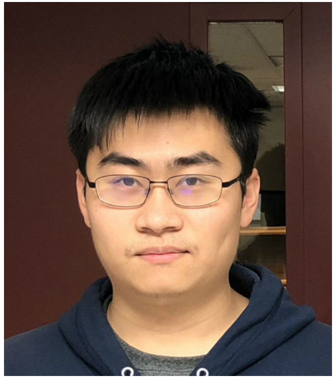
Thermodynamic-informed machine learning for solid mechanics

Steve Sun
Department of Civil Engineering and Engineering Mechanics
Columbia University

Major contributors of this talk

Nikolas Napoleon Vlassis (PhD student)
Geometric Learning of poly-crystals



Kun Wang (PhD graduate)
game theory for model validation

Bahador Bahmani (PhD student)
Model-free poromechanics

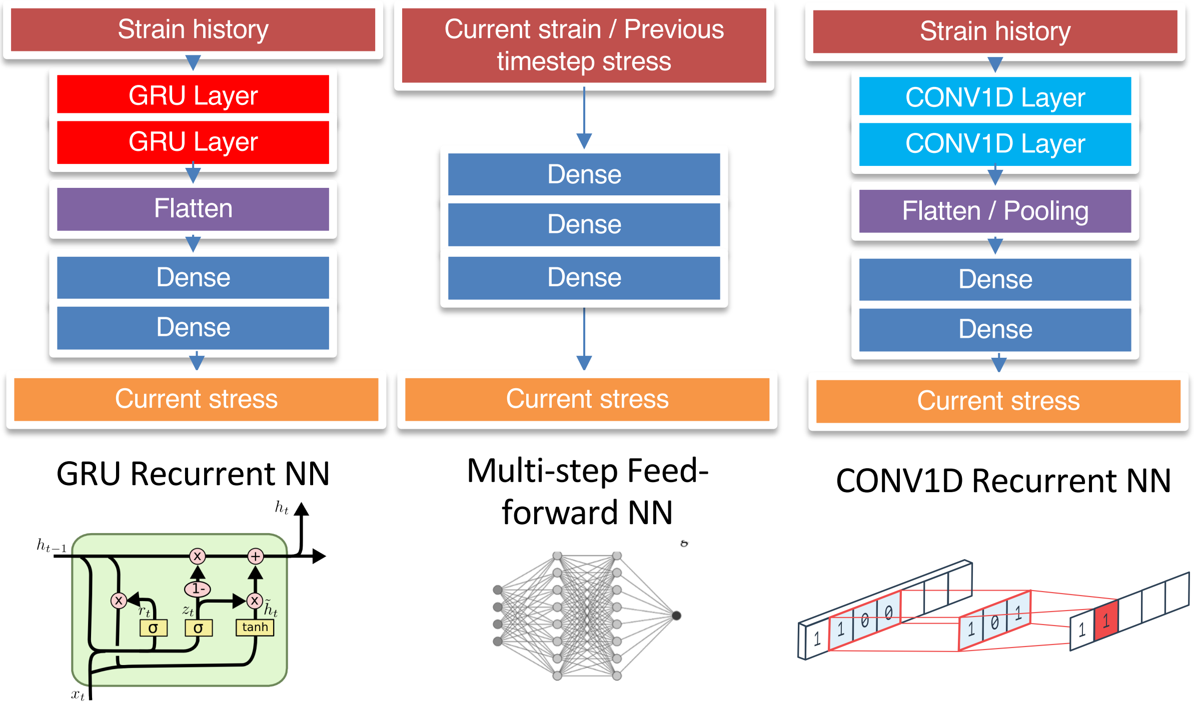
Ran Ma
Associate research scientist
FFT simulations for fast database generation

Yousef Heider
Former associate research scientist,
Lie group mapping

What can we learn from what the machine learned?

Classical elastoplasticity black-box neural networks

Classical “recurrent” black-box architectures



Machine learning approaches for constitutive modeling

Easy to interpret results (e.g. check convexity..etc.)

How do we get here?
How do we gain new knowledge?
How do we verify beyond data?

Neural network material parameter identification
(but nothing new discovered, just use NN or ML as an optimization tool)

Requires big data to function

Perform well with limited data

Symbolic regression
(often leads to unreadable long expressions)

Model-free approach
(no model to interpret)

Neural network constitutive laws
(indirect interpretation)

Hard to interpret results

How about hand-crafted plasticity models?

Von Mises J2 plasticity 1910s

Drucker-Prager 1950s

Critical state soil mechanics 1960s

Sand model with fabric tensor 2004

Dependence on mean effective pressure

Dependence on void ratio

Dependence on fabric tensor

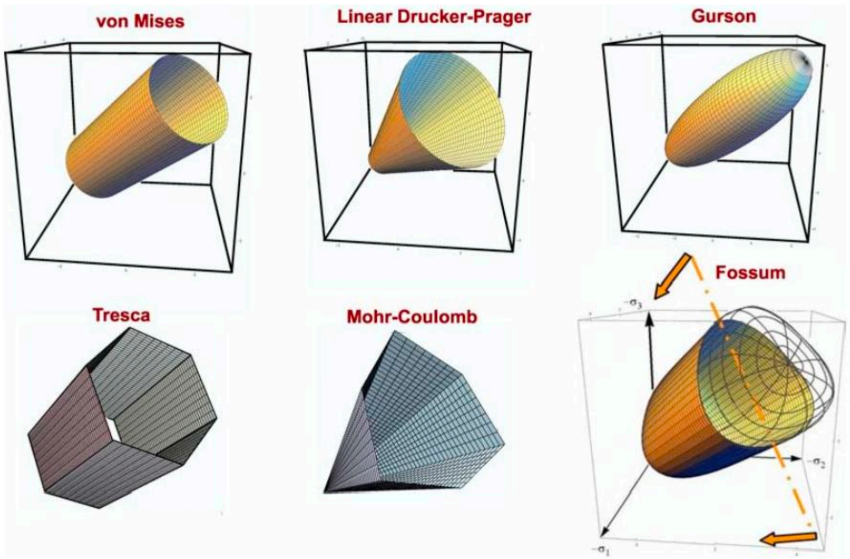


Figure from Rebecca Brannon

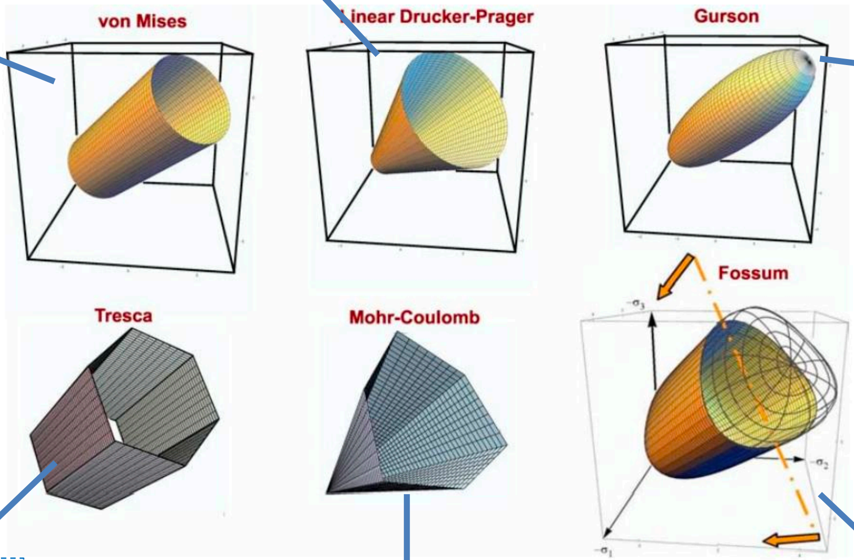
New yield surface takes long time to discover

$$\tilde{f}(\sigma, \kappa) = \sqrt{2J_2} - \kappa \leq 0$$

$$\tilde{f} = \sqrt{\frac{2}{3}q - (A - Bp)} \leq 0$$

$$\tilde{f}(\sigma, \sigma_M, f^*) = \frac{\sigma_e^2}{\sigma_M^2} + 2q_1 f^* \cosh\left(\frac{q_2 tr\sigma}{2\sigma_M}\right) - 1 - (q_1 f^*)^2 <= 0$$

- **Yield surface shapes** and their evolution **(hardening)** are described with **complex mathematical expressions.**
- **Many yield surface models require special treatment** and model specific algorithms (e.g. lack of smoothness in the Mohr-Coulomb surface tips).



$$\tilde{f}(\sigma_1, \sigma_2, \sigma_3, S_y) = \frac{1}{2}\max(|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|) - \frac{1}{2}S_y \leq 0$$

$$\tilde{f} = 2\sqrt{J_2}\cos(\theta - \pi/6) - Y \leq 0$$

$$\tilde{f}(\boldsymbol{\sigma}, \boldsymbol{\alpha}, \boldsymbol{\kappa}) = J_2^{\xi}\Gamma^2(\bar{\theta}) - [F_f(I_1) - N]^2 F_c^2(I_1, \boldsymbol{\kappa}) \leq 0$$

New hardening model and flow rules takes long time to discover and they are even harder to calibrate

GÉOTECHNIQUE

Géotechnique

ISSN 0016-8505 | **E-ISSN** 1751-7656

Volume 63 Issue 16, December 2013, pp. 1406-1418 < Prev Next >

Anatomy of rotational hardening in clay plasticity

Authors: Y.F. DAFALIAS*, and M. TAIEBAT†

https://doi.org/10.1680/geot.12.P.197

Published Online: May 25, 2015

Keywords: anisotropy ; clays ; constitutive relations ; fabric/structure of soils ; ... Show All

Open PDF

JOURNAL OF GEOPHYSICAL RESEARCH

Solid Earth

AN AGU JOURNAL

Regular Section | Full Access

Stabilization of rapid frictional slip on a weakening fault by dilatant hardening

John W. Rudnicki, Chao-Hsun Chen

First published: 10 May 1988 | https://doi.org/10.1029/JB093iB05p04745 | Citations: 73

ELSEVIER

Computer Methods in Applied Mechanics and Engineering

Volume 194, Issues 50–52, 1 December 2005, Pages 5109-5138

Implicit numerical integration of a three-invariant, isotropic/kinematic hardening cap plasticity model for geomaterials

C.D. Foster [a], R.A. Regueiro [b], A.F. Fossum [c], R.I. Borja [a]

Show more

Google Scholar | Gurson model

Articles | About 12,400 results (**0.05** sec)

Any time
Since 2021
Since 2020
Since 2017
Custom range...

