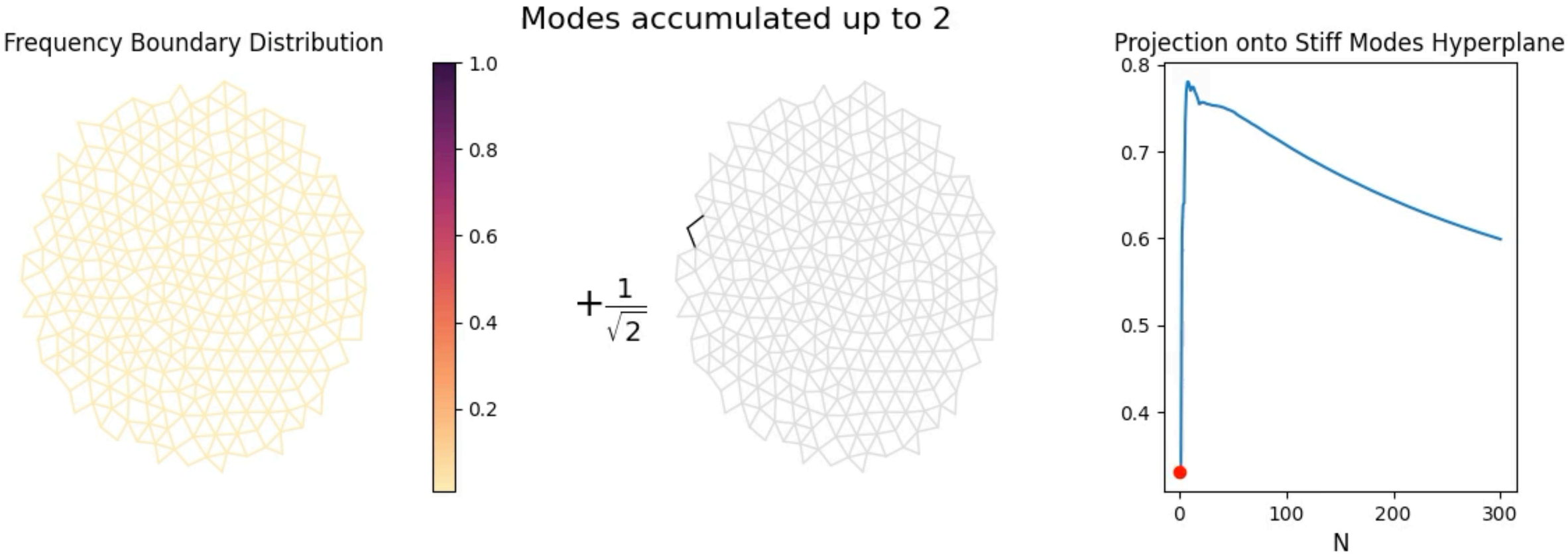




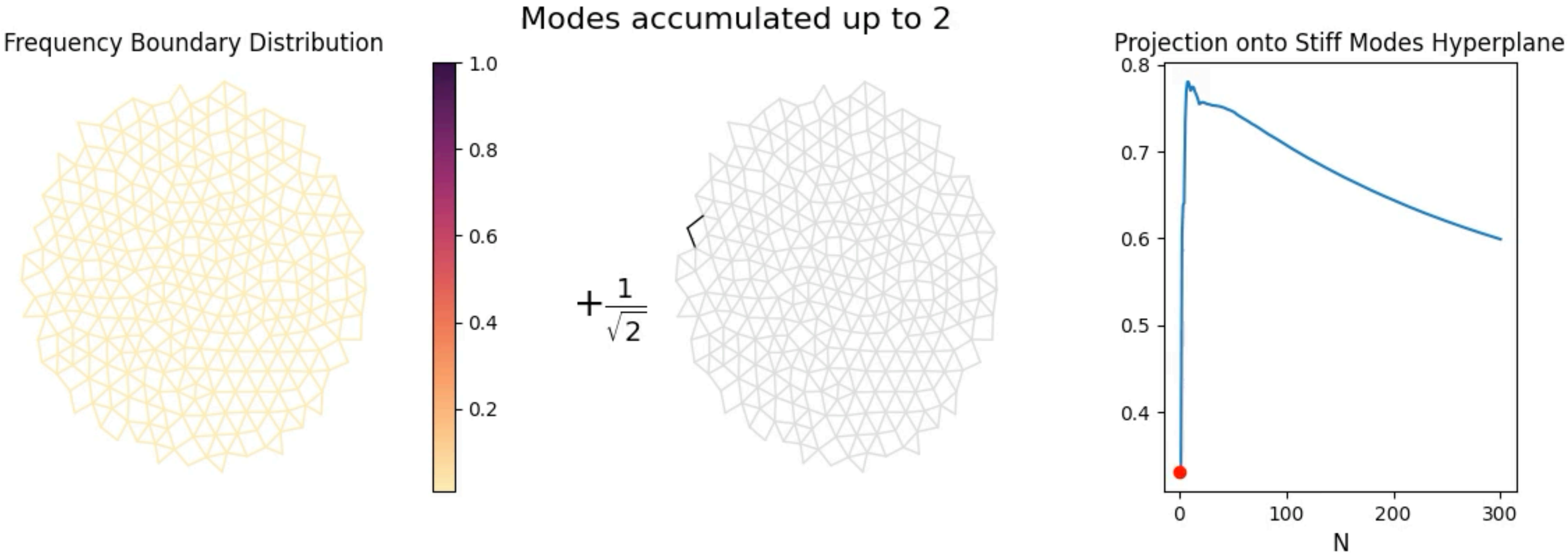
Boundary Edge and Boundary Edge Frequency

- The number of boundary edges ~ \sqrt{N_m}
  - For each N, assign index n_i to each edge I so that n_i=1 if the edge is a boundary edge, 0 if not
  - Define a weighted boundary frequency for each edge, f_i = \sum{N_m=1, N_mtot= dN} n_i /sqrt{N_m}

