Geometric deep learning for computational mechanics

Nikolaos N. Vlassis, Ran Ma, WaiChing "Steve" Sun

Theoretical and Computational Poromechanics Laboratory
The Fu Foundation School of Engineering and Applied Science
Columbia University in the city of New York

Wednesday, 18 December, 16:40 ~ 18:10, Conference Room 102 (This talk starts at 4:15pm)
Organizer: C.T. Wu, Chuin-Shan (David) Chen, Wen-Jay Lee, Nan-yow Chen, Masataka Koishi, Shaoqiang Tang and Wing-Kam Liu
Chair: Wen-Jay Lee, C.T. Wu

What is geometric deep learning?

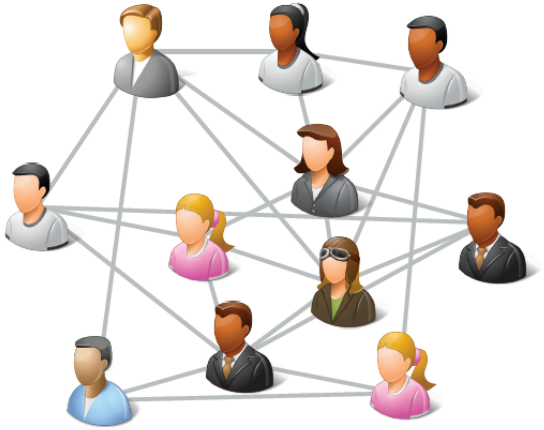
– ML on graphs and manifolds

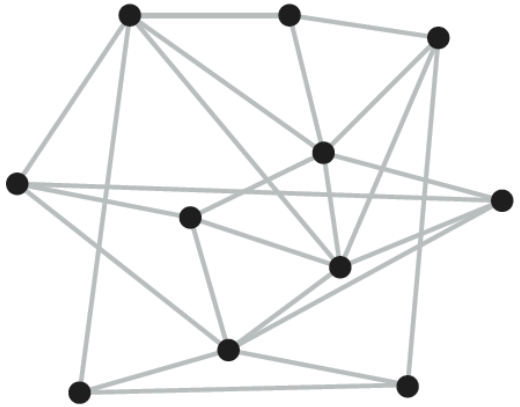
Why switch to a non-Euclidean space?

- New descriptors can be incorporated to constitutive laws

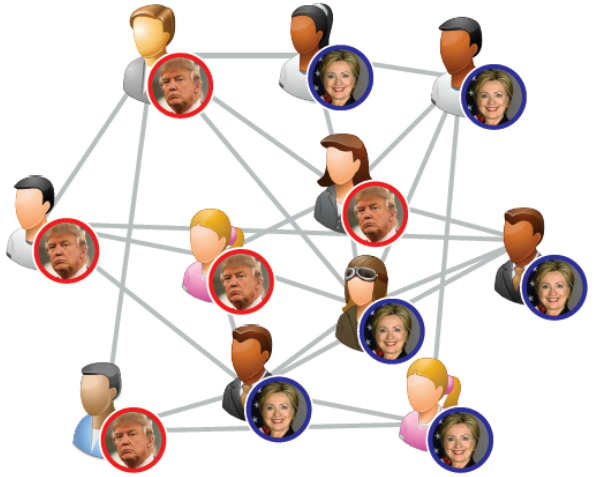
Data Representation

Consider descriptors of data as the ingredients for theory





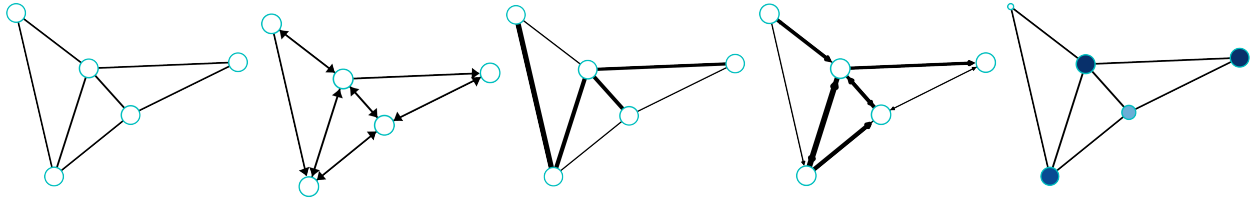
Domain structure

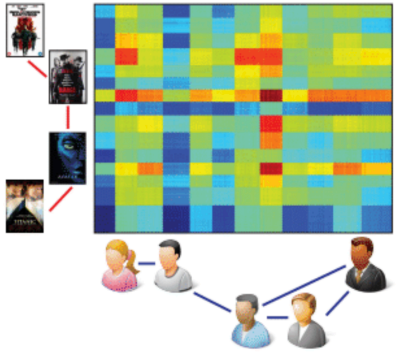


Social network

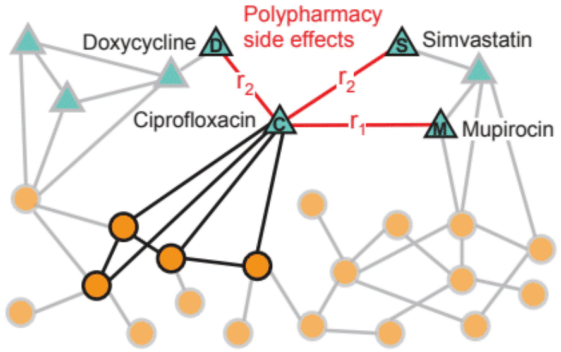
Figures from geometriclearning.com

Descriptors in a non-Euclidean space

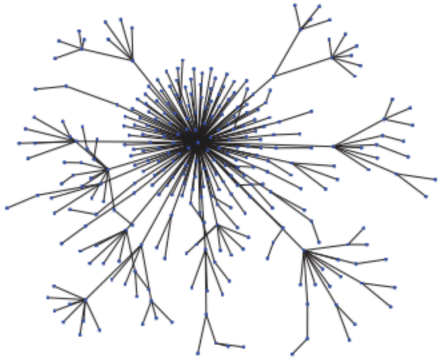




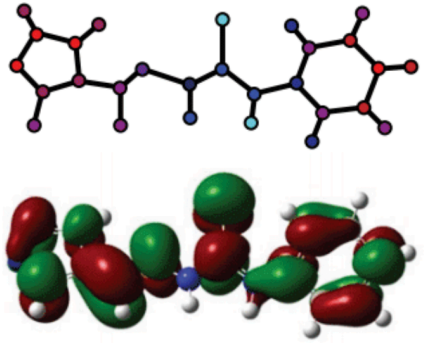
Recommender system



Drug repurposing



Fake news detection

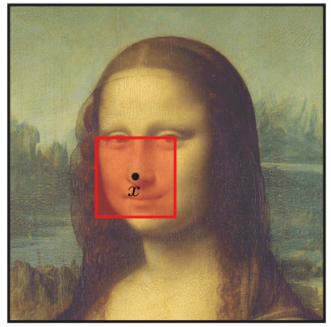


Chemistry

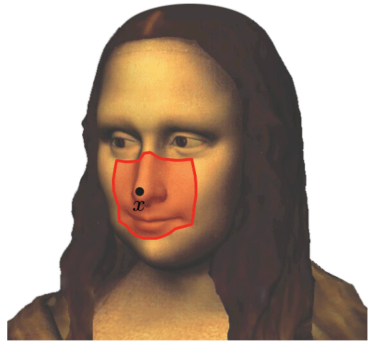
Images from geometricdeeplearning.com

Why switch from Euclidean to Non-Euclidean space:

- Data structures crafted meaningfully leveraging **domain expertise** / **interpretable** structures
- Graph structures (direction - weights) → **expressivity**
- Euclidean grid data (eg. images) → ambiguity of interpreted features
- Eliminate grid resolution dependency → **computational efficiency**



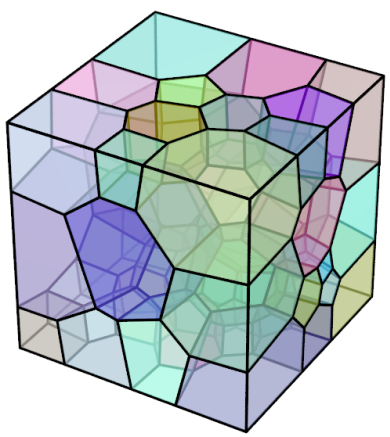
Euclidean



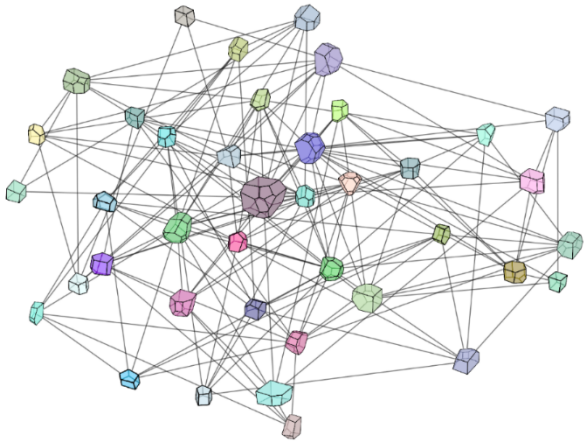
Non-Euclidean

(Image: © Egor Zakharov)