Sort by relevance
Sort by date

include patents
include citations

Create alert

Modification of the **Gurson model** for shear failure — [PDF] psu.edu, Columbia e-link >>
K Nahshon, JW Hutchinson - European Journal of Mechanics-A/Solids, 2008 - Elsevier
Recent experimental evidence points to limitations in characterizing the critical strain in ductile fracture solely on the basis of stress triaxiality. A second measure of stress state, such as the Lode parameter, is required to discriminate between axisymmetric and shear …
Cited by 874 Related articles All 8 versions Web of Science: 580

[HTML] A complete **Gurson model** approach for ductile fracture — [HTML] sciencedirect.com, Columbia e-link >>
ZL Zhang, C Thaulow, J Ødegård - Engineering Fracture Mechanics, 2000 - Elsevier
Recently, a complete **Gurson model** has been introduced by the authors. The complete **Gurson model** is a combination of the modified **Gurson model** which deals with microvoid nucleation and growth, and a physical microvoid coalescence criterion based on the plastic …
Cited by 391 Related articles All 3 versions Web of Science: 243

[HTML] The modified **Gurson model** accounting for the void size effect — [HTML] sciencedirect.com, Columbia e-link >>
J Wen, Y Huang, KC Hwang, C Liu, M Li - International Journal of Plasticity, 2005 - Elsevier
Abstract The **Gurson model** [J. Engrg. Mater. Technol. 99 (1977) 2] has been widely used to study the deformation and failure of metallic materials containing microvoids. The void volume fraction is the only parameter representing voids since the void size does not come …
Cited by 171 Related articles All 9 versions Web of Science: 115

[HTML] Ductile shear failure or plug failure of spot welds modelled by modified **Gurson model** — [HTML] sciencedirect.com, Columbia e-link >>
KL Nielsen, V Tvergaard - Engineering Fracture Mechanics, 2010 - Elsevier
For resistance spot welded shear-lab specimens, interfacial failure under ductile shearing or ductile plug failure are analyzed numerically, using a shear modified **Gurson model**. The interfacial shear failure occurs under very low stress triaxiality, where the original **Gurson** …
Cited by 153 Related articles All 5 versions Web of Science: 121

Related searches

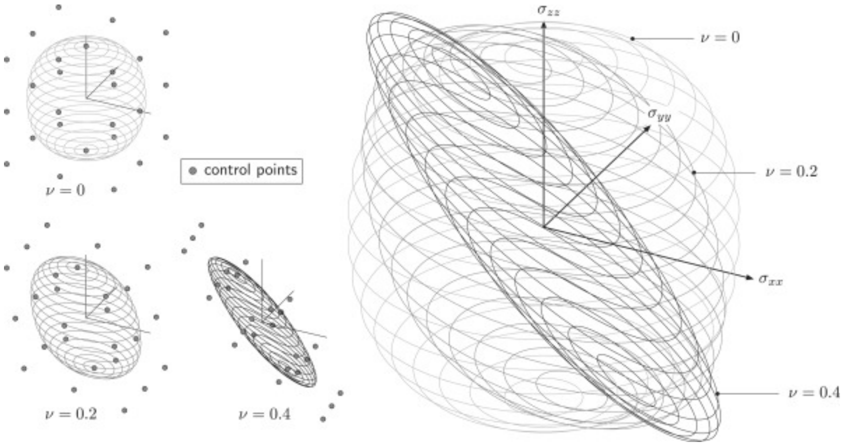
gurson model **ductile fracture**	gurson model **shear**
gurson model **failure**	gurson model **void**
complete gurson model	gurson model **numerical**
modified gurson model	gurson model **abaqus**

[HTML] A modified **Gurson model** and its application to punch-out experiments — [HTML] sciencedirect.com, Columbia e-link >>
K Nahshon, Z Xue - Engineering fracture mechanics, 2009 - Elsevier
Recent experimental evidence has reiterated that ductile fracture is a strong function of stress triaxiality. Under high stress triaxiality loading, failure occurs as a result of void growth and subsequent necking of inter-void ligaments while under low stress triaxiality failure is …
Cited by 87 Related articles All 5 versions Web of Science: 61

[HTML] Aspects on the finite-element implementation of the **Gurson model** including parameter identification — [HTML] sciencedirect.com, Columbia e-link >>
R Mahnken - International Journal of Plasticity, 1999 - Elsevier
In this contribution, various aspects on the finite-element implementation of the **Gurson model** are considered. In particular, a linear representation for the plastic potential is used, which shows superior convergence property in the local iteration procedure compared to the …
Cited by 82 Related articles All 3 versions Web of Science: 43

[HTML] Effect of a shear modified **Gurson model** on damage development in a FSW tensile specimen — [HTML] sciencedirect.com, Columbia e-link >>
KL Nielsen, V Tvergaard - International Journal of Solids and Structures, 2009 - Elsevier
For a friction stir welded aluminum plate the resistance to ductile failure is studied by analyzing tensile test specimens cut out across the weldline. As the stress triaxiality is rather low in these tests, the **Gurson** material **model** is not expected to give a very accurate …
Cited by 87 Related articles All 7 versions Web of Science: 57

Previous work on generalizing plasticity models



NURB Yield surface for perfect plasticity (Coombs, Petit, Motlagh, CMAME, 2016)

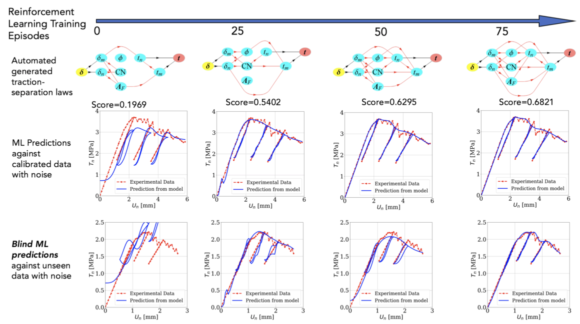


Figure 6: *Improved calibration and blind prediction scores throughout the training. As time progresses, the AI learn to write models with increasingly precise predictions. After 75 episodes (i.e. 75 different constitutive laws are built, both the calibration exercises and blind predictions (blue) are able to yield excellent matches with the benchmark (red).*

(P6) Plastic flow direction defined as [63,104]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ d_g = (1+\alpha)(M_g + q/p) \end{cases},$$

where α, M_g are material parameters.

(P7) Plastic flow direction defined as [44]

$$\begin{cases} m_v^{flow} = \dfrac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \dfrac{1}{\sqrt{1+d_g^2}} \\ d_g = (1+\alpha)(M_g \exp m_g \psi + q/p) \\ \psi = e - e_{c0} + \tilde{\lambda}(p/p_{at})^a \end{cases},$$

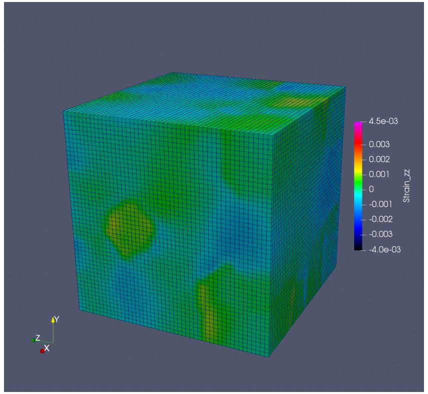
Cooperative game for deducing plasticity models via deep reinforcement learning (Wang & Sun, CMAME 2018, Wang, Sun, Du, CM, 2019)

Only works for perfect plasticity

Need to program all choices in the decision tree and required prior knowledge

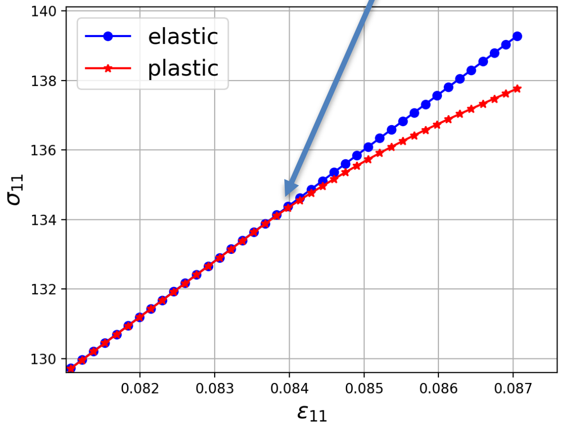
Machine learning plasticity with level set hardening

Machine learning with sub-goals -- Identification of initial yield surface using elasticity model

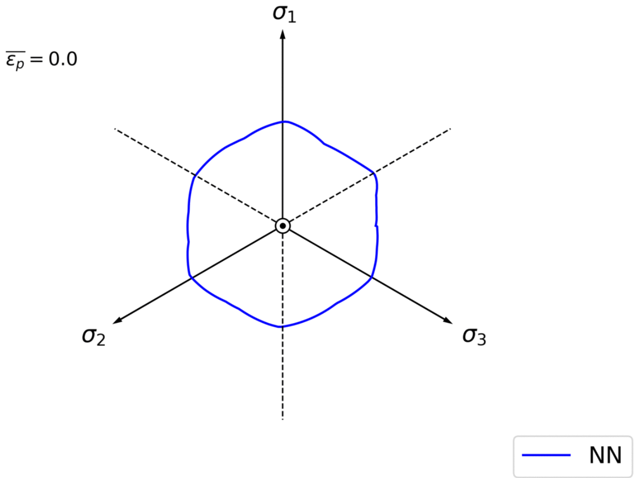


RVE simulations (w FFT solver)

Initial yield point



Identify initial yield surface and hardening

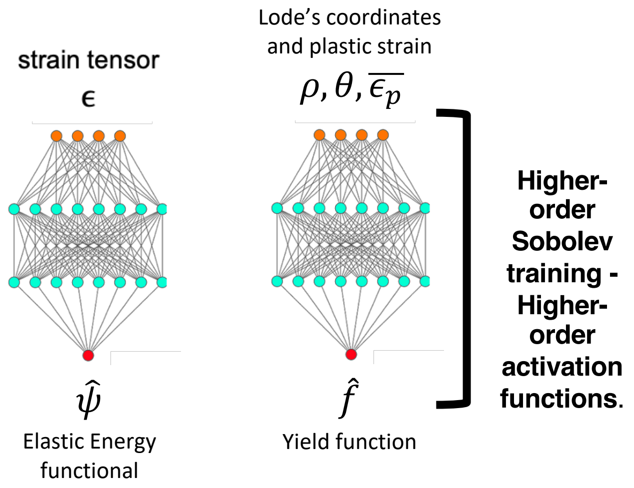


Sobolev training for yield function

Departure from classical elastoplasticity black-box neural networks

<u>Our approach:</u>

Elastic and **plastic** behaviors **decomposed**.