Boundary Frequency vs. Number of Modes

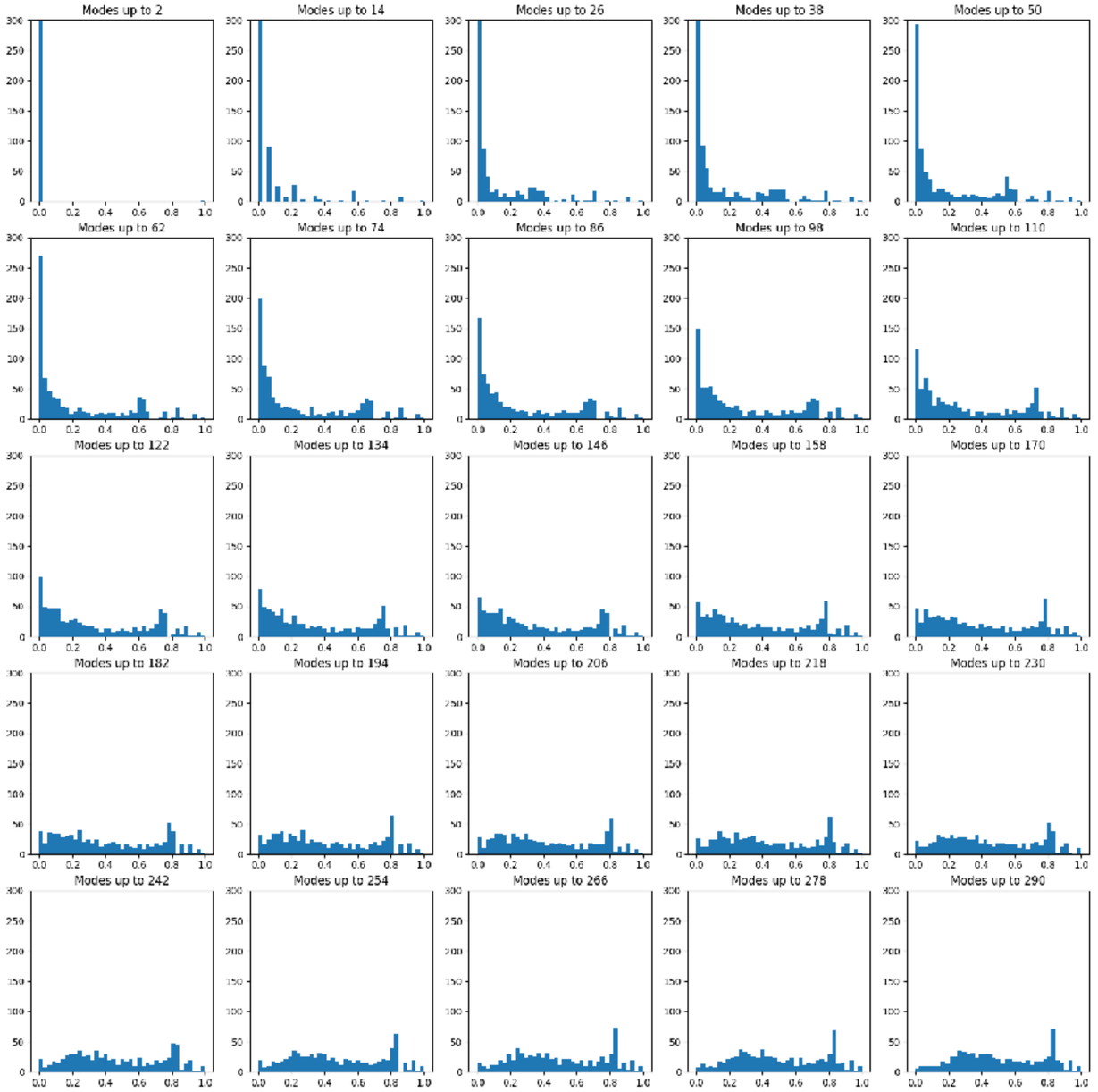


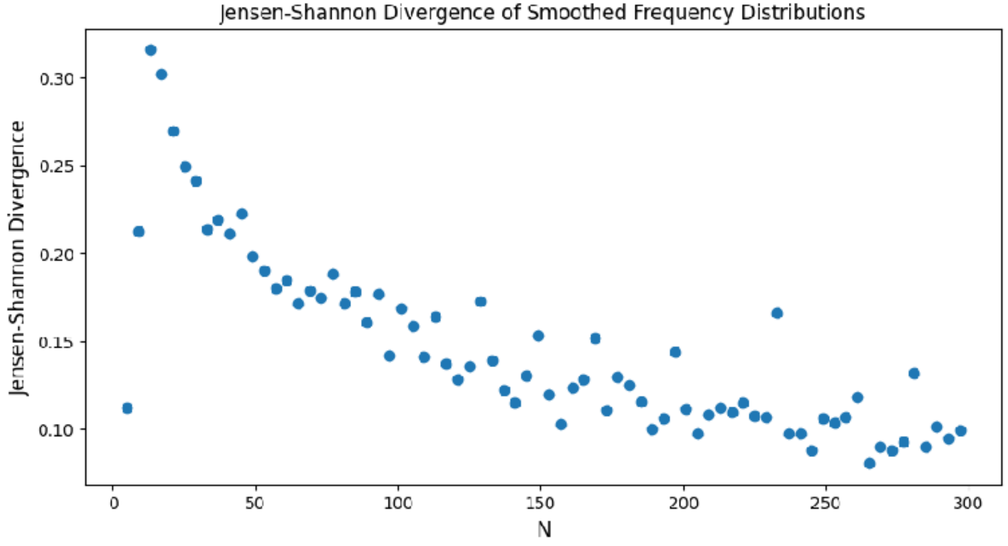
Boundary Frequency vs. Number of Modes



How to Determine Reasonable N_m Without Knowing Cost Hessian

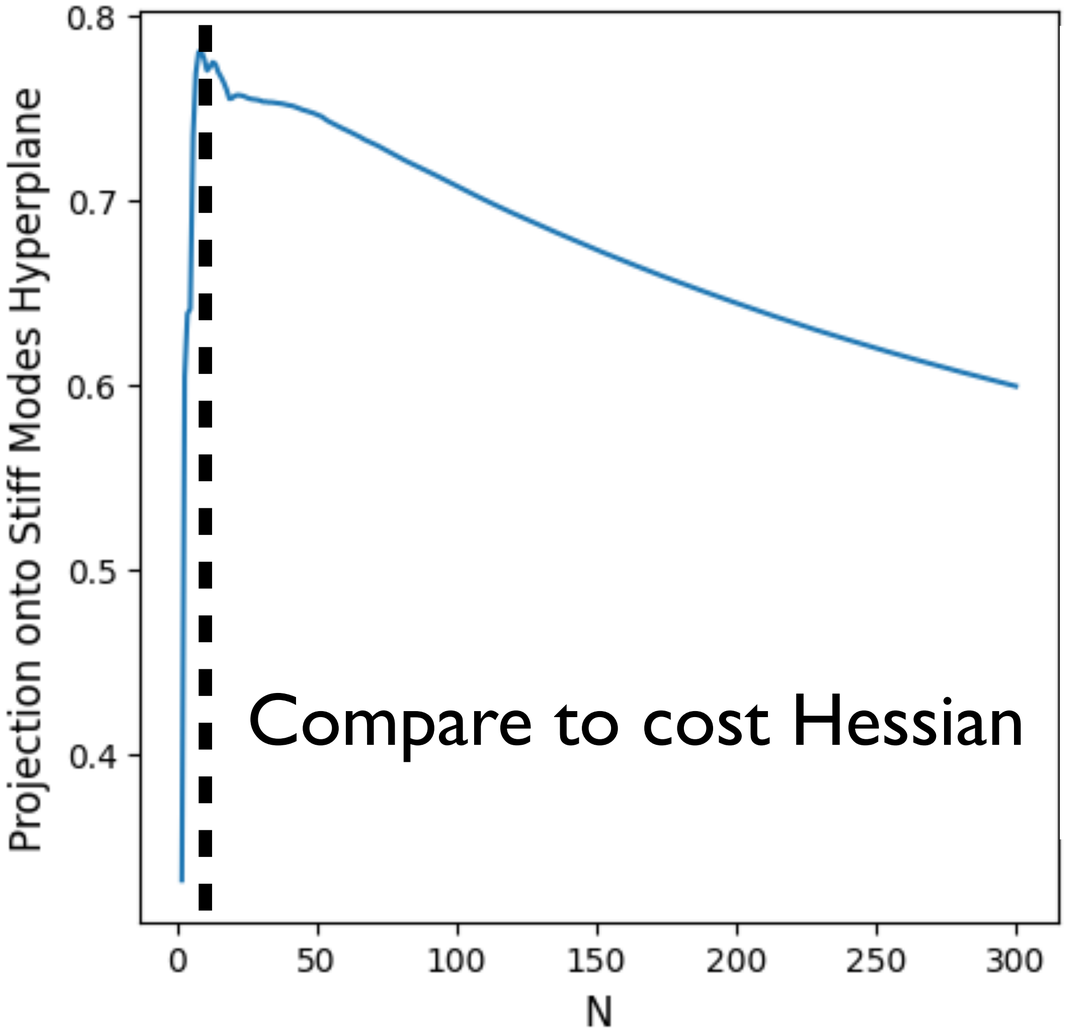
N_m

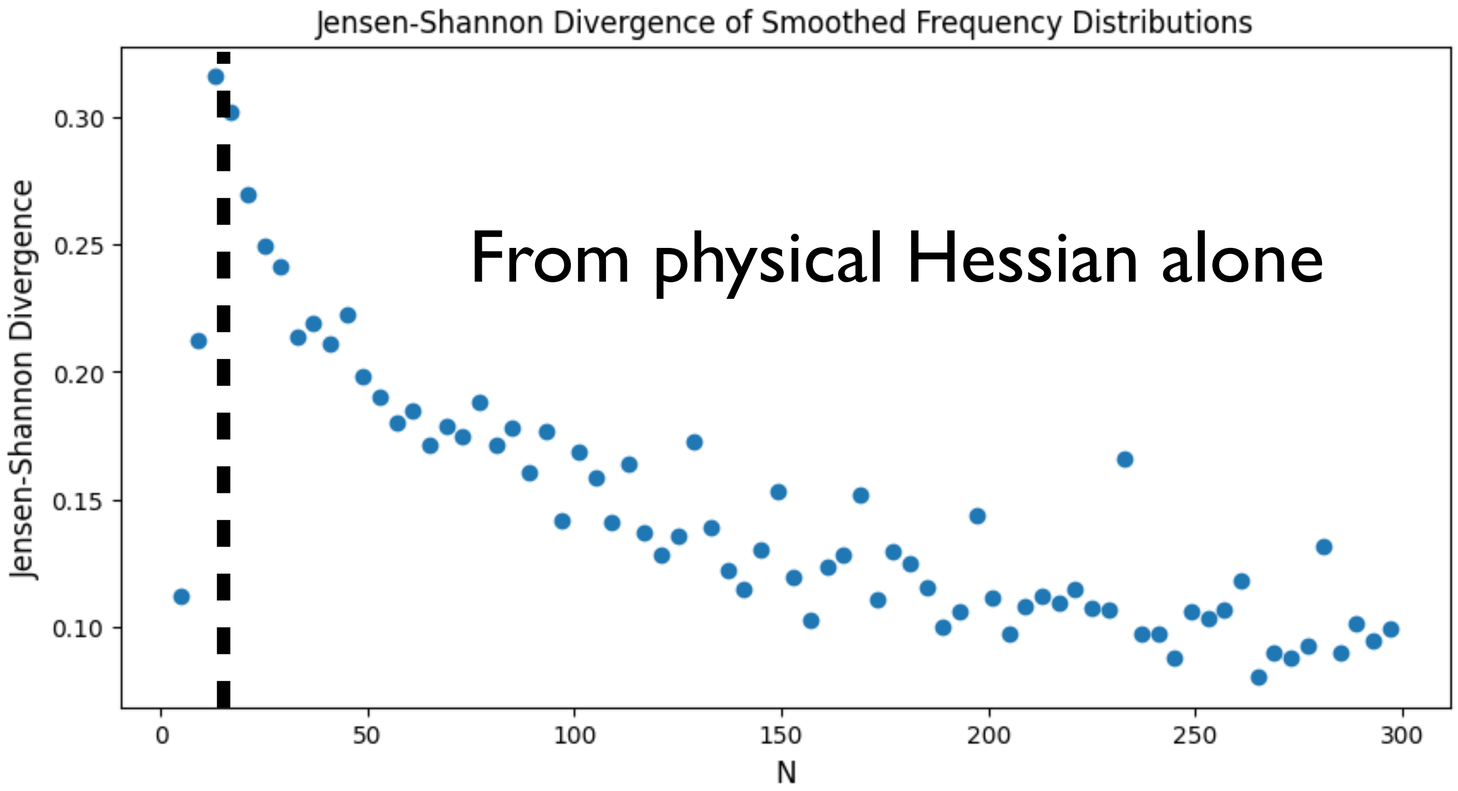




Compare Optimal N_m

- Number of modes to retain is similar



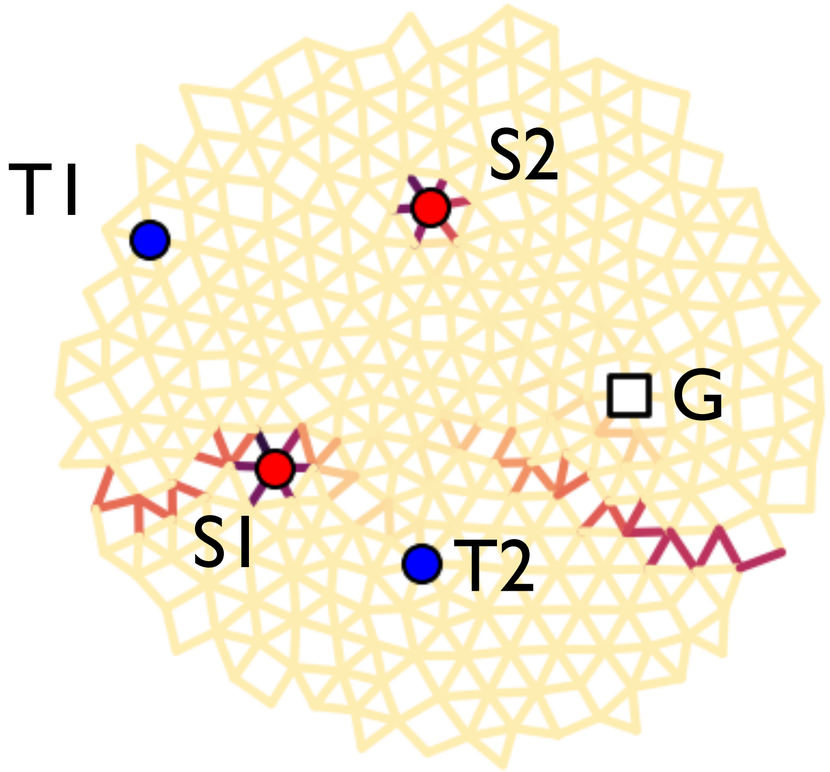


Physical Insight Into How Task Is Accomplished

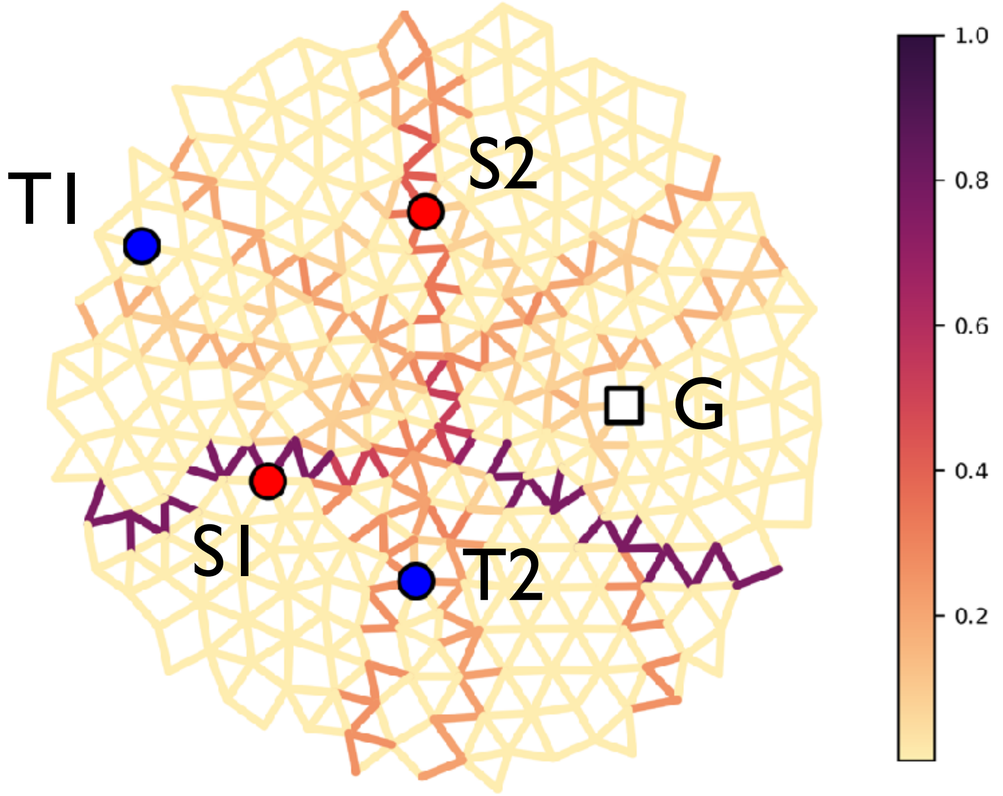
$$V_{T1} = 0.15V_{S1} + 0.3V_{S2}$$
$$V_{T2} = 0.2V_{S1} + 0.1V_{S2}$$

Barriers direct flow to make target more sensitive to one source than the other

Felipe Martins, Marcelo Guzman



from cost Hessian



from physical Hessian

"Statistical Physics" from Big Data

- Summary of our approach:
  - Create ensembles of networks that learn the same thing
  - Identify microscopic origin of learning with PH analysis/Hessian analysis
  - Dimensional reduction to relevant microscopic quantities: mechanism reduced to
    - sectors/strain pathways (PH)
    - Barriers/pipelines (physical Hessian)
- ANNs vs learning metamaterials
  - In ANNs, must rely on cost Hessian (or NTK) for insight
    - So must know cost function
    - That's ok because cost function is required to train
  - Learning metamaterials learn by local rules—don't use cost function