New descriptors for more accurate and more efficient predictions



Polycrystal RVE



Node-weighted undirected crystal connectivity graph

Meta-modeling games

Vlassis, Ma & Sun, under review
Wang & Sun, CMAME, 2018a, 2019a 2019b
Wang & Sun, under review

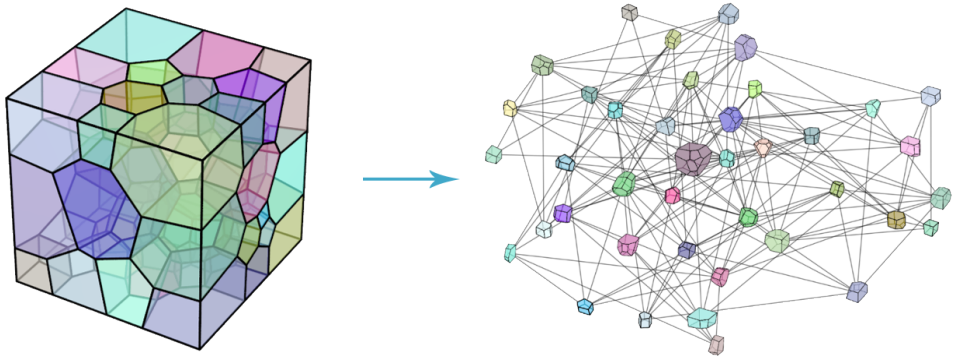
Language-game

Ludwig Wittgenstein
Philosophical Investigations (1953)

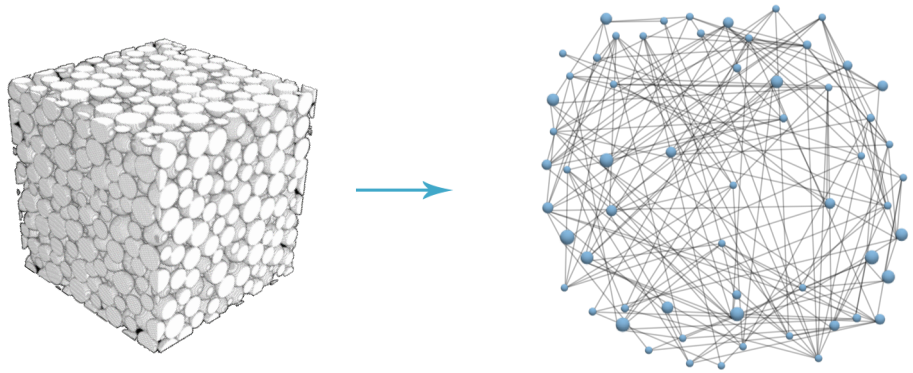
How to use graph in constitutive law generation?

Constitutive law generation with non-Euclidean descriptors

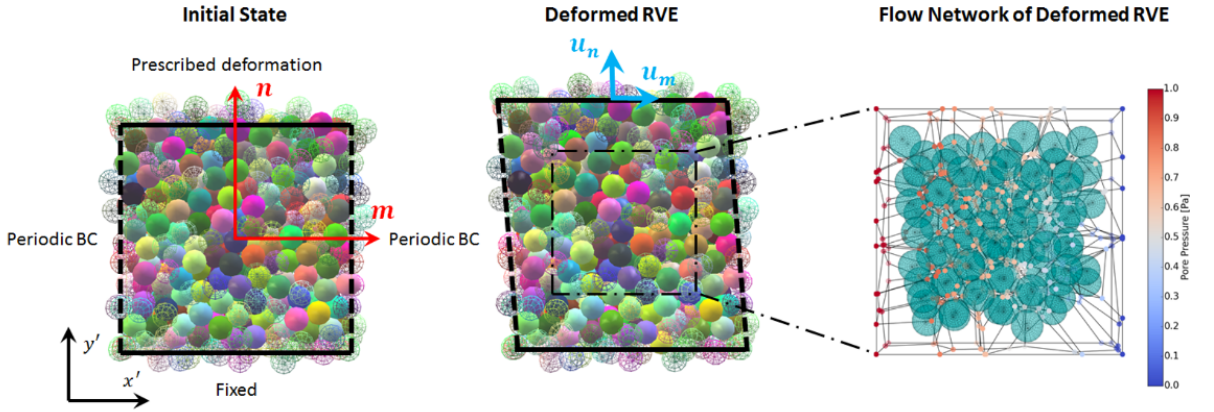
Some graph structures are readily available



The connectivity graph of the **crystals in a polycrystal formation** (this work)



The connectivity graph of the **grains in a granular assembly**.
The graph of the **pore network**



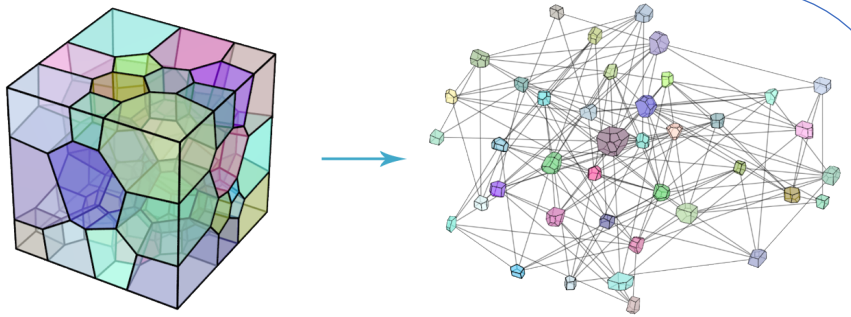
Sun, Andrade, Rudnicki, IJNME, 2011, Wang & Sun, CMAME 2016, Wang et al. IJMCE 2016, Wang & Sun, CMAME 2018

- Microstructural graph information accessible by postprocessing **imaging data**
- Combine with experimental and simulated data (strain – stress curves, flow simulations, etc)
- New insight in microstructure mechanism discovery – **learn directly from data** while **leveraging domain knowledge**

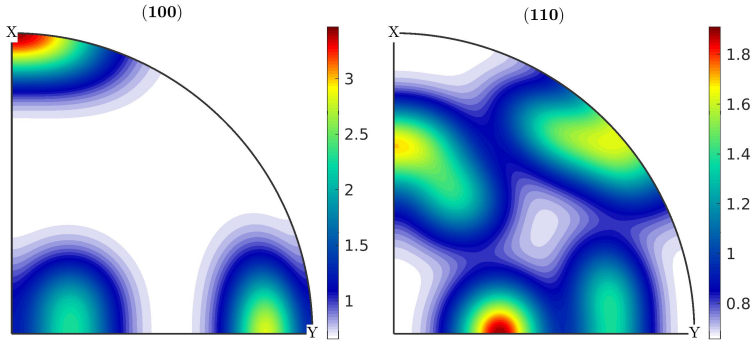
Predicting an anisotropic hyperelastic energy functional for polycrystal formations

This work: Predict $\psi = \psi(\boldsymbol{F}, \mathbb{G})$, $\boldsymbol{P} = \dfrac{\partial \psi}{\partial \boldsymbol{F}}$ with $\mathbb{G}$ the polycrystal microstructure data in non-Euclidean form (connectivity graph)

Framework:



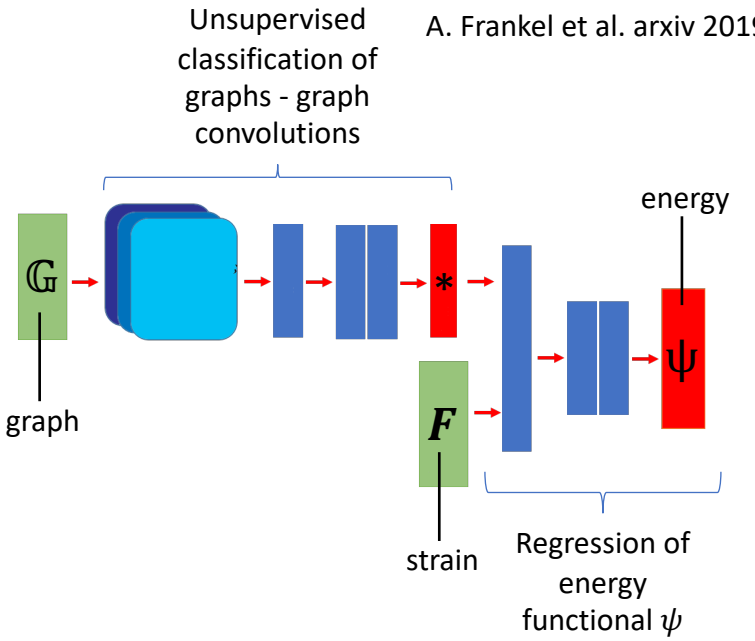
Translate polycrystal formations into undirected weighted connectivity graphs



Perform FFT simulations to generate homogenized energy and stress response database for polycrystal RVEs