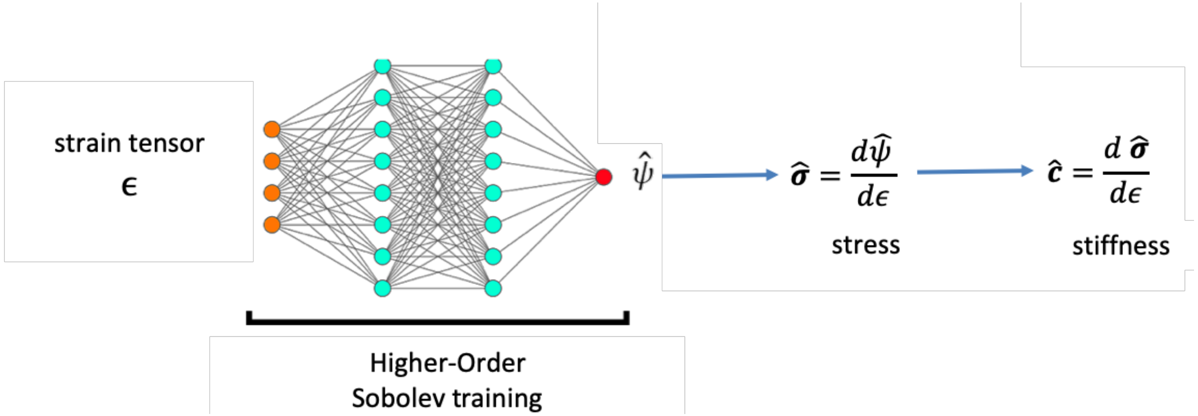


- **Two neural networks** – hyperelastic energy functional and yield function.
- Utilize plasticity theory to combine the two networks (**return mapping algorithm**).

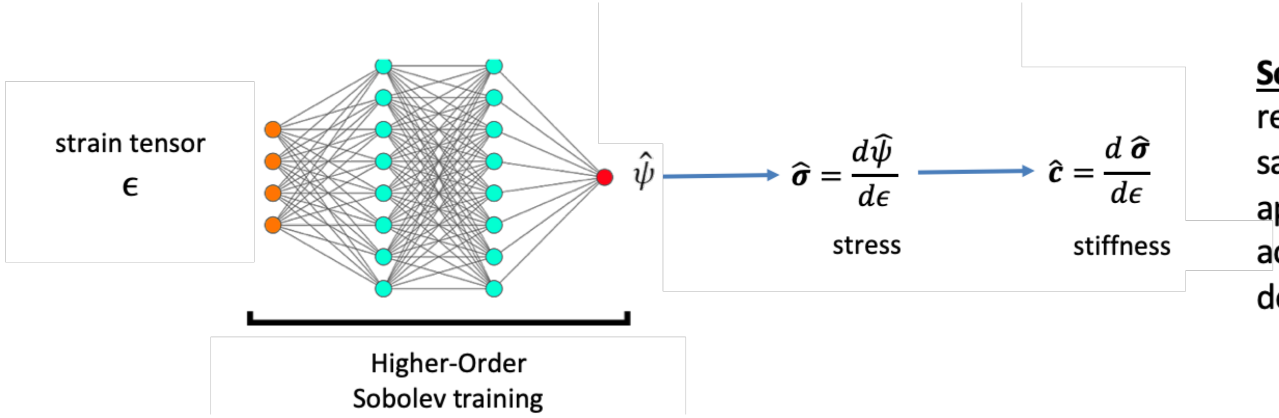
Vlassis et al, Sobolev training of thermodynamic-informed neural networks for smoothed elasto-plasticity models with level set hardening (2020)

ML learning for hyperelasticity energy functionals

Sobolev training of a hyperelastic energy functional



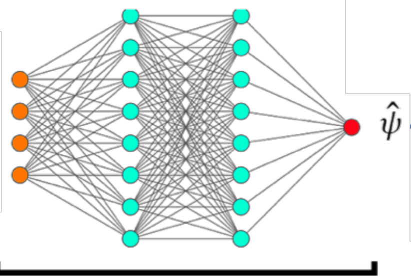
Sobolev training of a hyperelastic energy functional



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

Sobolev training of a hyperelastic energy functional

strain tensor
ϵ



$$\hat{\boldsymbol{\sigma}} = \frac{d\hat{\psi}}{d\epsilon} \quad \text{stress}$$

$$\hat{\boldsymbol{c}} = \frac{d\,\hat{\boldsymbol{\sigma}}}{d\epsilon} \quad \text{stiffness}$$

Higher-Order
Sobolev training

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

---

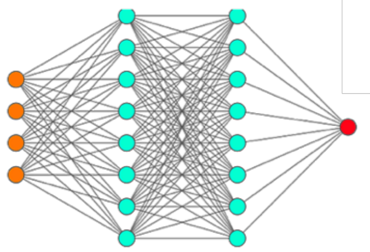
L_2 norm: Constrain energy predictions [classical NN]:

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\psi_i - \widehat{\psi_i}\right\|_2^2$$

Sobolev training of a hyperelastic energy functional

strain tensor

ϵ



$\hat{\psi}$ → $\hat{\boldsymbol{\sigma}} = \frac{d\hat{\psi}}{d\epsilon}$ → $\hat{\boldsymbol{c}} = \frac{d\,\hat{\boldsymbol{\sigma}}}{d\epsilon}$

stress　　stiffness

Higher-Order
Sobolev training

<u>Sobolev training:</u>
requires less data samples, smooth approximated functions, accurate approximated derivatives

---

$\boldsymbol{L_2}$ **norm**: Constrain energy predictions [classical NN]:

$$\frac{1}{N}\sum_{i=1}^{N}\|\psi_i - \hat{\psi}_i\|_2^2$$

$\boldsymbol{H_1}$ **norm:** Constrain energy and stress predictions [Sobolev training, Google Deep Mind]:

$$\frac{1}{N}\sum_{i=1}^{N}\|\psi_i - \hat{\psi}_i\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\sigma}_i - \hat{\boldsymbol{\sigma}}_i\|_2^2$$

$\boldsymbol{H_2}$ **norm:** Constrain energy, stress and stiffness predictions [our extension]:

$$\frac{1}{N}\sum_{i=1}^{N}\|\psi_i - \hat{\psi}_i\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\sigma}_i - \hat{\boldsymbol{\sigma}}_i\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{c}_i - \hat{\boldsymbol{c}}_i\|_2^2$$

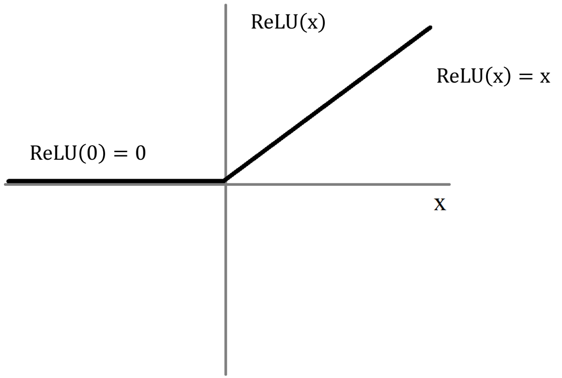
use of higher-order activations functions

- **Interpretable** neural network derivatives – higher-order derivative – thermodynamic constraints in the loss function (**Sobolev training)**.
- H^2 **training**: **higher accuracy** for energy, stress, and stiffness for the **same number of samples.**

Higher-order activation functions

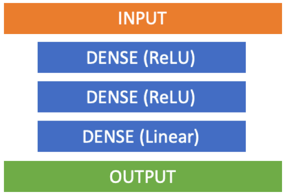
In order to implement **implicit algorithms** (elastoplasticity) ⇢ **second-order derivatives** of functionals

Classical feed-forward architectures



$$\mathrm{ReLU}(\bullet) = \max(0, \bullet)$$

Architecture: ddd



- Classical regression activation functions ⇢ **piece-wise linear** ⇢ **second-order derivatives** of networks are **locally zero**

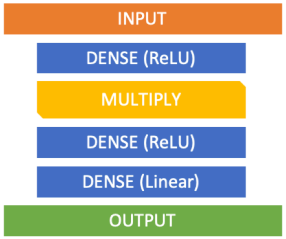
Our approach:

Increase "non-linearity" by adding **Multiply** layers

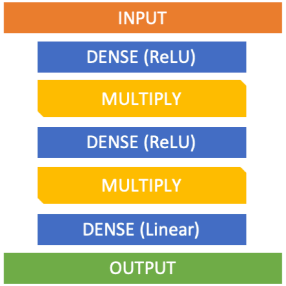
$$\boldsymbol{h}^n = \mathrm{Multiply}(\boldsymbol{h}^{n-1}) = \boldsymbol{h}^{n-1} \circ \boldsymbol{h}^{n-1}$$

where ∘ is the element-wise multiplication of vectors

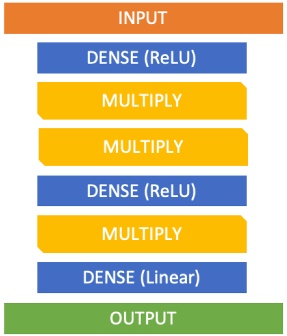
Architecture: dmdd



Architecture: dmdmd

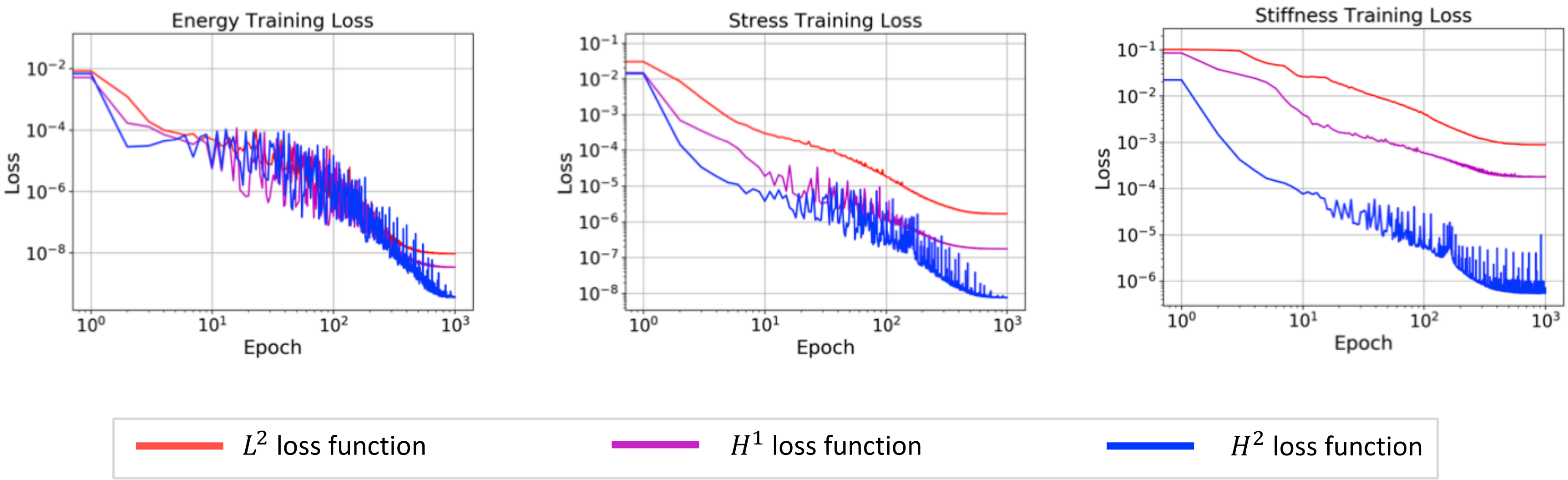


Architecture: dmmdmd



Increase number of **Multiply** layers:

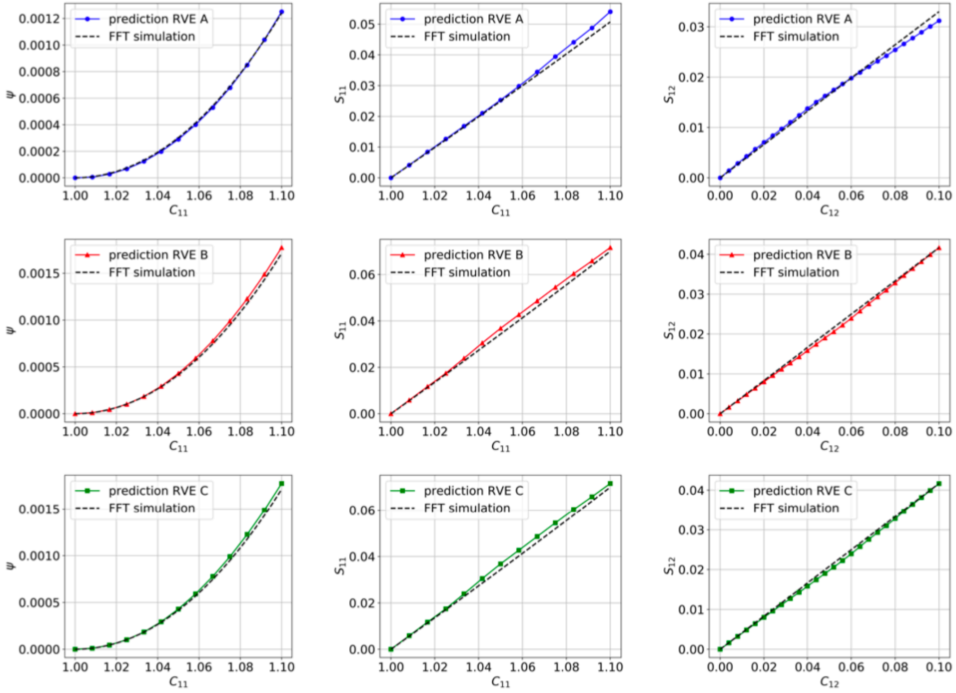
L^2 norm - H^1 norm - H^2 norm Training Comparison



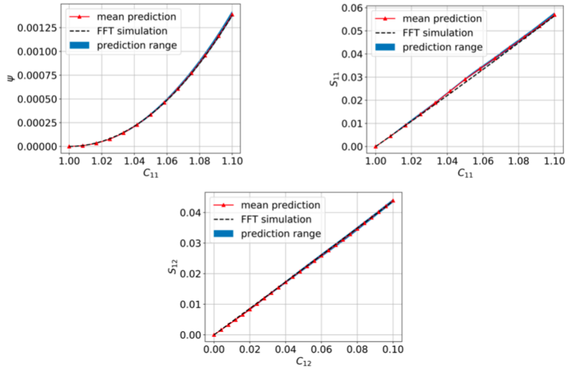
***H^2* training** and **higher order activation functions** procure **higher accuracy in the energy, stress, and stiffness** of predictions for the **same number of samples** compared to the classical L_2 training methods.

Predictions of polycrystal elasticity for calibrated and unseen RVEs

Predictions of elastic responses on unseen RVEs

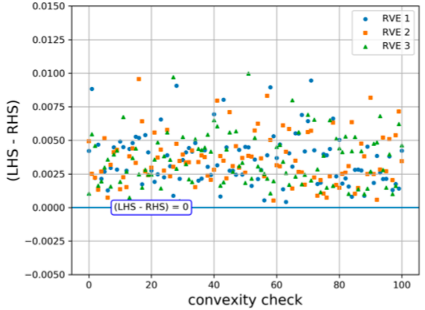


Graph isomorphism test



Convexity test

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

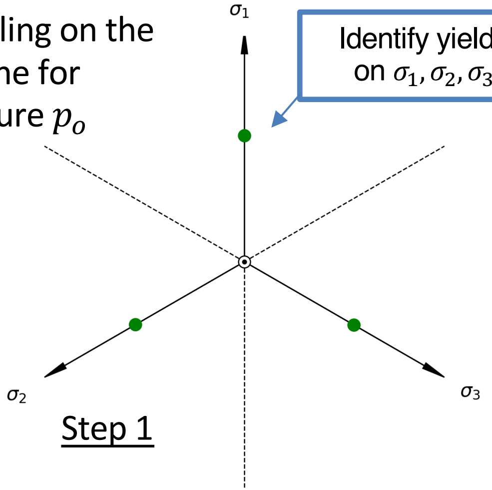


Vlassis, Ma & Sun, CMAME, 2020, Vlassis & Sun, CMAME, 2021

ML learning for evolving yield function

Efficient yield function data sampling for new material

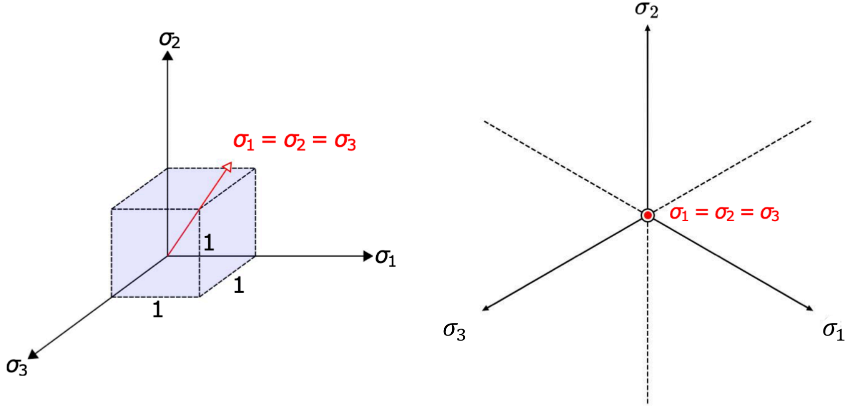
Sampling on the π-plane for pressure p_o



Identify yield points on $\sigma_1, \sigma_2, \sigma_3$ axes

Step 1

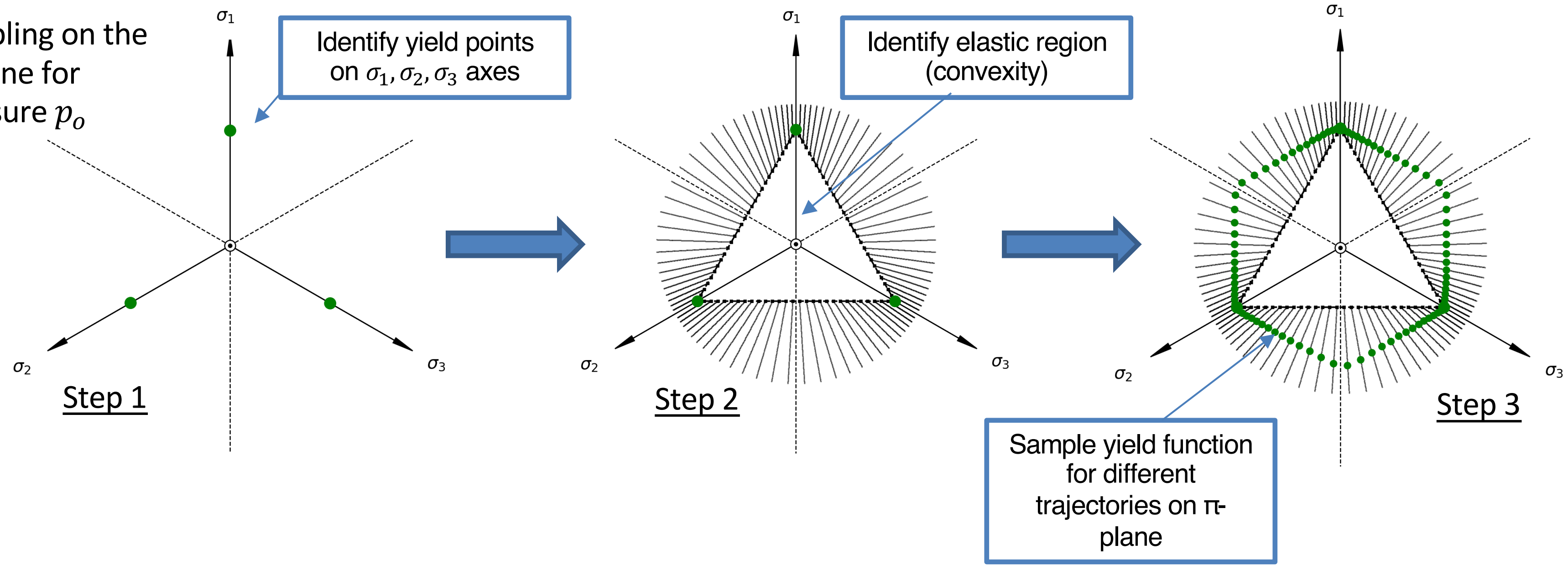
Lower-dimensional stress representation on the π-plane with **Lode's coordinates** (ρ,θ)



The π-plane is perpendicular to the space diagonal and is passing through the origin of the principal stress axes.