  - Can use physical Hessian to gain insight into emergent learning without knowing what network was trained to learn

Structure/Function Relation from "Statistical Physics"

- Holy Grail for proteins: understand sequence/mechanics/structure/function relations
  - ability to create ensembles of networks with desired functions allows study of statistics
  - new application of persistent homology for identifying structure/function relation that can be applied to real systems

- Learning sculpts both the learning landscape and the physical landscape and leaves detectable signatures
  - e.g. connection between allostery and low-frequency vibrational modes (Jacobs/Rader/Kuhn/MF Thorpe Proteins (2001), Yan/Ravasio/Brito/Wyart (2017), Husain/Murugan Mol Biol Evol (2020)

Prove that we can replace cost with contrast as $\eta \to 0$.

Recall that there is a Lyapunov fcn e.g. power P for linear resistor networks that is minimized to satisfy Kirchhoff's law.

omit

For nonlinear resistors,

$$P = \text{co-content} = \frac{1}{2} \sum_{i \text{ edges}} \int_{V_{i1}}^{V_{i2}} dV \; I(V) V$$

In the free state we are applying inputs I_α at a subset of nodes; call this vector ξ_α. Let V_α be the node voltages & K_i be the conductances.

So $P_F = P_F(V_F, K, \xi)$

Note that V_F are the voltages that minimize P_F so

$$\frac{\partial}{\partial V} P_F(V, \kappa)\Big|_{V=V_F} = 0.$$

Now consider some cost fcn

$$C = \frac{1}{2} \tilde{\kappa} \sum_{\beta} \left(V_\beta - \tilde{V}_\beta \right)^2$$