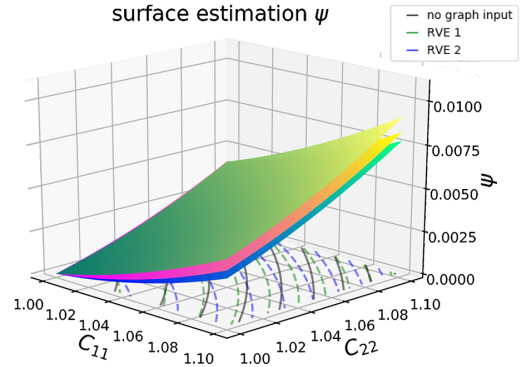
Hybrid neural network architecture

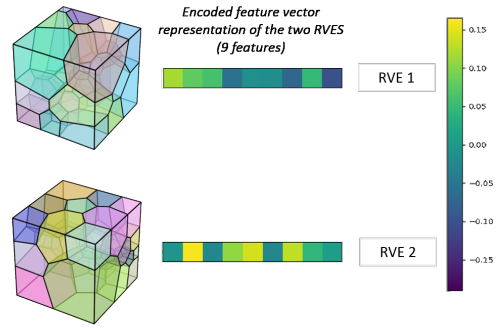
A. Frankel et al. arxiv 2019



Geometric deep learning for computational mechanics Part I: Anisotropic Hyperelasticity,

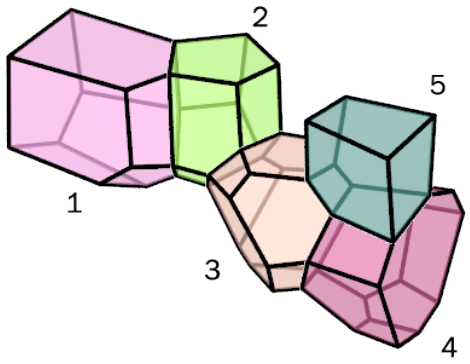
Vlassis et al, under review



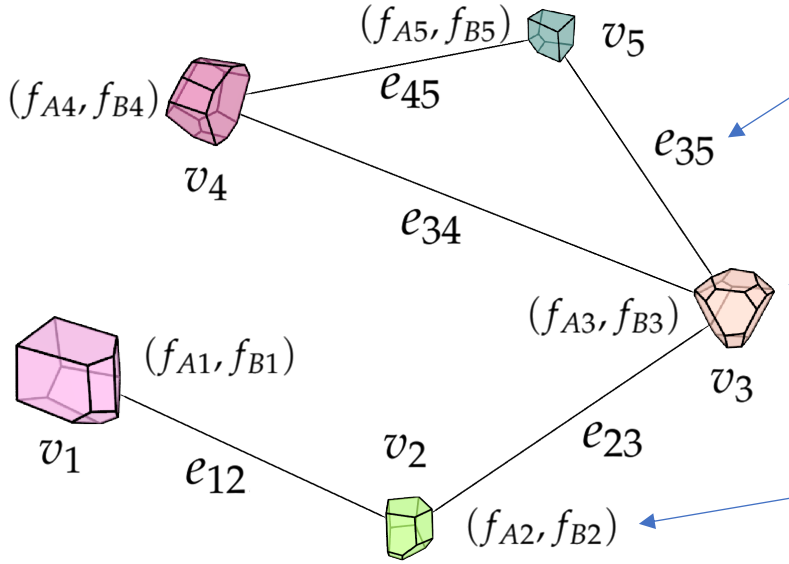


* Graphs encoded in feature vectors

Translating polycrystal formations into undirected weighted graphs



Polycrystal formation



Undirected weighted (connectivity) graph

- Two crystals that are in contact are connected by an **edge**
- Every crystal is a **node**
- Every node has **weights** – **features**

[volume, orientation, number of neighbors, number of edges, number of surfaces, etc.]

A **graph** is a two-tuple $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where $\mathbb{V} = \{v_1, \ldots, v_N\}$ is a non-empty **vertex set** (also referred to as nodes) and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is an **edge set**. To define a graph, there exists a relation that associates each edge with two vertices.

In the above example:

$$\mathbb{V} = \{v_1, v_2, v_3, v_4, v_5\}$$

$$\mathbb{E} = \{e_{12}, e_{23}, e_{34}, e_{35}, e_{45}\}$$

Vertex and edge sets

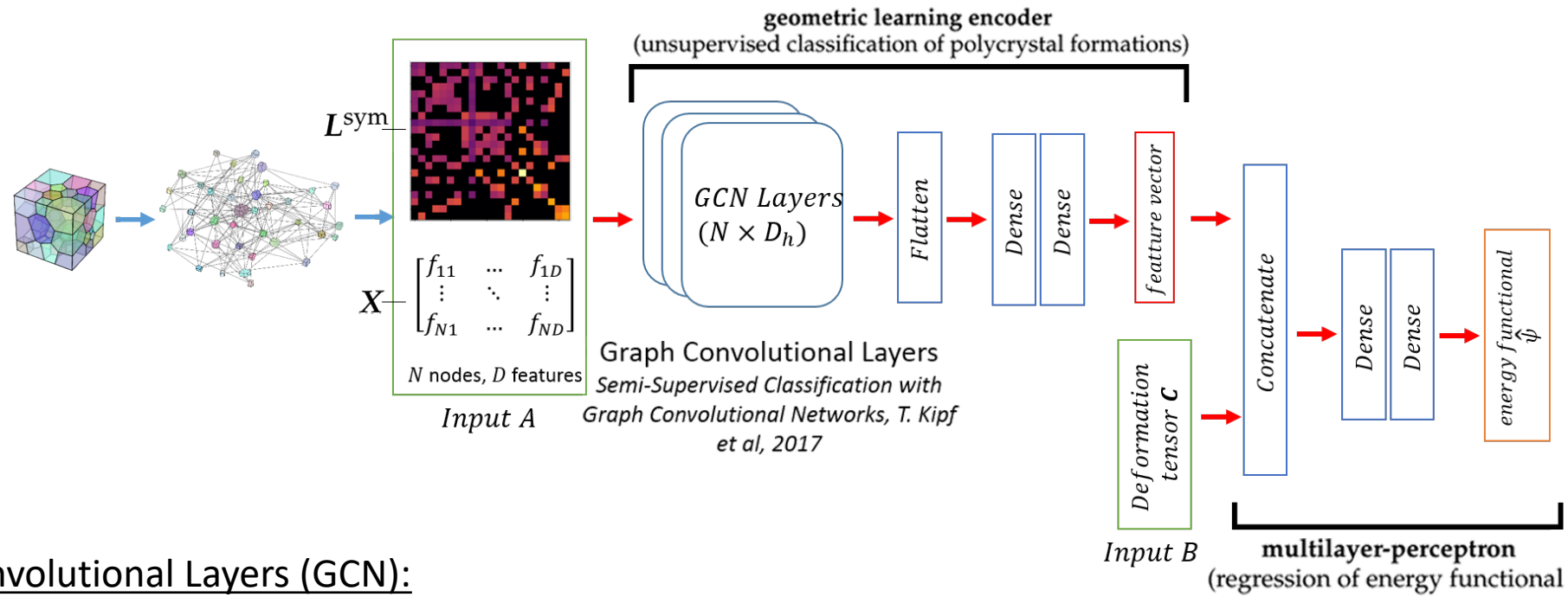
$$\boldsymbol{L}^{\text{sym}} = \begin{bmatrix} 1 & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ -\frac{\sqrt{2}}{2} & 1 & -\frac{\sqrt{6}}{6} & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{6} & 1 & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{6} \\ 0 & 0 & -\frac{\sqrt{6}}{6} & 1 & -\frac{1}{2} \\ 0 & 0 & -\frac{\sqrt{6}}{6} & -\frac{1}{2} & 1 \end{bmatrix}$$

Symmetric normalized Laplacian matrix

$$\boldsymbol{X} = \begin{bmatrix} f_{A1} & f_{B1} \\ f_{A2} & f_{B2} \\ f_{A3} & f_{B3} \\ f_{A4} & f_{B4} \\ f_{A5} & f_{B5} \end{bmatrix}$$

Feature matrix

Hybrid neural network architecture

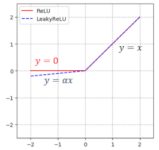


Graph Convolutional Layers (GCN):