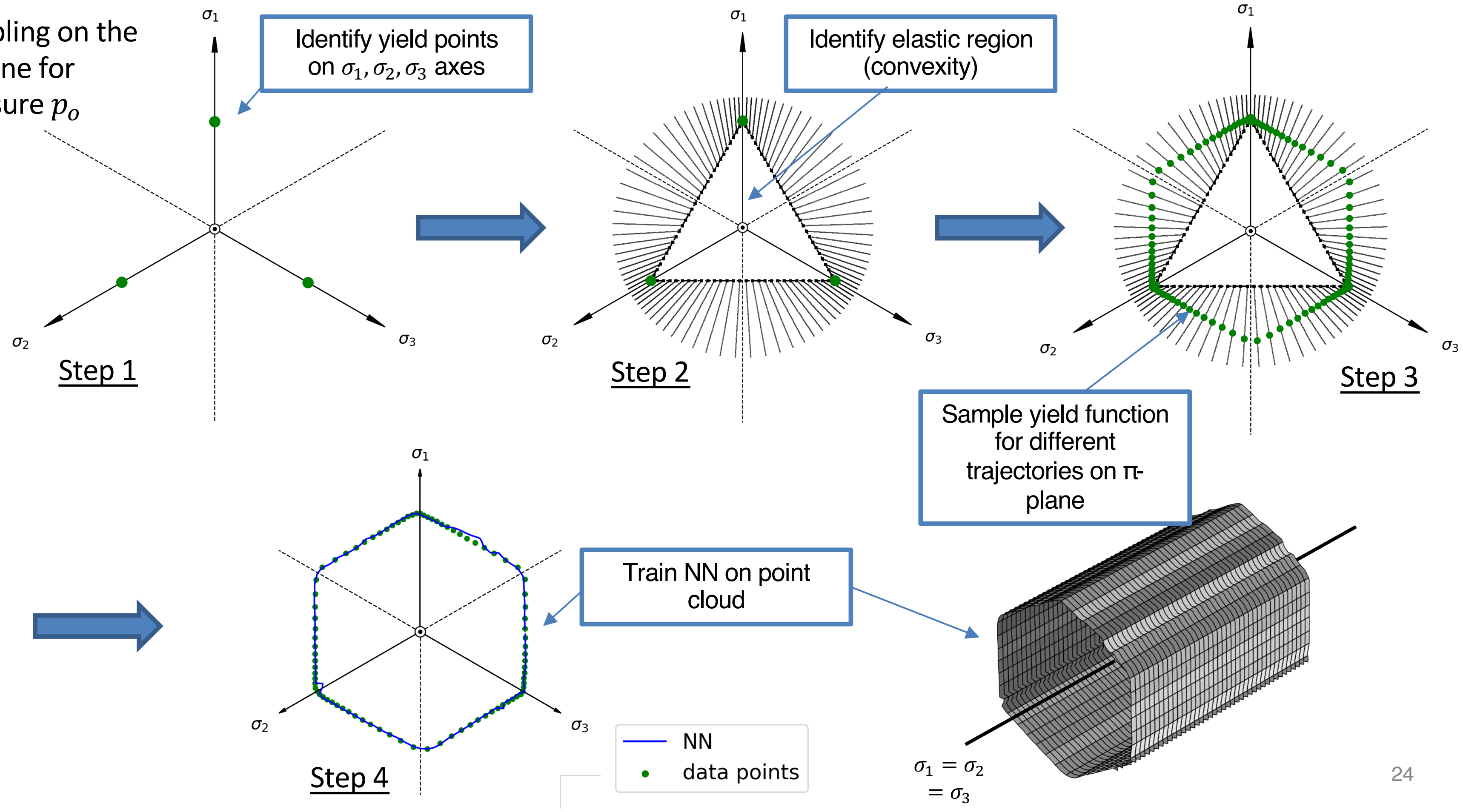
Efficient yield function data sampling for new material

Sampling on the π-plane for pressure p_o



Efficient yield function data sampling for new material

Sampling on the π-plane for pressure p_o



Converting yield surface into a signed distance function

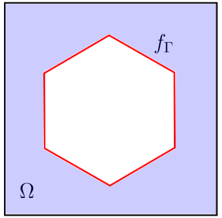
Preprocess data as a level set initialization problem

1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11},\sigma_{22},\sigma_{33},\sigma_{12},\sigma_{23},\sigma_{13}) = \overline{\boldsymbol{x}}(\sigma_1,\sigma_2,\sigma_3) = \widehat{\boldsymbol{x}}(\rho,\theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

$$|\nabla^{\widehat{x}}\phi| = 1 \longrightarrow \left(\frac{\partial\phi}{\partial\rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial\phi}{\partial\theta}\right)^2 = 1.$$



3. The resultant yield surface becomes

$$\phi(\widehat{\mathbf{x}},t) = \begin{cases} d(\widehat{\mathbf{x}}) & \text{outside } f_\Gamma \text{ (inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{ (yielding)} \\ -d(\widehat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\widehat{\boldsymbol{x}}) = \min\left(|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}_\Gamma|\right).$$

Converting yield surface into a signed distance function

Preprocess data as a level set initialization problem

1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11},\sigma_{22},\sigma_{33},\sigma_{12},\sigma_{23},\sigma_{13}) = \bar{\boldsymbol{x}}(\sigma_1,\sigma_2,\sigma_3) = \hat{\boldsymbol{x}}(\rho,\theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

$$|\nabla^{\hat{x}}\phi| = 1 \longrightarrow \left(\frac{\partial\phi}{\partial\rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial\phi}{\partial\theta}\right)^2 = 1.$$

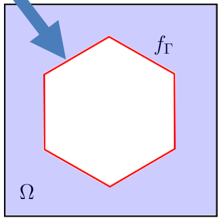
3. The resultant yield surface becomes

$$\phi(\hat{\mathbf{x}},t) = \begin{cases} d(\hat{\mathbf{x}}) & \text{outside } f_\Gamma \text{ (inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{ (yielding)} \\ -d(\hat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\hat{\boldsymbol{x}}) = \min\left(|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_\Gamma|\right).$$

Yield surface data



Converting yield surface into a signed distance function

Preprocess data as a level set initialization problem

1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11},\sigma_{22},\sigma_{33},\sigma_{12},\sigma_{23},\sigma_{13}) = \overline{\boldsymbol{x}}(\sigma_1,\sigma_2,\sigma_3) = \widehat{\boldsymbol{x}}(\rho,\theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

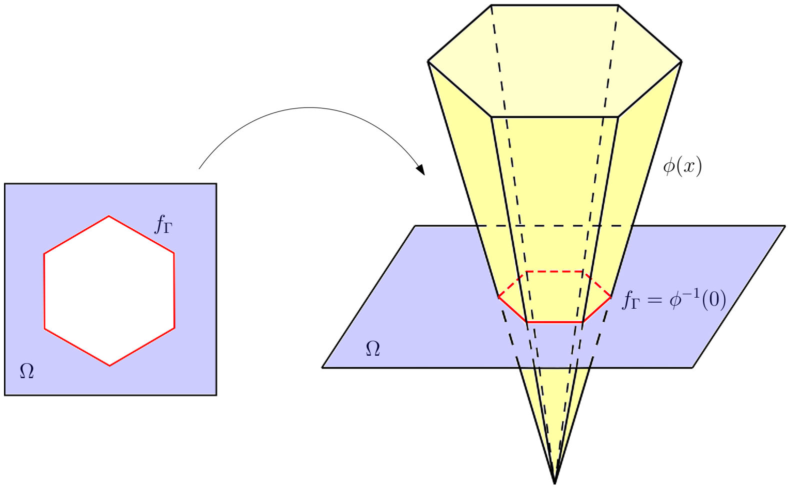
$$|\nabla^{\widehat{x}}\phi| = 1 \longrightarrow \left(\frac{\partial\phi}{\partial\rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial\phi}{\partial\theta}\right)^2 = 1.$$

3. The resultant yield surface becomes

$$\phi(\widehat{\mathbf{x}},t) = \begin{cases} d(\widehat{\mathbf{x}}) & \text{outside } f_\Gamma \text{(inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{(yielding)} \\ -d(\widehat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\widehat{\boldsymbol{x}}) = \min\left(|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}_\Gamma|\right).$$



Converting yield surface into a signed distance function

Preprocess data as a level set initialization problem

1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{13}) = \overline{\boldsymbol{x}}(\sigma_1, \sigma_2, \sigma_3) = \widehat{\boldsymbol{x}}(\rho, \theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

$$|\nabla^{\hat{x}} \phi| = 1 \longrightarrow \left(\frac{\partial \phi}{\partial \rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial \phi}{\partial \theta}\right)^2 = 1.$$

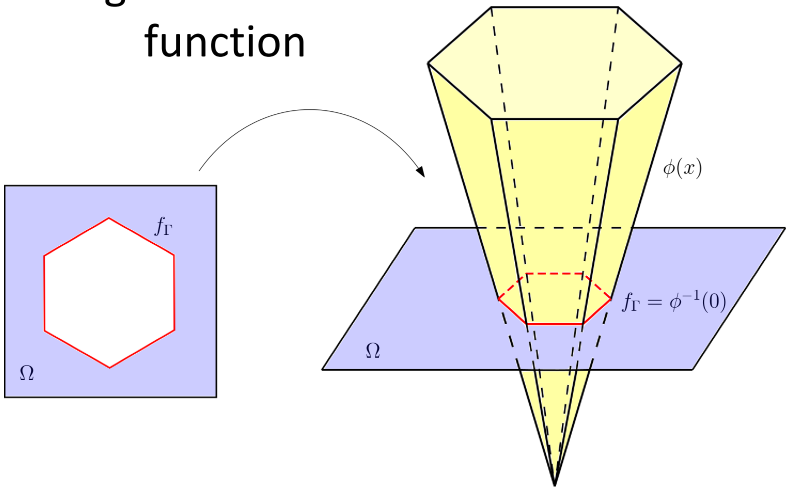
3. The resultant yield surface becomes

$$\phi(\widehat{\mathbf{x}}, t) = \begin{cases} d(\widehat{\mathbf{x}}) & \text{outside } f_\Gamma \text{ (inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{ (yielding)} \\ -d(\widehat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\widehat{\boldsymbol{x}}) = \min\left(|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}_\Gamma|\right).$$

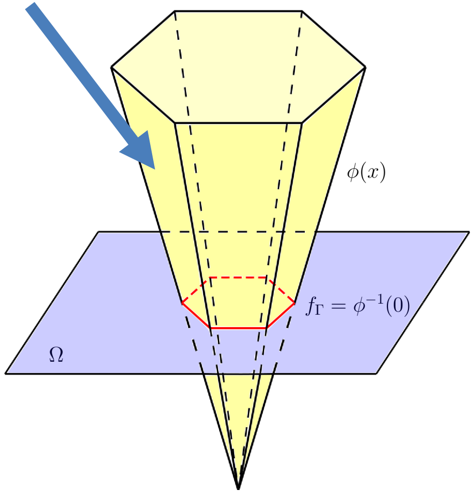
Pre-process into signed distance function