($\tilde{V}_\beta$: desired; $\tilde{\kappa}$: turns error into power!; β: target nodes)

Define a new "power" as $P = P_F + \eta C$

Then following same linear algebra as before for applying inputs

$$P = P + \lambda \cdot (V - \tilde{V})$$ where λ is a vector of Lagrange multipliers,

we see this looks the same if we set $\lambda = \frac{1}{2}\tilde{\kappa}(V - \tilde{V})$. λ acts like a current

This is the clamped power.

Define the clamped state as V_c

with

$P_c(^*V_c, \kappa, \eta)$ where V_c minimizes P_c

Now define the contrast function

$$C = \eta^{-1}\left[P_c - P_F\right]$$

Show that

$$\frac{d}{d\kappa} C(V_F(\kappa)) = \lim_{\eta \to 0} \frac{\partial}{\partial \kappa} C(V_F, \kappa, \eta)$$

↑ total derivative — ↑ partial derivative

clamped state

$$\frac{\partial}{\partial V} P_c(V, \kappa, \eta)\Big|_{V=V_c} = 0.$$

$$\frac{dP_c}{d\kappa}\Big|_{V_c} = \frac{\partial P_c}{\partial \kappa}\Big|_{V_c} + \cancel{\frac{\partial P_c}{\partial V}\Big|_{V_E}} \overset{0}{\cancel{\frac{\partial V}{\partial \kappa}\Big|_{V_E}}}$$

$$\frac{dP_c}{d\eta}\Big|_{V_c} = \frac{\partial P_c}{\partial \eta}\Big|_{V_c} + \cancel{\frac{\partial P}{\partial V}\Big|_{V_c}} \overset{0}{\cancel{\frac{\partial V}{\partial \eta}\Big|_{V_c}}}$$

$$\frac{d}{d\kappa}\left(\frac{dP_c}{d\eta}\right)\Big|_{V_c} = \frac{d}{d\eta}\frac{dP_c}{d\kappa}\Big|_{V_c}$$