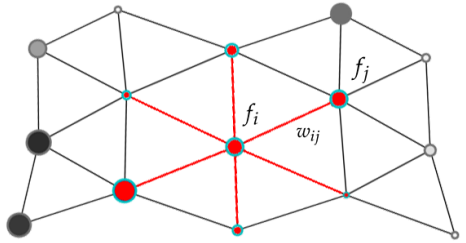
output — features — bias

$$\boldsymbol{h}_{\mathrm{GCN}}^{(l+1)} = \sigma\left(\boldsymbol{L}^{\mathrm{sym}} \boldsymbol{h}^{(l)} \boldsymbol{W}^{(l)} + \boldsymbol{b}^{(l)}\right)$$

non-linear activation — normalized graph Laplacian — weights

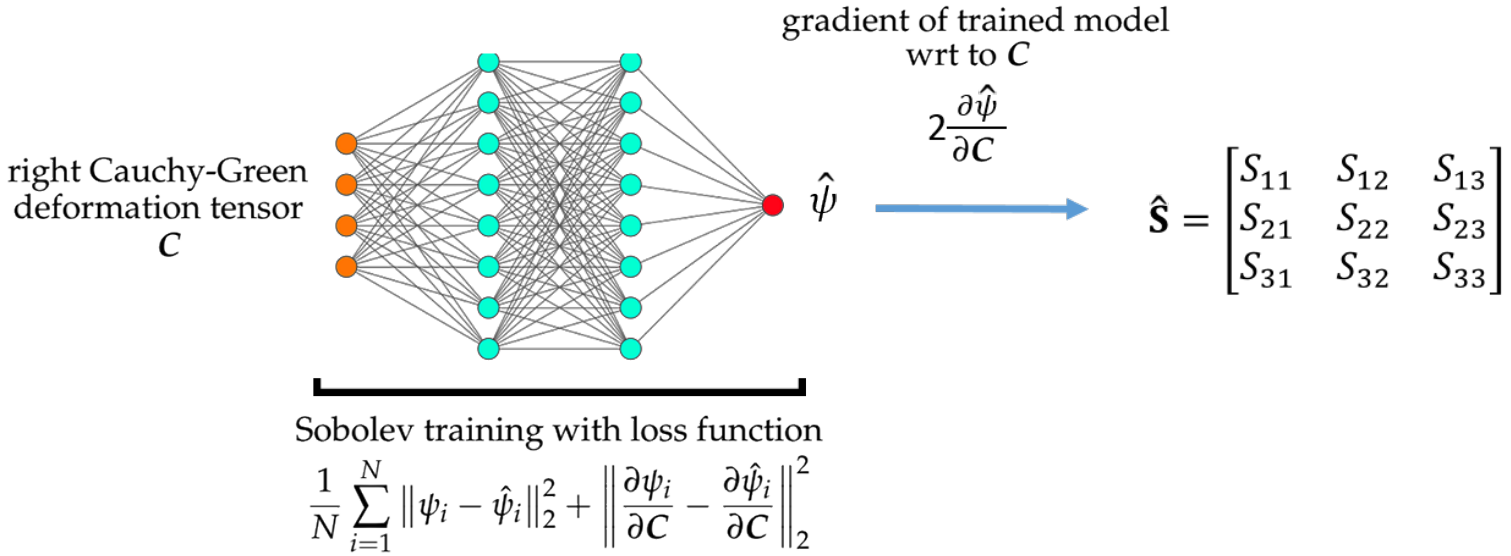


Graph convolution of the spectrally decomposed graph Laplacian approximated by stacking first order **Chebyshev polynomials filters**



The graph Laplacian operator describes the difference between a value f at a node and its local average – spatial filter of the node feature relations in the graph

Robust regression of anisotropic energy functional



Training schematic:

- Constrain both energy and stress predictions during optimization
- The derivative of the trained neural network wrt the input strain is a valid stress tensor

---

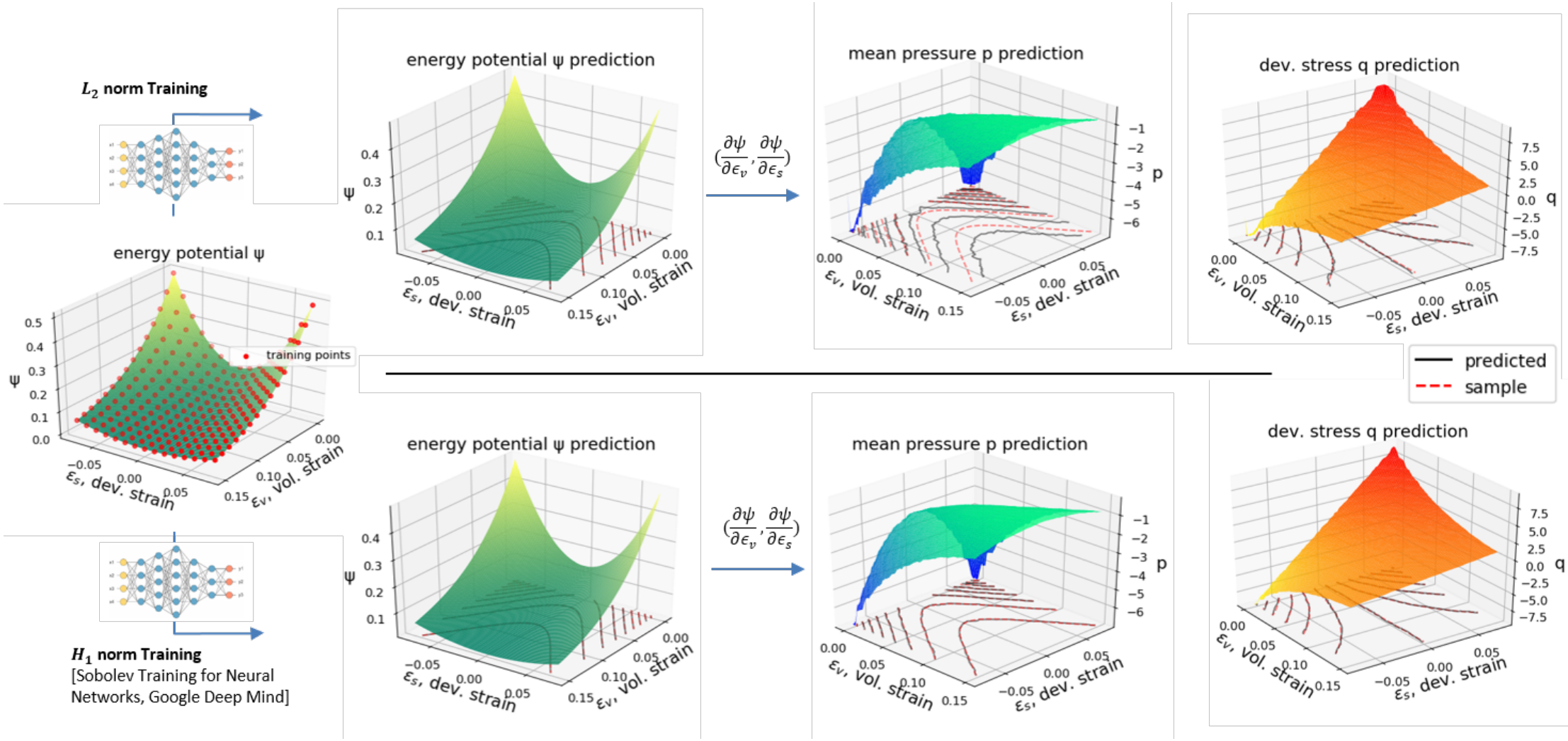
Higher order training constraining both the function's values and derivatives [Sobolev training, Google Deep Mind]:

L_2 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \underbrace{\|\psi_i - \hat{\psi}_i\|_2^2}_{\text{Constrain energy}} \right)$$

H_1 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \underbrace{\|\psi_i - \hat{\psi}_i\|_2^2 + \left\|\frac{\partial \psi_i}{\partial \boldsymbol{C}} - \frac{\partial \hat{\psi}_i}{\partial \boldsymbol{C}}\right\|_2^2}_{\text{Constrain energy and stress}} \right)$$

Sobolev training: requires less data samples, smoothened approximated functions, accurate approximated derivatives

L_2 norm - H_1 norm Training Comparison



Database generation with FFT simulations

- **Fast Fourier Transform method** (FFT) for mesoscale homogenization of the polycrystal formation response
- 3D **periodic** domain
- Locally a **generalized Fung elasticity** model is used, with strain density function W:

$$W = \frac{1}{2}c\left[\exp\left(Q\right) - 1\right], \quad Q = \frac{1}{2}\boldsymbol{E} : \boldsymbol{a} : \boldsymbol{E}$$

$$Q = c^{-1}\sum_{a=1}^{3}\left[2\mu_a \boldsymbol{A}_a^0 : \boldsymbol{E}^2 + \sum_{b=1}^{3}\lambda_{ab}\left(\boldsymbol{A}_a^0 : \boldsymbol{E}\right)\left(\boldsymbol{A}_b^0 : \boldsymbol{E}\right)\right], \quad \boldsymbol{A}_a^0 = \boldsymbol{a}_a^0 \otimes \boldsymbol{a}_a^0$$

where c is a scalar material constant, $\boldsymbol{E}$ is the Green strain tensor, μ_a and λ_{ab} are anisotropic Lame constants, and $\boldsymbol{a}_a^0$ is the unit vector of the orthotropic plane normal.