Converting yield surface into a signed distance function

Preprocess data as a level set initialization problem

Signed distance function



1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{13}) = \overline{\boldsymbol{x}}(\sigma_1, \sigma_2, \sigma_3) = \widehat{\boldsymbol{x}}(\rho, \theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

$$|\nabla^{\widehat{\boldsymbol{x}}} \phi| = 1 \longrightarrow \left(\frac{\partial \phi}{\partial \rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial \phi}{\partial \theta}\right)^2 = 1.$$

3. The resultant yield surface becomes

$$\phi(\widehat{\mathbf{x}}, t) = \begin{cases} d(\widehat{\mathbf{x}}) & \text{outside } f_\Gamma \text{ (inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{ (yielding)} \\ -d(\widehat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\widehat{\boldsymbol{x}}) = \min\left(|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}_\Gamma|\right).$$

Converting yield surface into a signed distance function

Preprocess data as a level set initialization problem

1. Reduce dimensionality with **π-plane**:

$$\boldsymbol{x}(\sigma_{11},\sigma_{22},\sigma_{33},\sigma_{12},\sigma_{23},\sigma_{13}) = \overline{\boldsymbol{x}}(\sigma_1,\sigma_2,\sigma_3) = \widehat{\boldsymbol{x}}(\rho,\theta).$$

2. Convert yield function into signed distance function by solving **Eikonal equation** in polar coordinates while enforcing the boundary f=0

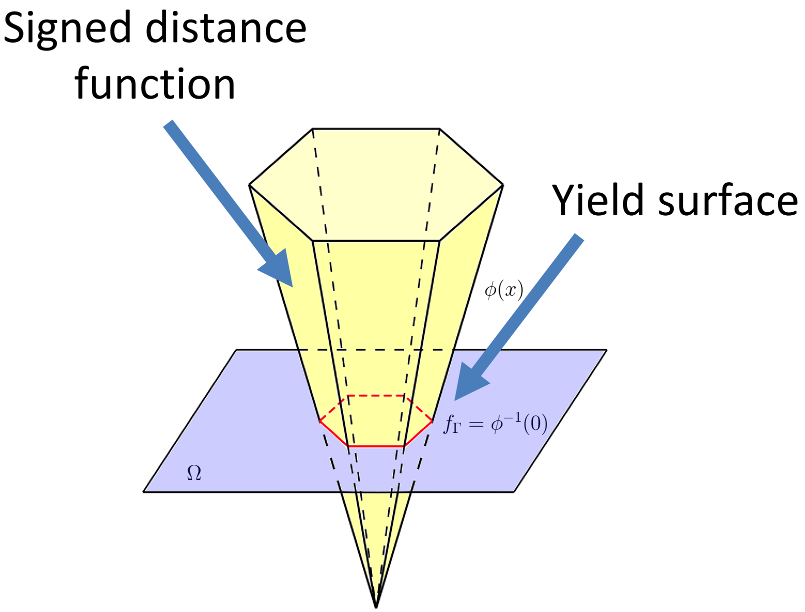
$$|\nabla^{\widehat{x}}\phi| = 1 \longrightarrow \left(\frac{\partial\phi}{\partial\rho}\right)^2 + \frac{1}{\rho^2}\left(\frac{\partial\phi}{\partial\theta}\right)^2 = 1.$$

3. The resultant yield surface becomes

$$\phi(\widehat{\mathbf{x}},t) = \begin{cases} d(\widehat{\mathbf{x}}) & \text{outside } f_\Gamma \text{ (inadmissible stress)} \\ 0 & \text{on } f_\Gamma \text{ (yielding)} \\ -d(\widehat{\mathbf{x}}) & \text{inside } f_\Gamma \text{ (elastic region)} \end{cases},$$

where

$$d(\widehat{\boldsymbol{x}}) = \min\left(|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{x}}_\Gamma|\right).$$



Evolving yield surface by solving a Hamilton-Jacobi Equation via neural networks

Hardening interpreted as a level set extension problem

Hamilton-Jacobi Equation

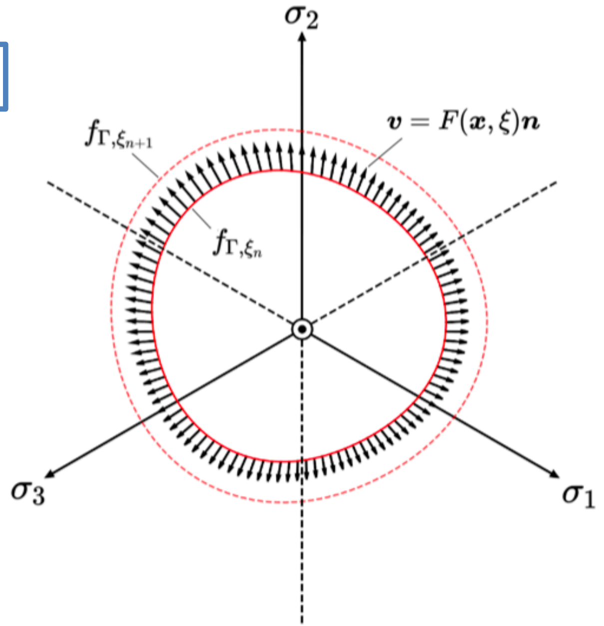
$$\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla^{\hat{\boldsymbol{x}}} \phi = 0,$$

Assuming stationary flow

$$\frac{\partial \phi}{\partial t} + F|\nabla^{\hat{\boldsymbol{x}}} \phi| = 0.$$

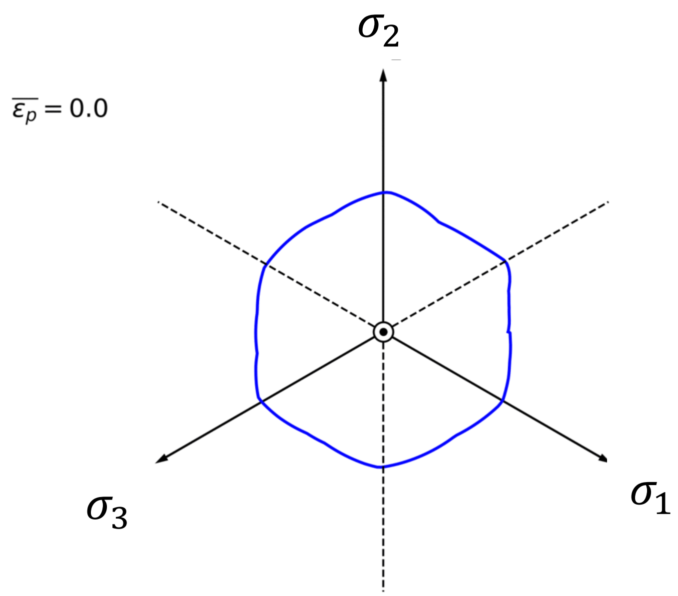
where **F** is the **speed function**. Note that t is a pseudo-time. For our purpose, we replace it with a scalar internal variable **ξ** which is a monotonically increasing with time.

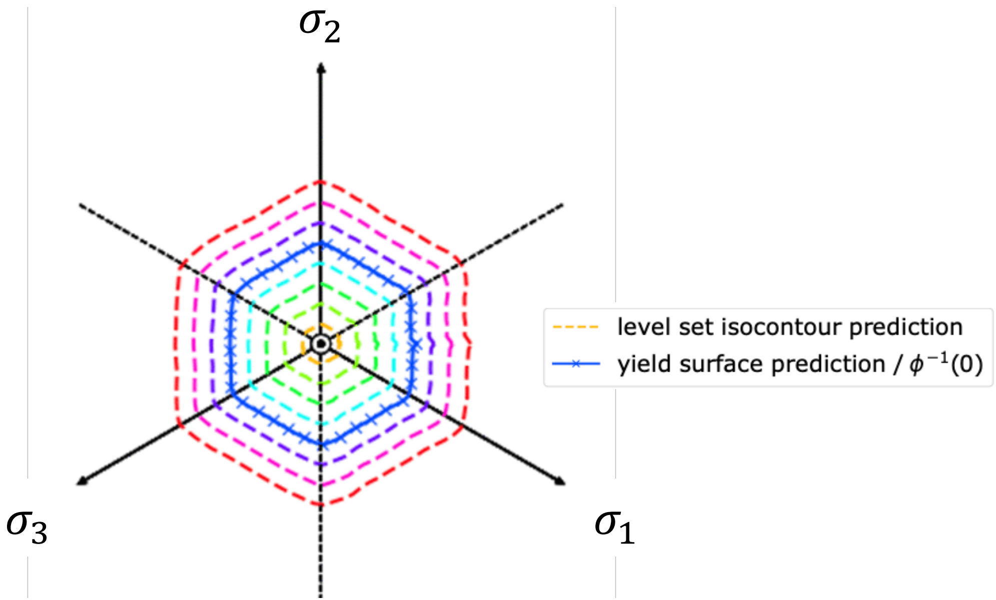
$$F_i \approx \frac{\phi_i - \phi_{i+1}}{\xi_{i+1} - \xi_i}, \quad \text{where} \quad \xi = \int_0^t \dot{\lambda} dt,$$



We then use **neural network** to find F such that the **solution of the H-J equation** is the signed distance function version of **yield function for a given ξ**