But $P_c = P_F + \eta C$

$$\frac{d}{d\kappa}\frac{d}{d\eta}(P^F + \eta\mathcal{C}) = \frac{d\mathcal{C}}{d\kappa}$$

and $$\frac{d}{d\eta}\frac{dP_C}{d\kappa} = \frac{d}{d\eta}\left[\frac{\partial P_C(V,\kappa,\eta)}{\partial\kappa}\right]\Big|_{V_C} = \frac{1}{\eta}\lim_{\eta\to 0}\frac{\partial}{\partial\kappa}\left[P_C(V,\kappa,\eta)\Big|_{V_C} - P_C(V,\kappa,0)\Big|_{V_C}\right]$$

$$= \frac{1}{\eta}\lim_{\eta\to 0}\frac{\partial}{\partial\kappa}\left[P_C(V,\kappa,\eta)\Big|_{V_C} - P_F(V,F)\right]$$

So $$\frac{d\mathcal{C}}{d\kappa} = \frac{1}{\eta}\lim_{\eta\to 0}\frac{\partial}{\partial\kappa}\left[P_C(V,\kappa,\eta)\Big|_{V_C} - P_F(V,F)\right]$$

$$\frac{d\mathcal{C}}{d\kappa} = \lim_{\eta\to 0}\frac{\partial C}{\partial\kappa}$$

Subtleties: nudge towards desired voltage w/ current or voltage
EP CL

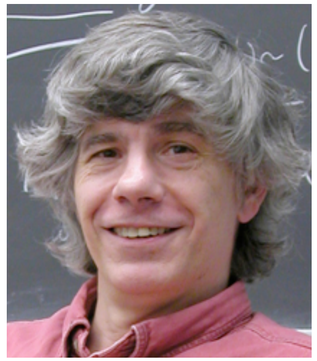
For CL there is an extra term that doesn't vanish as $\eta \to 0$

see Scellier, Ernoult, Kendall, Kumar
Pro Neur IPS (2023) App B

Thanks to

Ben
Pisanty

Jovana
Andrejevic



Sid
Nagel

DOE Biomolecular Materials

Simons Foundation

Coupled learning for spring networks

$C = \frac{1}{2}(\text{desired response} - \text{free response})^2$

$C = \frac{1}{\eta}\left[E^{cl} - E^{free}\right] \geq 0$ due to physics

Just as before, $\frac{\partial C}{\partial k}$ approximates $\frac{dC}{dk}$

↑ spring constant vector

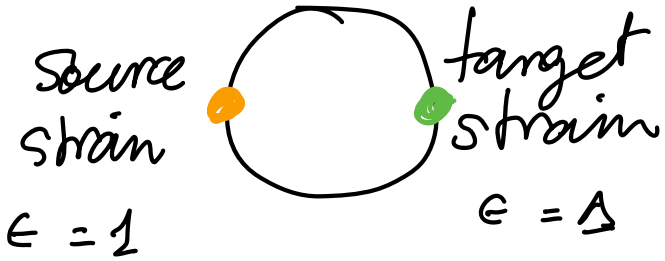
$$\dot{k}_j = -\frac{\alpha}{\eta}\frac{\partial}{\partial k_j}\left[E^{cl} - E^{free}\right]$$

$$= -\frac{\alpha}{2\eta}\frac{\partial}{\partial k_j}\sum_i k_i\left(\left[l_i^C - l_0\right]^2 - \left[l_i^F - l_0\right]^2\right)$$

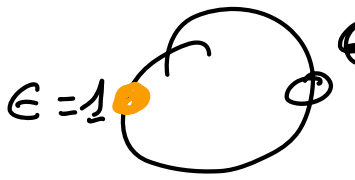
$$= -\frac{\alpha}{2\eta}\left[\left(l_j^C - l_0\right)^2 - \left(l_j^F - l_0\right)^2\right]$$

Task: Cooperative binding vs allostery

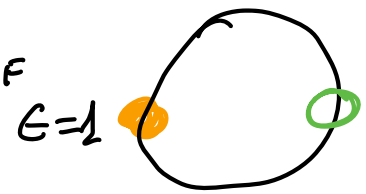
allostery



free BC:



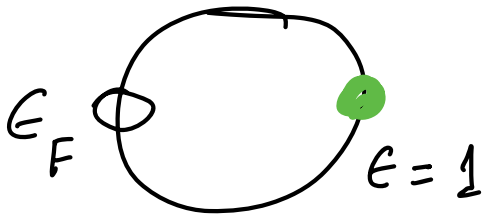
clamped



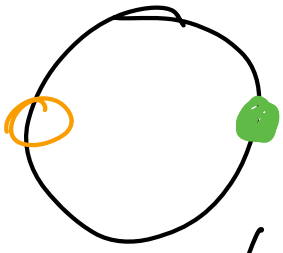
$\epsilon = \epsilon_F + \eta(\Delta - \epsilon_F)$