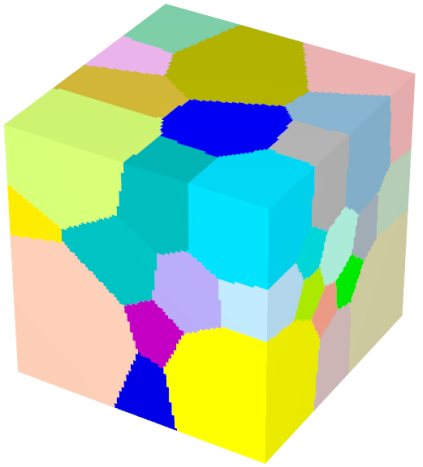
- **150 randomly generated RVEs** of 40 to 50 crystals each

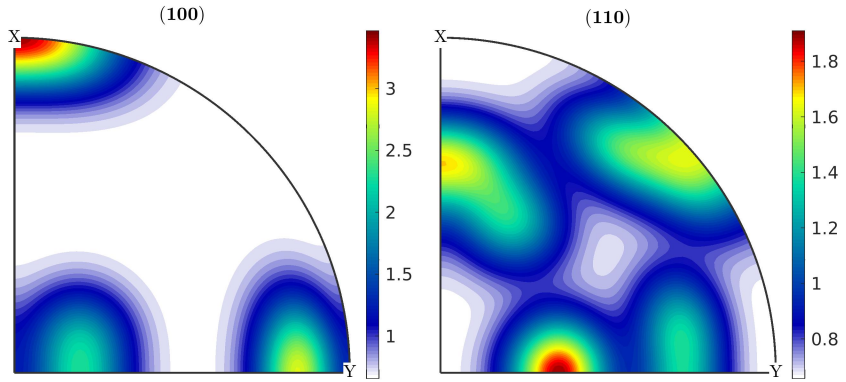
The **orientation distribution function (ODF)** is randomly generated by combining uniform orientation and unimodal orientation:

$$f(\boldsymbol{x};\boldsymbol{g}) = w + (1-w)\psi(\boldsymbol{x},\boldsymbol{g}), \qquad \boldsymbol{x} \in SO(3).$$

- Simulations are performed for each RVE with **200 average strains**.

Note: constitutive relation is **hyperelastic** ⇢ simulation result is **path independent**



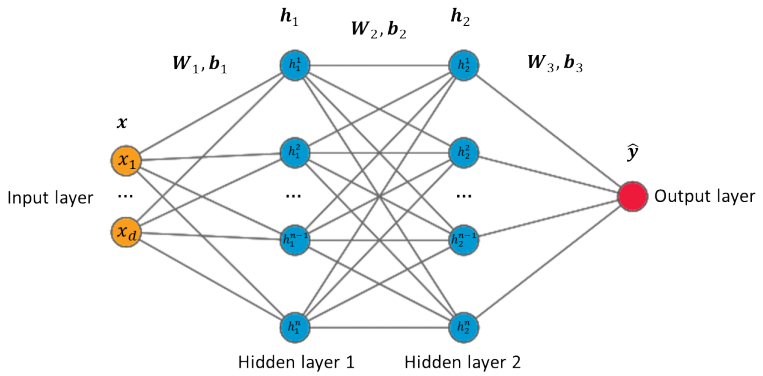


Pole figure plot of initial orientation distribution function (ODF) combining uniform and unimodal ODF. The Euler angles of the unimodal direction are $(207.1^o, 17.8^o, 159.0^o)$ in Bunge notation, and the half width of the unimodal ODF is 10^o

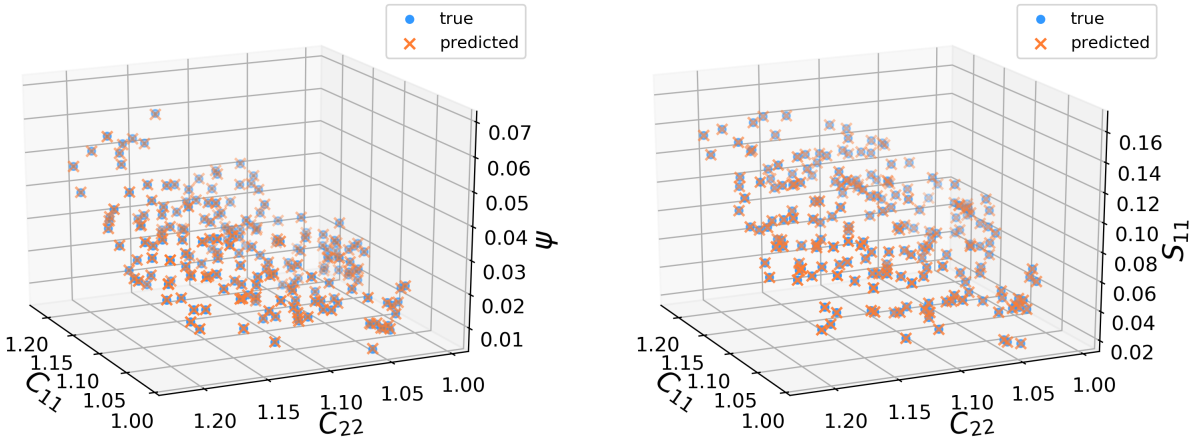
R. Ma, W.C. Sun, FFT-based solver for higher-order and multi-phase-field fracture models applied to strongly anisotropic brittle materials and poly-crystals, Computer Methods in Applied Mechanics and Engineering, 2019.

Results?

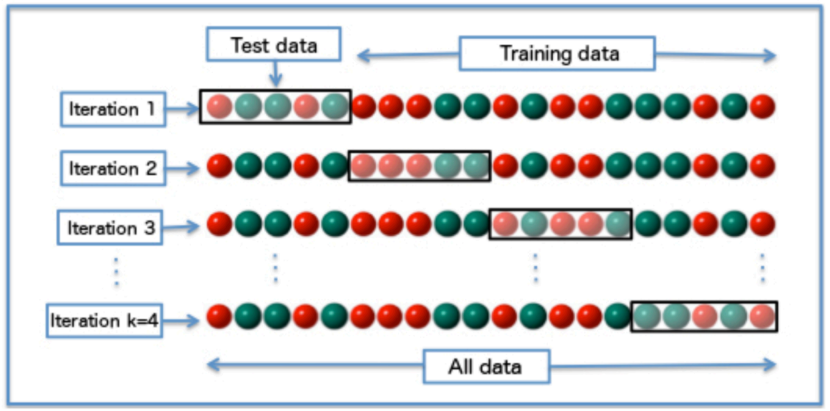
Numerical experiment 1: The material response of a single polycrystal formation



A simple multilayer-perceptron (MLP) with Sobolev training can capture the behavior of a single polycrystal formation

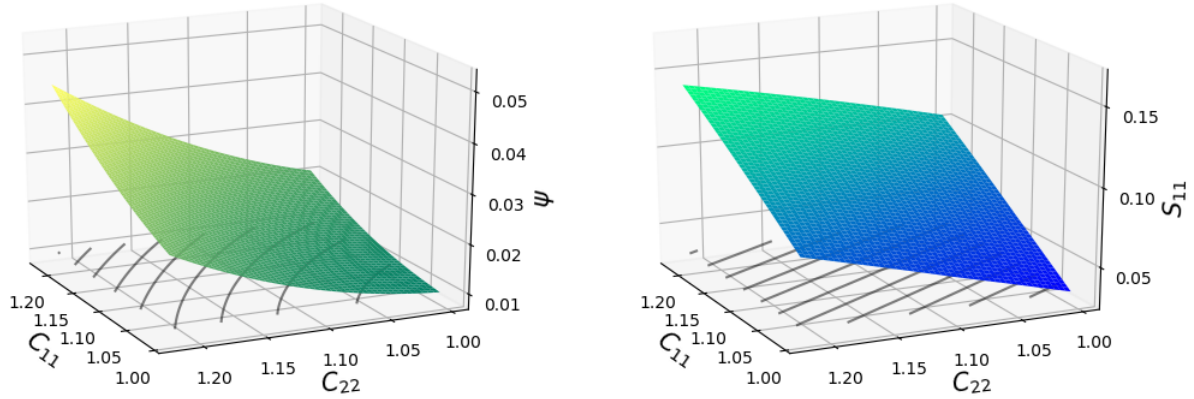


K-fold blind prediction results for 200 randomly generated deformation gradients for a single polycrystal formation



K-fold Cross-validation algorithm

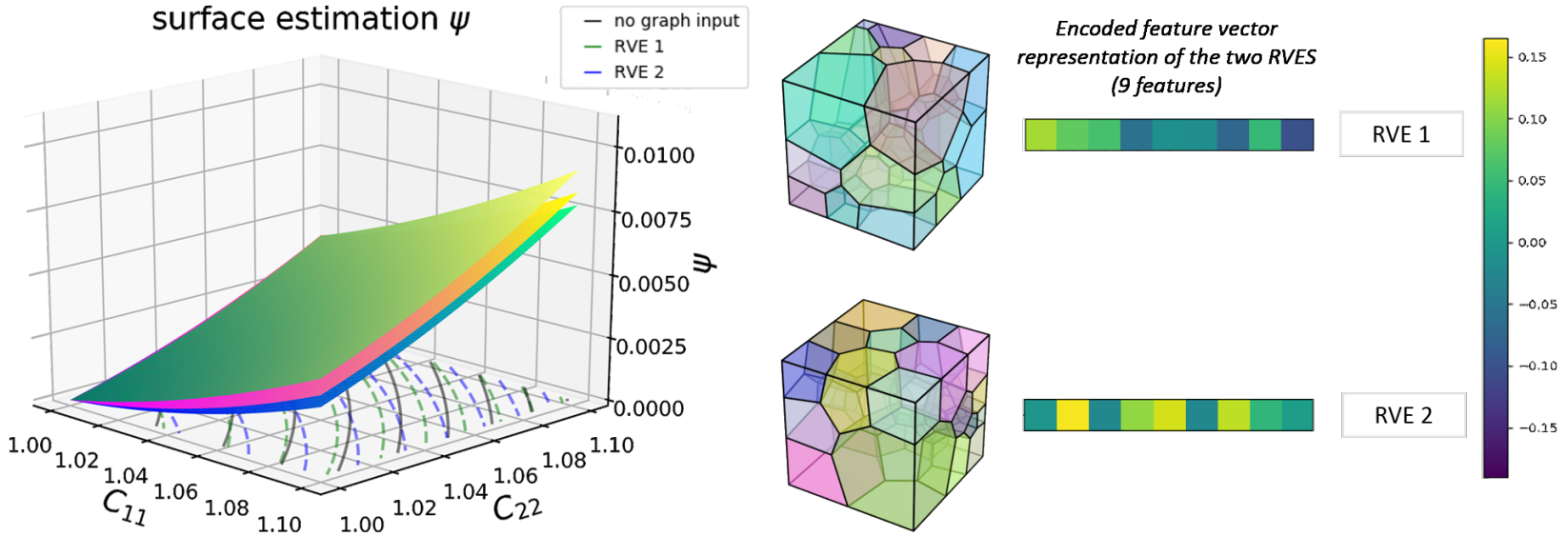
Figure from Wikipedia



Energy and stress response surface estimation for a single polycrystal formation

Numerical experiment 2: The anisotropic behavior is captured with the graph input