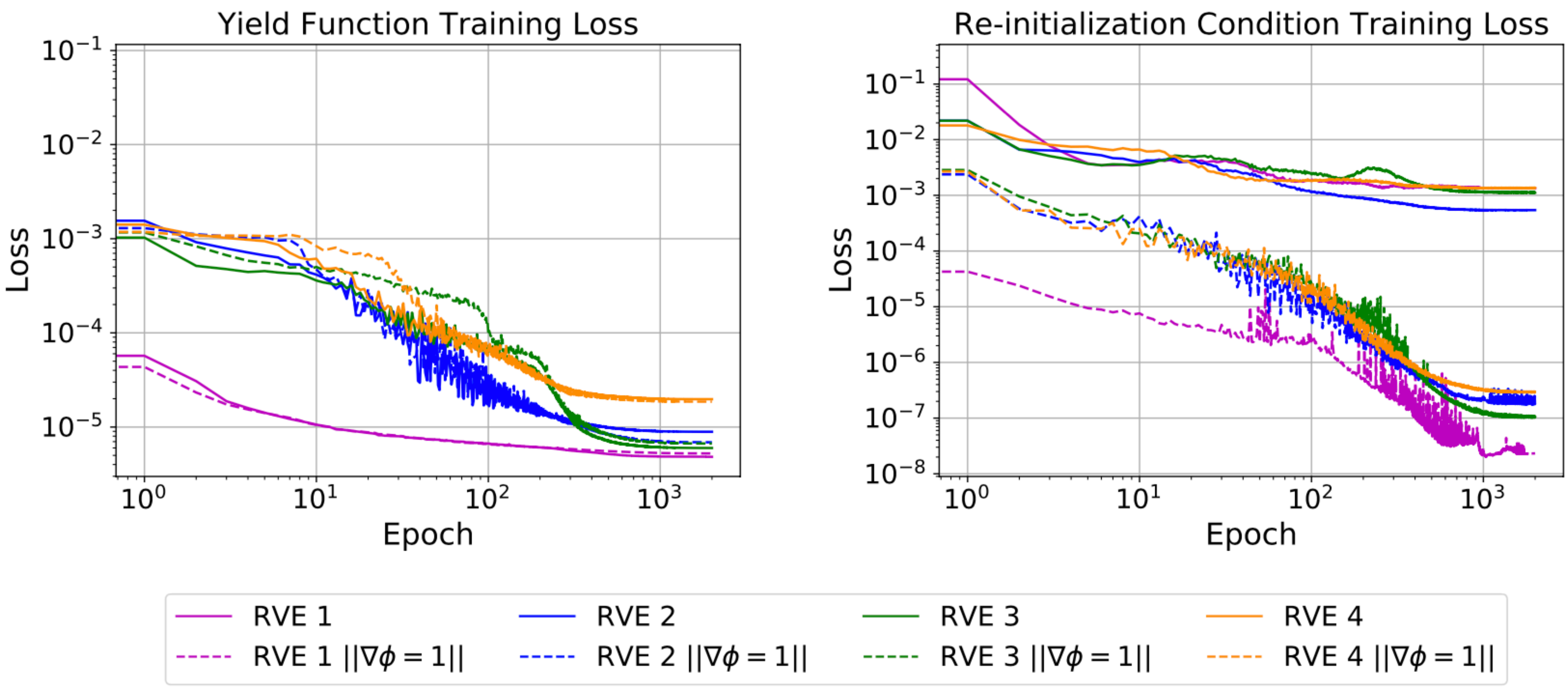
Capture complex hardening mechanisms





- The yield function function neural network can **capture a complex yield surface evolution** and **predict the entire level set** for an internal variable value (accumulated plastic strain $\overline{\varepsilon_p}$).

Performance comparisons with and without signed distance yield function



Vlassis & Sun, under revision

Elastoplasticity Framework with Neural Network Ingredients

Algorithm 1 Return mapping algorithm in strain-space in principal axes for an isotropic hyperelastic-plastic model

1: Compute $\epsilon^{e\,tr}_{n+1} = \epsilon^{e}_{n} + \Delta\epsilon$.
2: Spectrally decompose $\epsilon^{e\,tr}_{n+1} = \sum_{A=1}^{3} \epsilon^{e\,tr}_{A} \boldsymbol{n}^{tr(A)} \otimes \boldsymbol{n}^{tr(A)}$.
3: Compute $\boxed{\sigma^{tr}_{A} = \partial\widehat{\psi}^{e} / \partial\epsilon^{e}_{A}}$ at $\epsilon^{e\,tr}_{n+1}$.
4: **if** $\widehat{f}\left(\sigma^{tr}_{1}, \sigma^{tr}_{2}, \sigma^{tr}_{3}, \kappa_{n}\right) \leq 0$ **then**
5: Set $\boldsymbol{\sigma}_{n+1} = \sum_{A=1}^{3} \sigma^{tr}_{A} \boldsymbol{n}^{tr(A)} \otimes \boldsymbol{n}^{tr(A)}$ and exit.
6: **else**
7: Solve for $\epsilon^{e}_{1}, \epsilon^{e}_{2}, \epsilon^{e}_{3}$, and κ such that $\boxed{\widehat{f}\left(\sigma^{tr}_{1}, \sigma^{tr}_{2}, \sigma^{tr}_{3}, \kappa_{n}\right) = 0.}$
8: Compute $\boldsymbol{\sigma}_{n+1} = \sum_{A=1}^{3} \left(\partial\widehat{\psi}^{e} / \partial\epsilon^{e}_{A}\right) \boldsymbol{n}^{tr(A)} \otimes \boldsymbol{n}^{tr(A)}$ and exit.

- **Two neural networks** can be combined to predict the elastoplastic response – hyperelastic energy functional and yield function.
- Depart from black-box recurrent neural network architectures – more **interpretable** and **robust**
- **Flexibility** to change between different elastic and plastic neural network "ingredients".
- Framework readily usable for **FEM simulations**.

<u>Algorithmic tangent operator:</u>

Made possible with **higher-order Sobolev training** and **higher-order activation functions.**

$$c_{n+1} = \frac{\partial \sigma_{n+1}}{\partial \epsilon_{n+1}} \equiv \frac{\partial \sigma_{n+1}}{\partial \epsilon^{etr}_{n+1}}$$

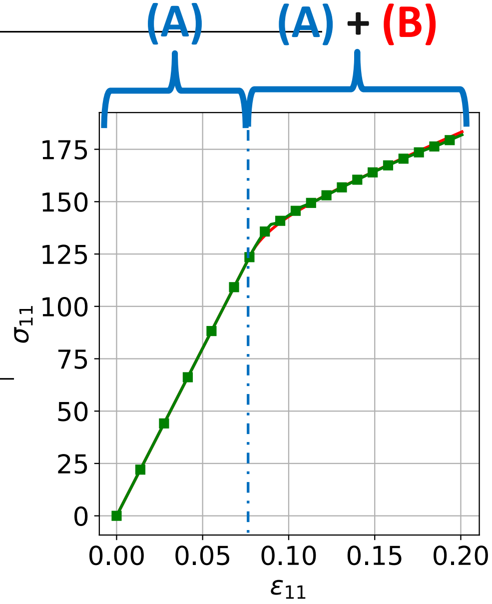
$$\boldsymbol{c}_{n+1} = \sum_{A=1}^{3} \sum_{B=1}^{3} a_{AB} \boldsymbol{m}^{(A)} \otimes \boldsymbol{m}^{(B)} + \frac{1}{2} \sum_{A=1}^{3} \sum_{B \neq A} \left(\frac{\sigma_B - \sigma_A}{\epsilon^{etr}_{B} - \epsilon^{etr}_{A}} \right) \left(\boldsymbol{m}^{(AB)} \otimes \boldsymbol{m}^{(AB)} + \boldsymbol{m}^{(AB)} \otimes \boldsymbol{m}^{(BA)} \right)$$

$$a_{AB} := \frac{\partial \sigma_A}{\partial \epsilon^{etr}_{B}} = \sum_{C=1}^{3} \frac{\partial \sigma_A}{\partial \epsilon^{e}_{C}} \frac{\partial \epsilon^{e}_{C}}{\partial \epsilon^{etr}_{B}} = \sum_{C=1}^{3} \boxed{\left(\frac{\partial^2 \widehat{\psi}^{e}}{\partial \epsilon^{e}_{A} \partial \epsilon^{e}_{C}} \right)} \boxed{\frac{\partial \epsilon^{e}_{C}}{\partial \epsilon^{etr}_{B}}}$$

Elastoplasticity Framework with Neural Network Ingredients

Algorithm 1 Return mapping algorithm in strain-space in principal axes for an isotropic hyperelastic-plastic model

1: Compute $\epsilon^{\mathrm{e\,tr}}_{n+1} = \epsilon^{\mathrm{e}}_{n} + \Delta\epsilon$.
2: Spectrally decompose $\epsilon^{\mathrm{e\,tr}}_{n+1} = \sum_{A=1}^{3} \epsilon^{\mathrm{e\,tr}}_{A} \boldsymbol{n}^{\mathrm{tr}(A)} \otimes \boldsymbol{n}^{\mathrm{tr}(A)}$.
3: Compute $\boxed{\sigma^{\mathrm{tr}}_{A} = \partial\widehat{\psi}^{\mathrm{e}} / \partial\epsilon^{\mathrm{e}}_{A}}$ at $\epsilon^{\mathrm{e\,tr}}_{n+1}$.
4: **if** $\widehat{f}\left(\sigma^{\mathrm{tr}}_{1}, \sigma^{\mathrm{tr}}_{2}, \sigma^{\mathrm{tr}}_{3}, \kappa_n\right) \leq 0$ **then**
5: Set $\boldsymbol{\sigma}_{n+1} = \sum_{A=1}^{3} \sigma^{\mathrm{tr}}_{A} \boldsymbol{n}^{\mathrm{tr}(A)} \otimes \boldsymbol{n}^{\mathrm{tr}(A)}$ and exit.
6: **else**
7: Solve for $\epsilon^{\mathrm{e}}_{1}, \epsilon^{\mathrm{e}}_{2}, \epsilon^{\mathrm{e}}_{3}$, and κ such that $\boxed{\widehat{f}\left(\sigma^{\mathrm{tr}}_{1}, \sigma^{\mathrm{tr}}_{2}, \sigma^{\mathrm{tr}}_{3}, \kappa_n\right) = 0.}$
8: Compute $\boldsymbol{\sigma}_{n+1} = \sum_{A=1}^{3} \left(\partial\widehat{\psi}^{\mathrm{e}} / \partial\epsilon^{\mathrm{e}}_{A}\right) \boldsymbol{n}^{\mathrm{tr}(A)} \otimes \boldsymbol{n}^{\mathrm{tr}(A)}$ and exit.