cooperative binding: alternate between case above and

free



clamped

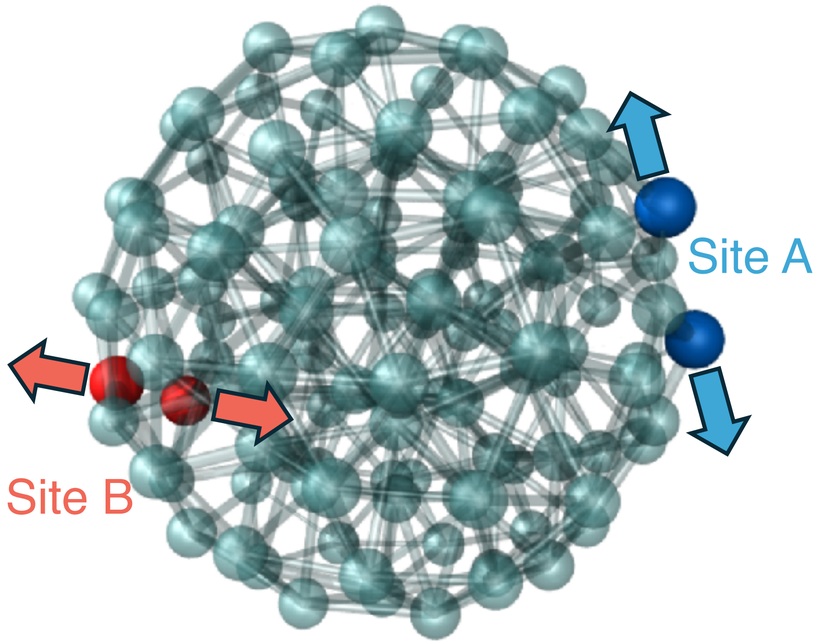


$$\epsilon = \epsilon_F + \eta(\Delta - \epsilon_F)$$

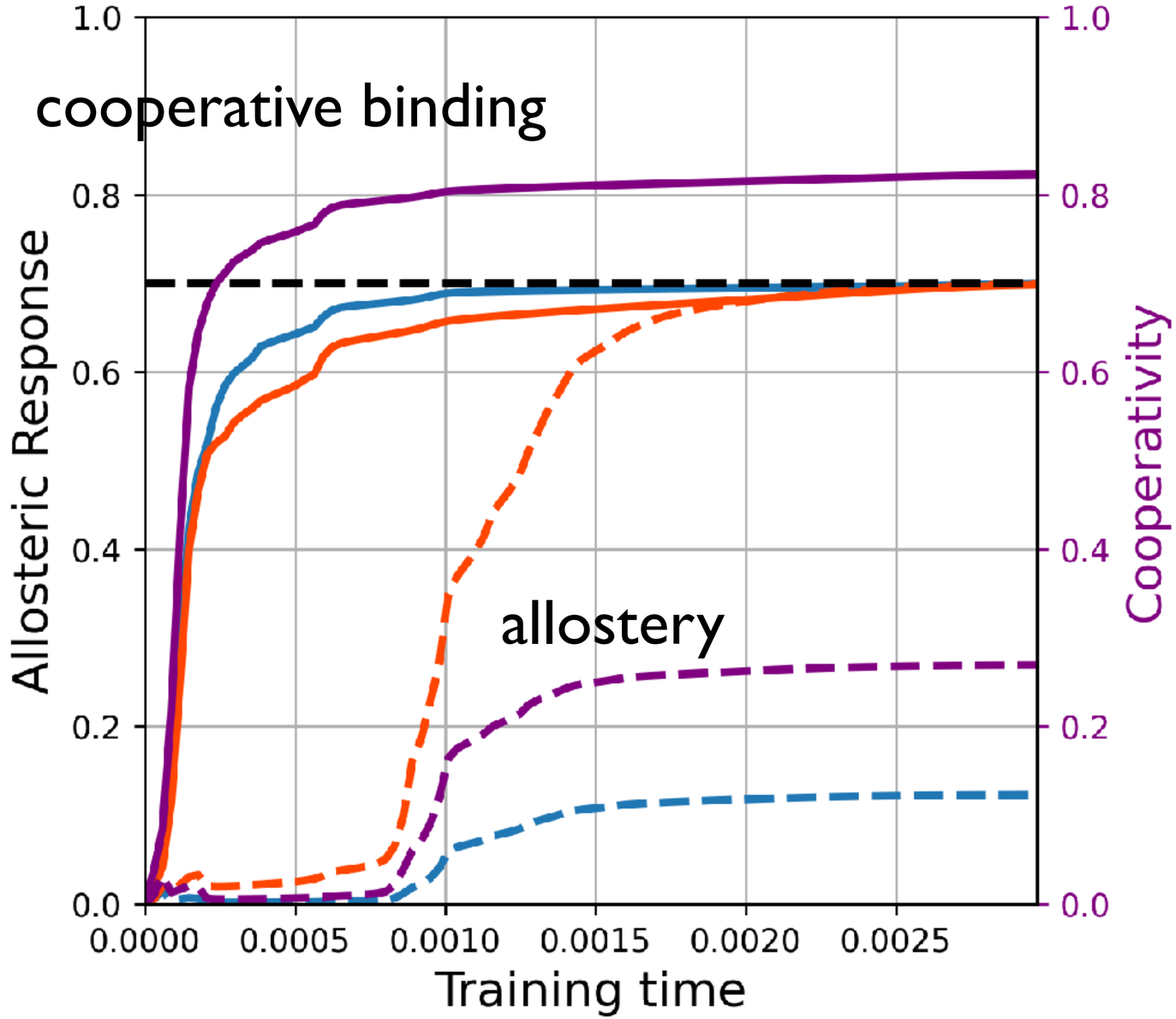
$$\dot{k}_i = -\frac{\alpha}{\eta}\frac{\partial}{\partial k_i}\left[\left(E^C - E^F\right)_{A\to B} + \left(E^C - E^F\right)_{B\to A}\right]$$

Go To SLIDES

Training for Cooperative Binding vs. Allostery



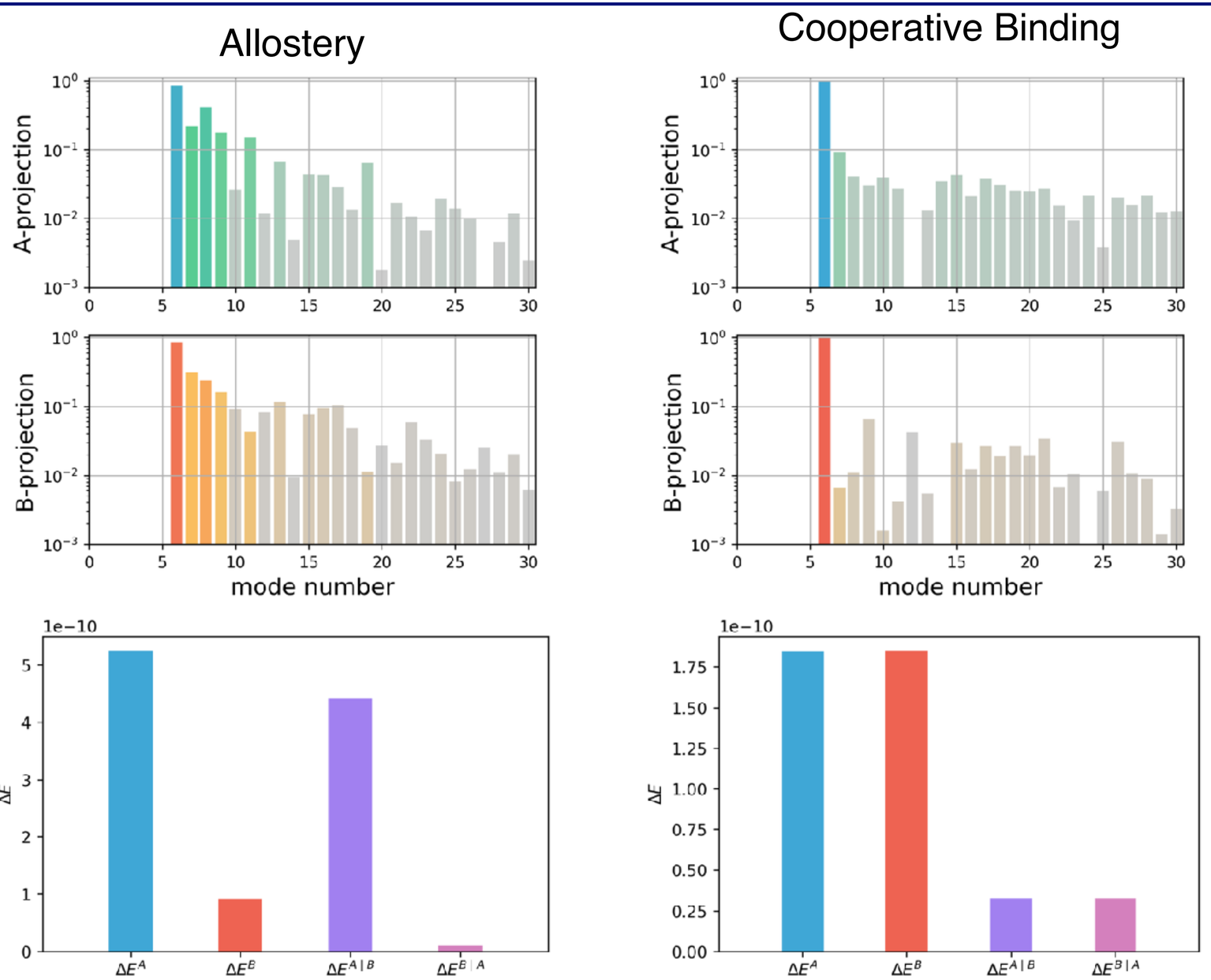
$$c = \frac{E^A + E^B - E^{A+B}}{\frac{1}{2}(E^A + E^B)}$$



Differences Between Allostery and Cooperative Binding

- Cooperative binding response projects more strongly onto lowest eigenmode of physical Hessian

- Energy required to apply strain at B (to bind B) is much higher for cooperative binding but energy to bind B once A is bound is small, as desired



Derivation of crossover T

Consider eigenmodes of physical Hessian (vibrational normal modes)

Denote these as $\vec{e}_{0,j}$ in unstrained case

j indexes mode.