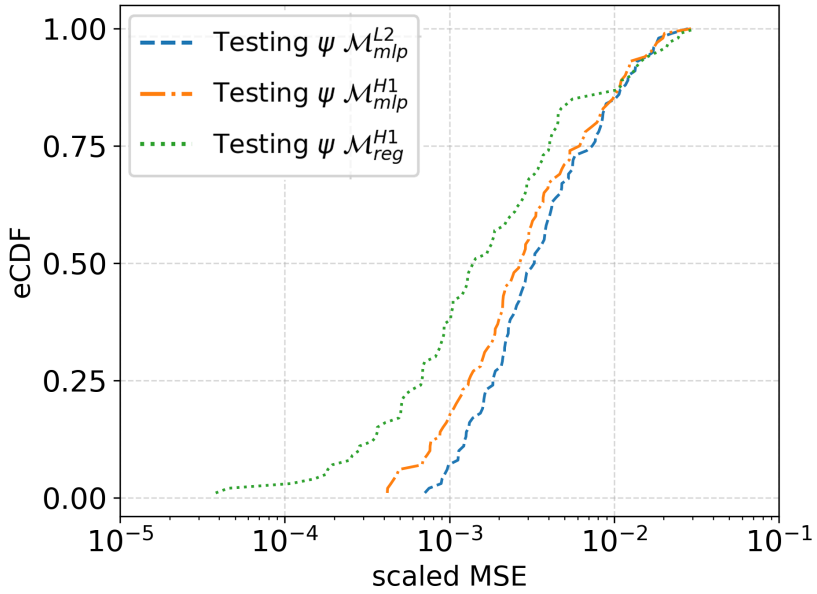
A simple multilayer-perceptron (MLP) **without the graph input cannot capture** the behavior of multiple polycrystal formations!

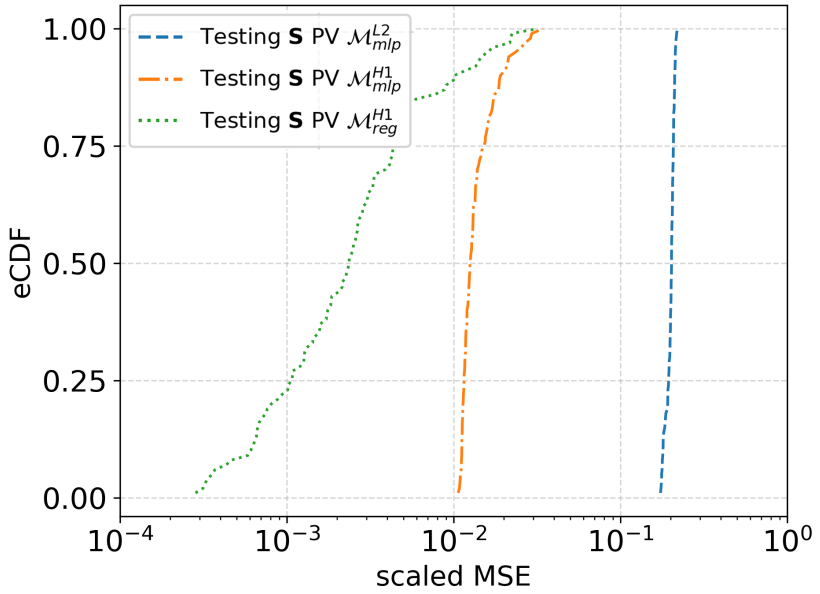


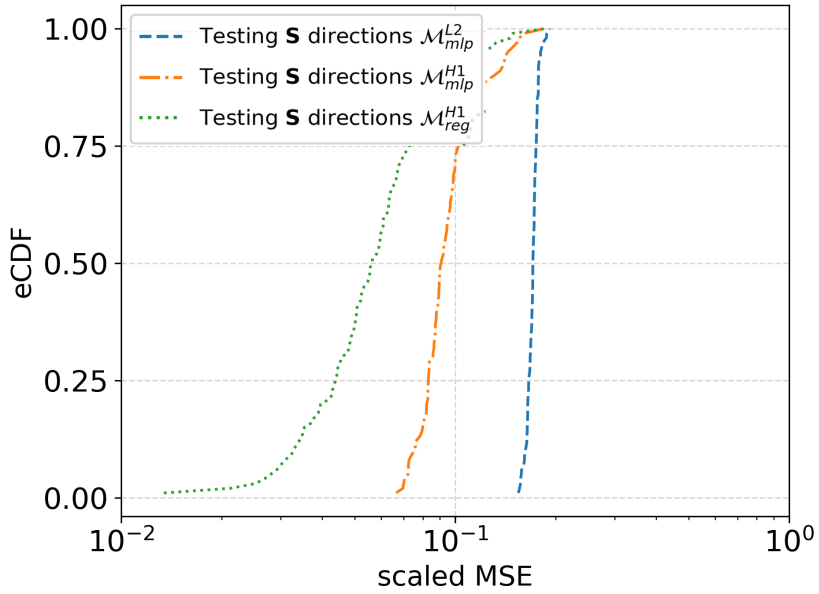
Energy response surface estimation for 2 RVEs with and without the graph input

- Graph input ⇢ distinguish between polycrystal formation behaviors
- The encoded vectors are inferred **anisotropy descriptors**
- **Without graph,** the prediction of the network is a mere projection of the true values (**mapping problem**: for 1 strain input – 2 output energy values)

K-fold Cross-validation results







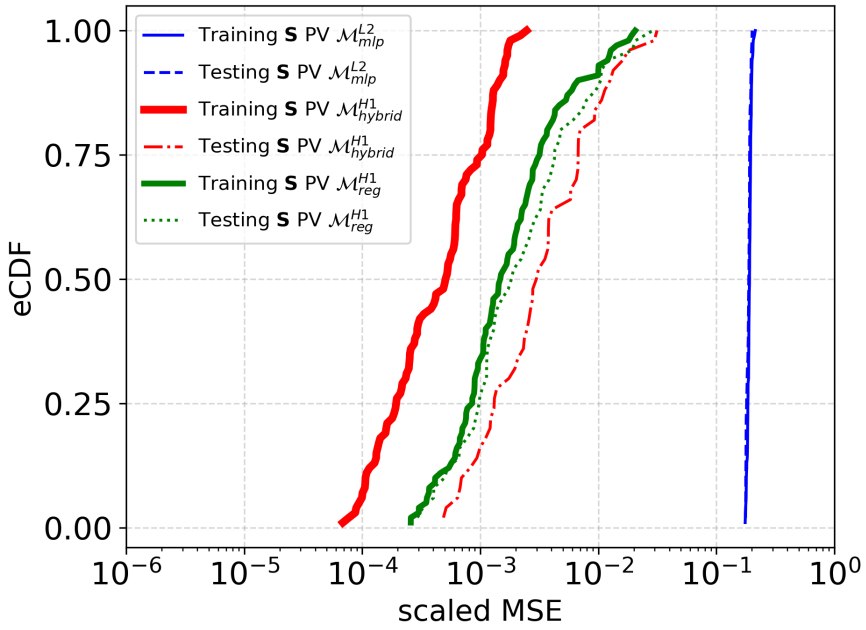
- K-fold cross validation blind forward prediction results for 100 different polycrystal formations
- Superior blind prediction results for hybrid architecture (compared to L_2 and H_1 constrained multilayer-perceptrons)