First 6 modes correspond to rigid trans & rotation in 3D. exclude these soft modes.

Consider $\vec{u}$, vector of node displacements.

$$\vec{u} = \sum_{\text{modes} \to j} a_j \vec{e}_j$$

At temp T, equipartition tells us that

a_j have Gaussian distbn w/ mean 0, variance $\frac{T}{m\omega_{0,j}^2}$

So clearly we can calculate the mean & variance of the strain @ the target

$\langle \epsilon_T \rangle_0 = 0$, $\sigma^2_{\epsilon_T,0}$ (w/out input strain)

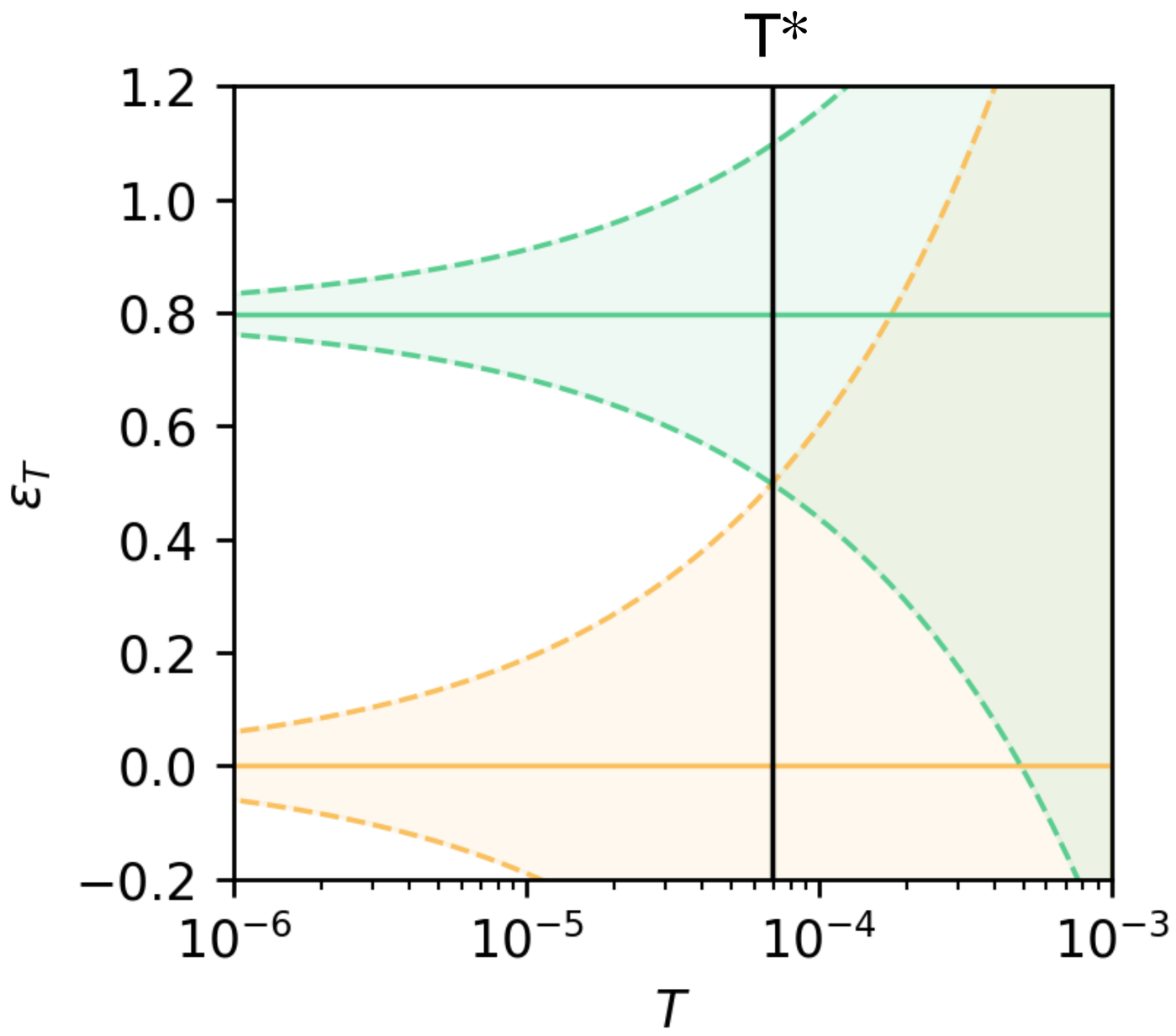
$\langle \epsilon_T \rangle$, $\sigma^2_{\epsilon_T}$ (with input strain)

in terms of eigenmodes of H.

Strains and Thermal Fluctuations

- Thermal fluctuations wipe out cooperative binding at ~T*

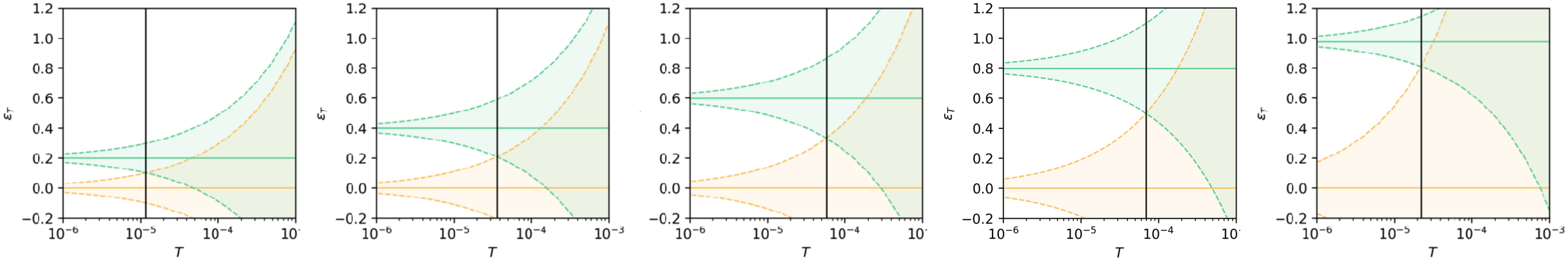
$$\langle \epsilon_T \rangle = \left(\sigma_{\epsilon_T} + \sigma^0_{\epsilon_T} \right) \sqrt{T^*}$$

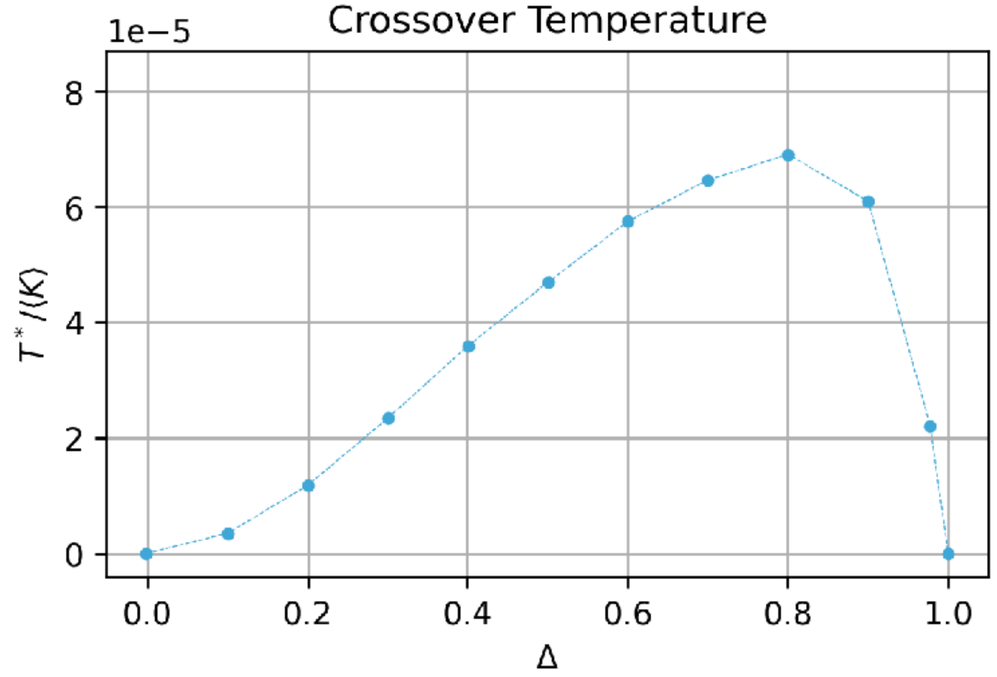


Crossover Temperature is Non-Monotonic in Response Δ

- As Δ increases, so do the fluctuations (lowest eigenfrequency drops)

$\Delta = 0.2$ $\Delta = 0.4$ $\Delta = 0.6$ $\Delta = 0.8$ $\Delta = 0.98$

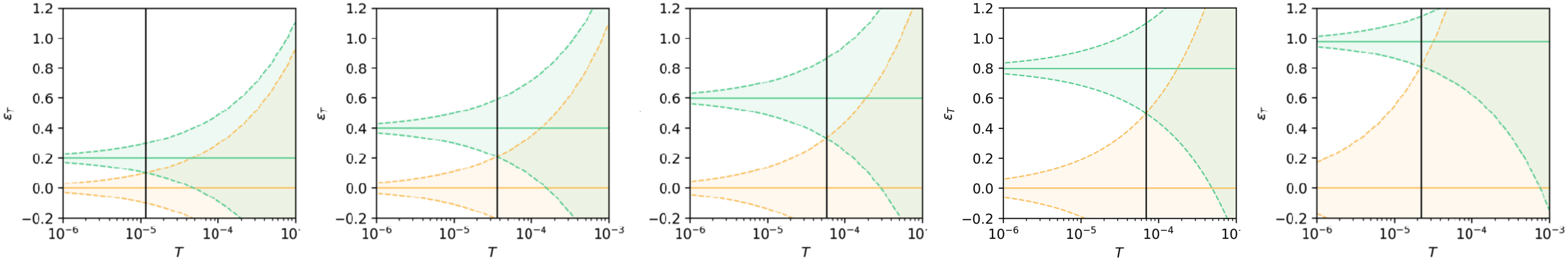


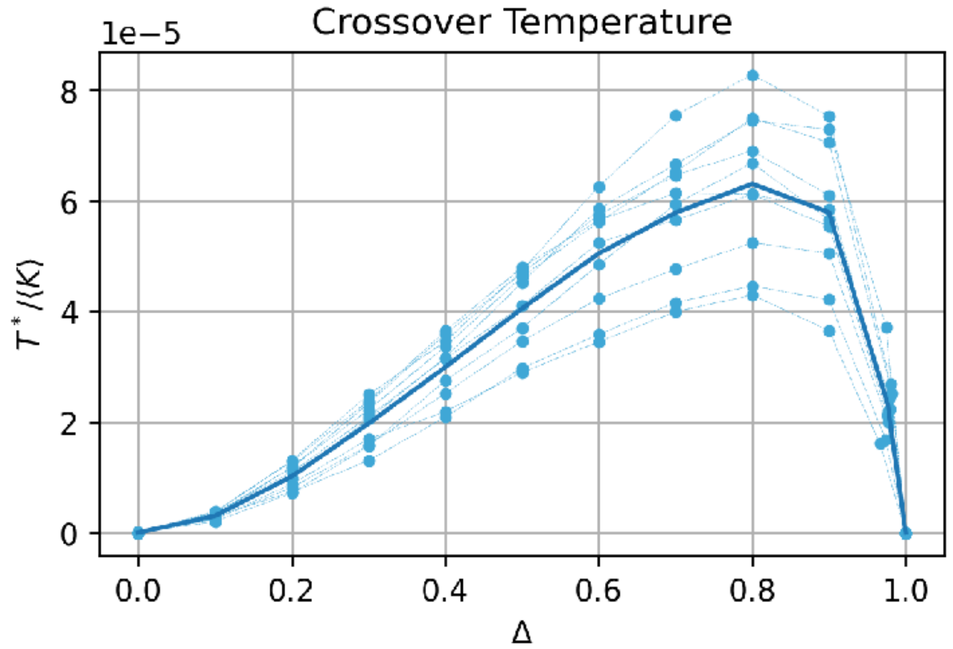


Crossover Temperature is Non-Monotonic in Response Δ

- As Δ increases, so do the fluctuations (lowest eigenfrequency drops)

$\Delta = 0.2$ $\Delta = 0.4$ $\Delta = 0.6$ $\Delta = 0.8$ $\Delta = 0.98$





How to increase T^*: energy "regularization"

original contrast:

$$C = \frac{1}{\eta}\left[\left(E^C - E^F\right)_{A\to B} + \left(E^C - E^F\right)_{B\to A}\right]$$

add a term to raise the energy of the free state:

$$C = \frac{1}{\eta}\left[\left(E^C - E^F\right)_{A\to B} + \lambda E^F_{A\to B} + \left(E^C - E^F\right)_{B\to A} + \lambda E^F_{B\to A}\right]$$

if E^F is higher then mode energies are higher.

Trade off: error is higher.

Energy Regularization Improves Thermal Resilience