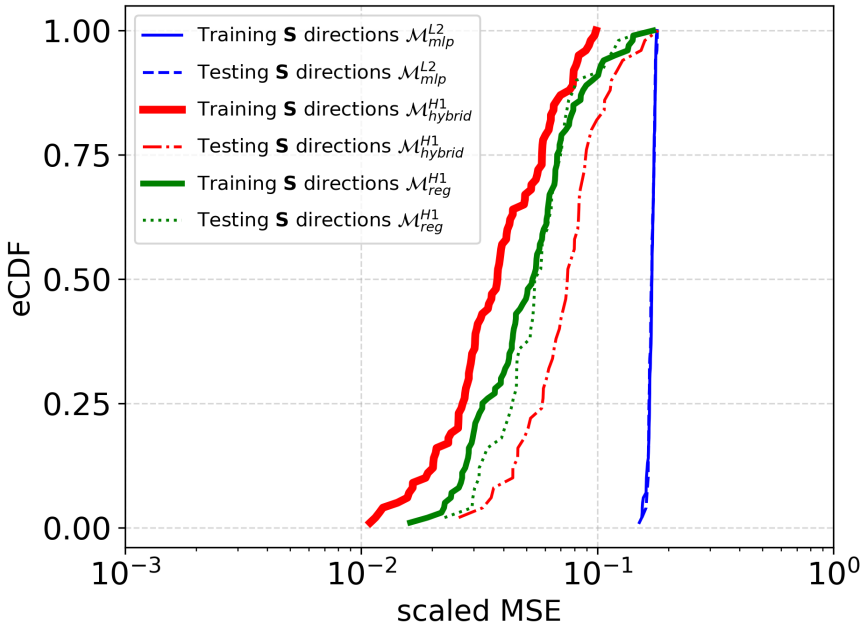
Model	Description
$\mathcal{M}^{L2}_{mlp}$	Multilayer perceptron feed-forward architecture. Loss function used is the L_2 norm
$\mathcal{M}^{H1}_{mlp}$	Multilayer perceptron feed-forward architecture. Loss function used is the H_1 norm
$\mathcal{M}^{H1}_{hybrid}$	Hybrid architecture . Loss function used is the H_1 norm
$\mathcal{M}^{H1}_{reg}$	Hybrid architecture . Loss function used is the H_1 norm The geometric learning branch of the network is regularized against overfitting.

L_2 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left\| \psi_i - \hat{\psi}_i \right\|_2^2 \right)$$

H_1 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left\| \psi_i - \hat{\psi}_i \right\|_2^2 + \left\| \frac{\partial \psi_i}{\partial \boldsymbol{C}} - \frac{\partial \hat{\psi}_i}{\partial \boldsymbol{C}} \right\|_2^2 \right)$$

Regularization





- Graph-encoding branch of the network is prone to overfitting
- Regularization needed (dropout, L_2 regularization)

Model	Description
$\mathcal{M}^{L2}_{mlp}$	Multilayer perceptron feed-forward architecture. Loss function used is the L_2 norm
$\mathcal{M}^{H1}_{mlp}$	Multilayer perceptron feed-forward architecture. Loss function used is the H_1 norm
$\mathcal{M}^{H1}_{hybrid}$	Hybrid architecture . Loss function used is the H_1 norm
$\mathcal{M}^{H1}_{reg}$	Hybrid architecture . Loss function used is the H_1 norm The geometric learning branch of the network is regularized against overfitting.

L_2 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left\| \psi_i - \hat{\psi}_i \right\|_2^2 \right)$$

H_1 norm: $$\boldsymbol{W}', \boldsymbol{b}' = \underset{\boldsymbol{W},\boldsymbol{b}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} \left\| \psi_i - \hat{\psi}_i \right\|_2^2 + \left\| \frac{\partial \psi_i}{\partial \boldsymbol{C}} - \frac{\partial \hat{\psi}_i}{\partial \boldsymbol{C}} \right\|_2^2 \right)$$

Numerical experiment 3: Objectivity, anisotropy and convexity of the material model

Objectivity check

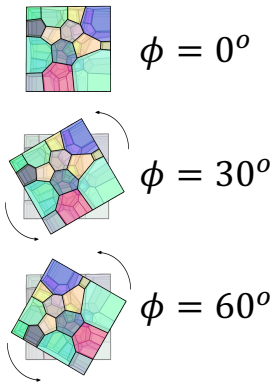
Function must be left rotationally invariant:

$\psi(\boldsymbol{F}) = \psi(\boldsymbol{F}^{+}), \qquad \boldsymbol{F}^{+} = \boldsymbol{Q}\boldsymbol{F}$

The model is trained as function of the right Cauchy deformation tensor $\boldsymbol{C}$, thus, the condition is **automatically met**:

$\hat{\psi} = \hat{\psi}(\boldsymbol{C}, \mathbb{G}) = \hat{\psi}(\boldsymbol{C}^{+}, \mathbb{G})$ with $\boldsymbol{C}^{+} = (\boldsymbol{F}^{+})^{T}\boldsymbol{F}^{+} = \boldsymbol{F}^{T}\boldsymbol{Q}^{T}\boldsymbol{Q}\boldsymbol{F} = \boldsymbol{F}^{T}\boldsymbol{F} \equiv \boldsymbol{C}$

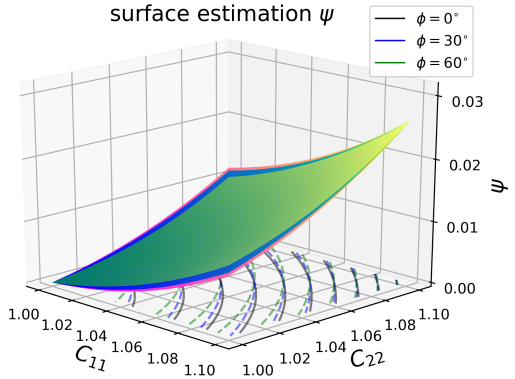
Anisotropic responses obtained from ML trained energy functional

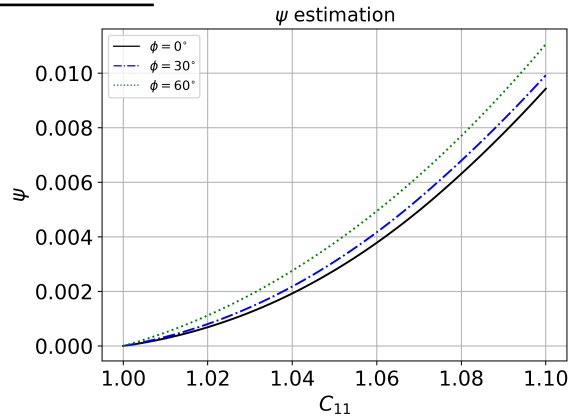


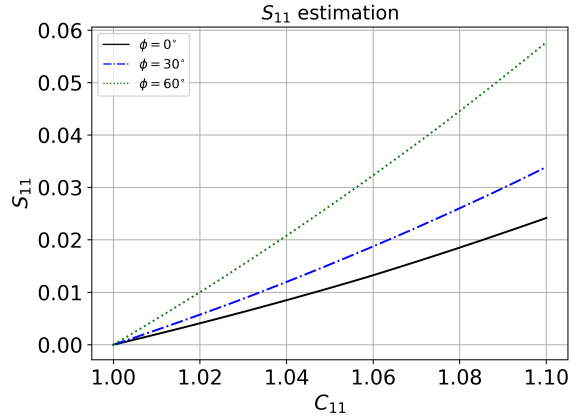
$\phi = 0^{o}$

$\phi = 30^{o}$

$\phi = 60^{o}$



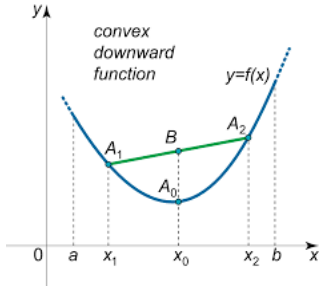


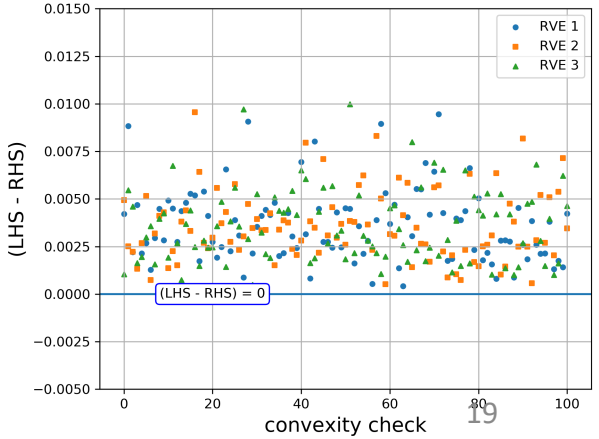


Convexity check

The **first order Taylor expansion** at any point of the domain is **a global under-estimator** of a convex function *f :*

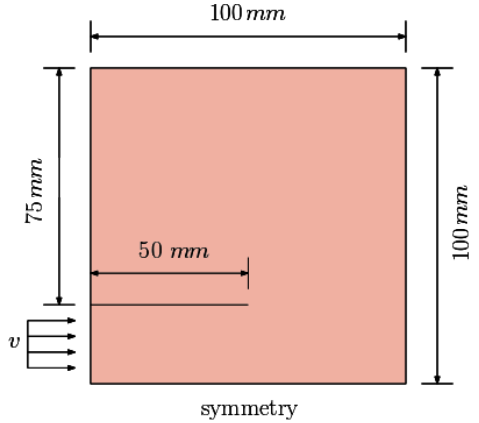
$\hat{\psi}(\boldsymbol{C}_{\alpha}, \mathbb{G}_{k}) \geq \hat{\psi}(\boldsymbol{C}_{\beta}, \mathbb{G}_{k}) + \frac{\partial \hat{\psi}}{\partial \boldsymbol{C}}(\boldsymbol{C}_{\beta}, \mathbb{G}_{k}) : (\boldsymbol{C}_{\alpha} - \boldsymbol{C}_{\beta}), \quad \text{for all} \quad \boldsymbol{C}_{\alpha}, \boldsymbol{C}_{\beta} \in D \text{ and } k \in [1, \ldots, N_{RVE}]$

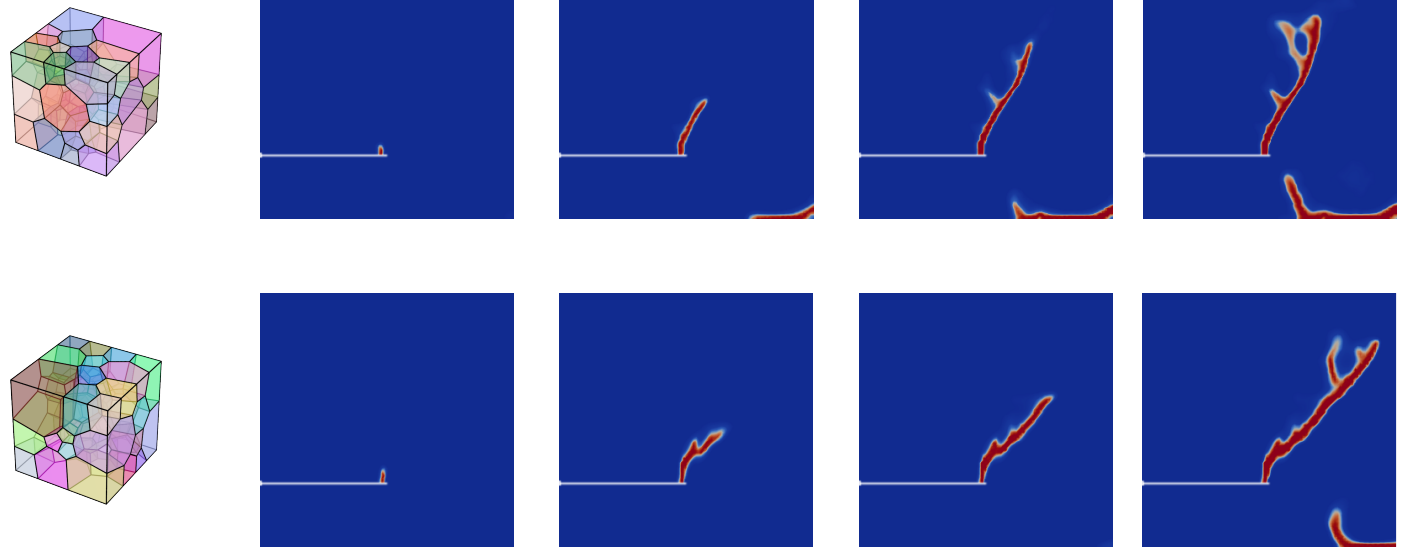




Parametric study: Anisotropic response of polycrystal formations in phase-field fracture

- Anisotropic model behavior examined with phase-field fracture numerical experiments





Crack patterns at 30 μs, 50 μs, 65 μs, 85 μs for the dynamic shear loading experiment with an impact velocity of v = 33.0 m/s for a model with two different polycrystal formations - graph inputs.

- Crack propagation under a dynamic shear load (Kalthoff and Winkler, 1988)

- The balance of linear momentum and phase-field equations are solved with the finite element method in large deformations.
- The homogenized response of the polycrystal formation is calculated at the material points by the trained hybrid neural network.

Isochoric-volumetric split:

$$\boldsymbol{F} = \boldsymbol{F}_{\text{iso}}\boldsymbol{F}_{\text{vol}} = \boldsymbol{F}_{\text{vol}}\boldsymbol{F}_{\text{iso}}, \quad J = \det(\boldsymbol{F})$$

"tensile-compressive split":

$$\psi^{+} = \begin{cases} \psi(\boldsymbol{F},\mathbb{G}) & J \geq 1 \\ \psi(\boldsymbol{F},\mathbb{G}) - \psi(\boldsymbol{F}_{\text{vol}},\mathbb{G}) & J < 1, \end{cases}$$

$$\psi^{-} = \begin{cases} 0 & J \geq 1 \\ \psi(\boldsymbol{F}_{\text{vol}},\mathbb{G}) & J < 1. \end{cases}$$

History variable:

$$H(\boldsymbol{F}_{t_n},\mathbb{G}) = \max_{t_n \leq t} \psi^{+}(\boldsymbol{F}_{t_n},\mathbb{G})$$

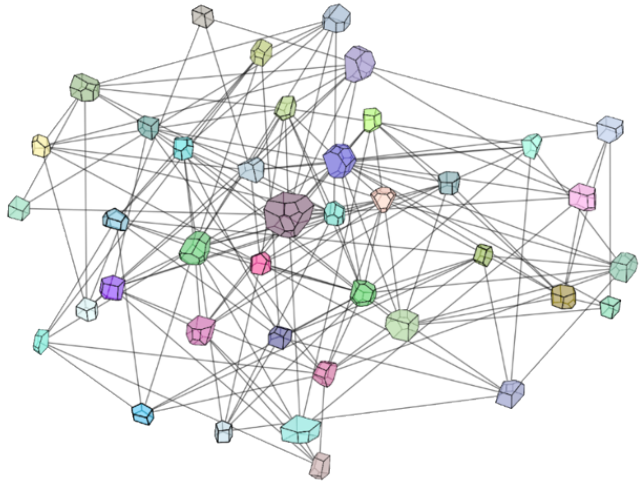
Stress response:

$$\boldsymbol{P}(\boldsymbol{F},d,\mathbb{G}) = 2\boldsymbol{F}\left[g(d)\frac{\partial\hat{\psi}^{+}(\boldsymbol{C},\mathbb{G})}{\partial\boldsymbol{C}} + \frac{\partial\hat{\psi}^{-}(\boldsymbol{C},\mathbb{G})}{\partial\boldsymbol{C}}\right]$$

Conclusion – Future work

Conclusion:

- First attempt to incorporate **non-Euclidean descriptors** in constitutive generation
- **Microstructures** translated into **undirected weighted graphs**
- Introduced **hybrid neural network** architecture ⇢ unsupervised classification of graphs / supervised regression of energy functional
- **Sobolev training** for robust energy functional prediction
- Inferred **encoded vectors** ⇢ **anisotropy descriptors**
- Introduced hybrid neural network model in a **phase-field fracture** framework

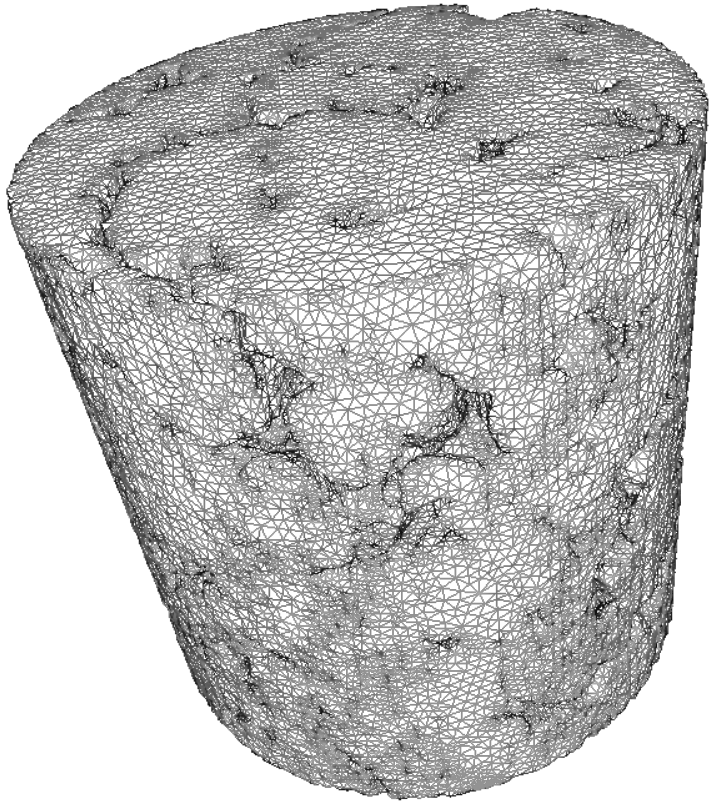


Future work:

- Expand geometric learning framework to other domains (granular materials, flow network)
- Explore path-dependent behavior (plasticity) – graphs changing in time
- More expressive graph structures – direction, weighted edges, higher order Laplacians.

Acknowledgements

- Army Research Office: Young Investigator Program Award
- Air Force Office of Scientific Research, Young Investigator Program Award
- National Science Foundation: NSF CAREER award
- Sandia National Laboratories
- US Department of Energy Office of Nuclear Energy
- Columbia University



THANK YOU!

More information can be found at
www.poromehanics.org