Algorithmic tangent operator:

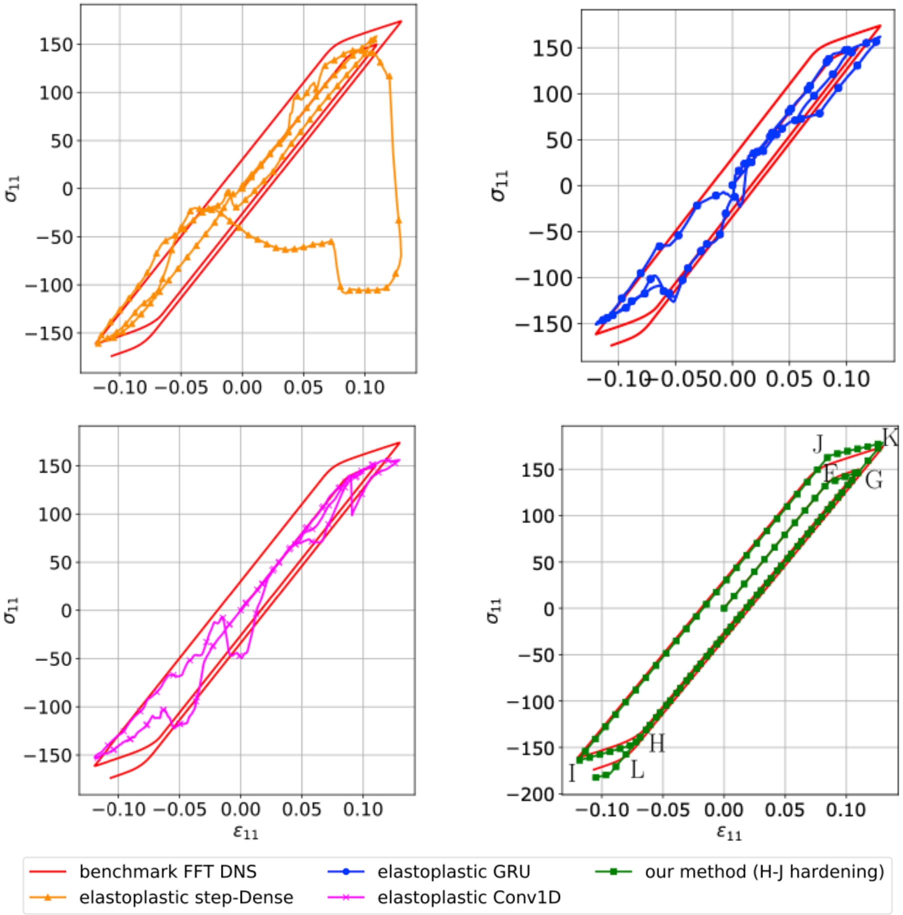
Made possible with **higher-order Sobolev training** and **higher-order activation functions.**

$$c_{n+1} = \frac{\partial \sigma_{n+1}}{\partial \epsilon_{n+1}} \equiv \frac{\partial \sigma_{n+1}}{\partial \epsilon^{\mathrm{etr}}_{n+1}}$$

$$\boldsymbol{c}_{n+1} = \sum_{A=1}^{3}\sum_{B=1}^{3} a_{AB}\boldsymbol{m}^{(A)} \otimes \boldsymbol{m}^{(B)} + \frac{1}{2}\sum_{A=1}^{3}\sum_{B\neq A}\left(\frac{\sigma_B - \sigma_A}{\epsilon^{\mathrm{etr}}_{B} - \epsilon^{\mathrm{etr}}_{A}}\right)\left(\boldsymbol{m}^{(AB)} \otimes \boldsymbol{m}^{(AB)} + \boldsymbol{m}^{(AB)} \otimes \boldsymbol{m}^{(BA)}\right)$$

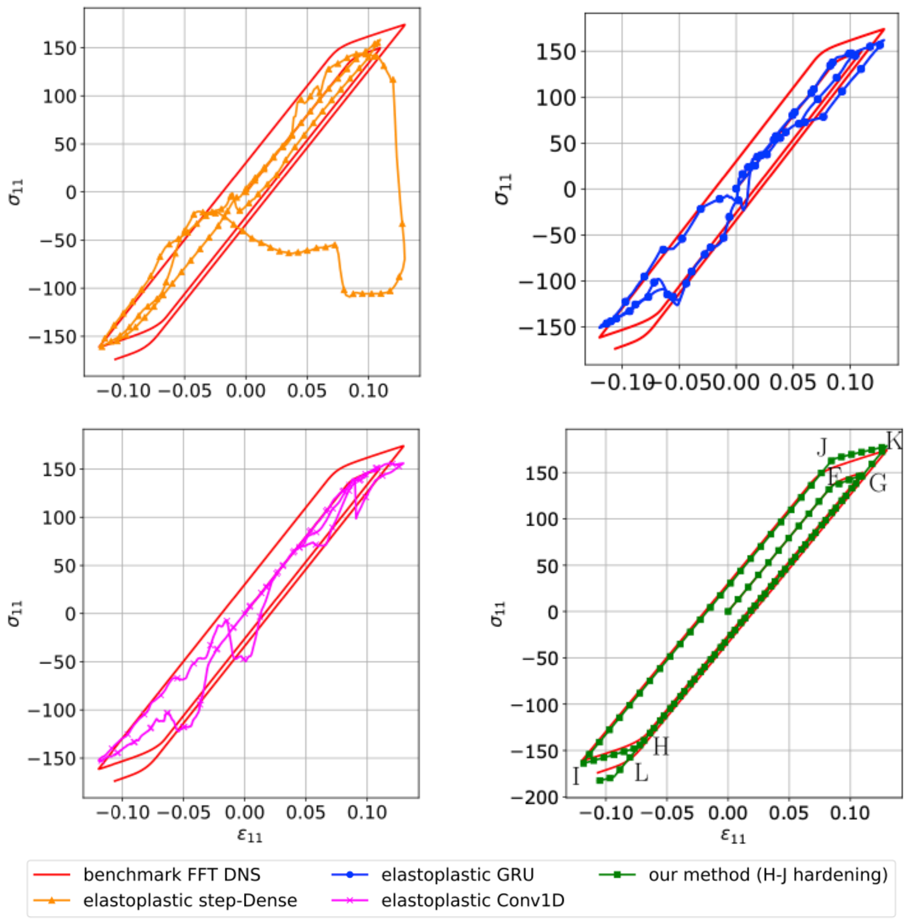
$$a_{AB} := \frac{\partial \sigma_A}{\partial \epsilon^{\mathrm{etr}}_{B}} = \sum_{C=1}^{3} \frac{\partial \sigma_A}{\partial \epsilon^{\mathrm{e}}_{C}} \frac{\partial \epsilon^{\mathrm{e}}_{C}}{\partial \epsilon^{\mathrm{etr}}_{B}} = \sum_{C=1}^{3} \boxed{\left(\frac{\partial^2 \widehat{\psi}^{\mathrm{e}}}{\partial \epsilon^{\mathrm{e}}_{A} \partial \epsilon^{\mathrm{e}}_{C}}\right)} \boxed{\frac{\partial \epsilon^{\mathrm{e}}_{C}}{\partial \epsilon^{\mathrm{etr}}_{B}}}$$

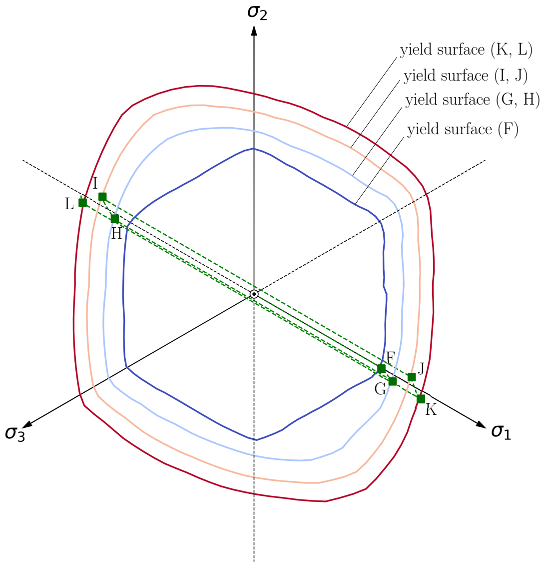
Results for unseen cyclic loading data



- A **classical machine learning** approach to predict **path-dependent elastoplasticity** behaviors uses **recurrent architectures** that are usually **black-box** and fail to predict unseen unloading paths
- We leverage classical plasticity theory to make **interpretable predictions even on unseen loading paths**.

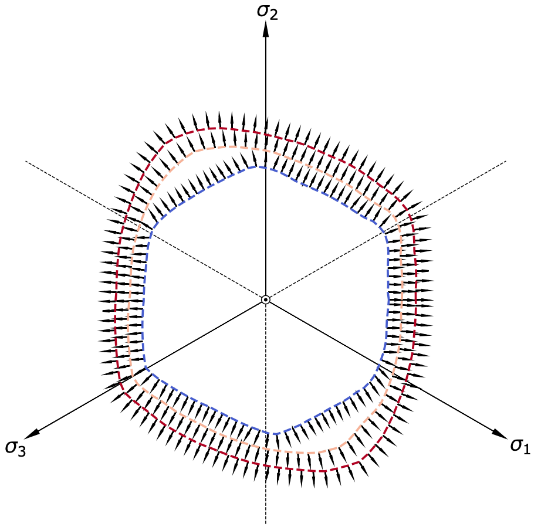
Results for unseen cyclic loading data

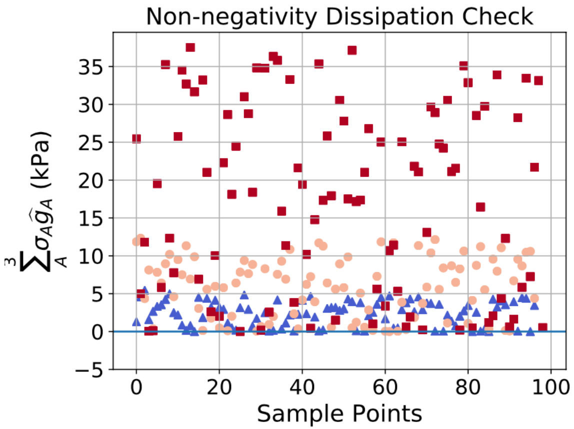




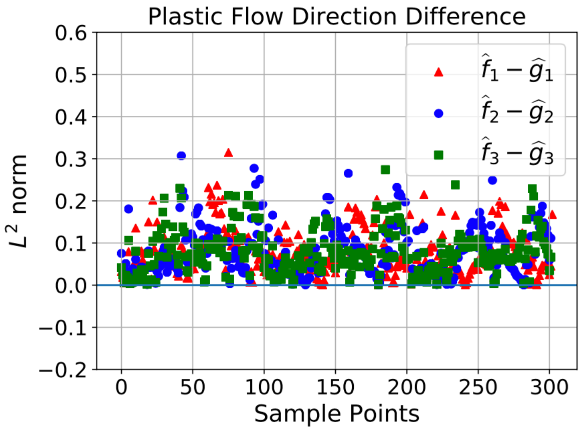
- Our elastoplastic framework can already **predict cyclic loading path** only **trained on monolithic data.**

ML-predicted dissipation and plastic flow direction





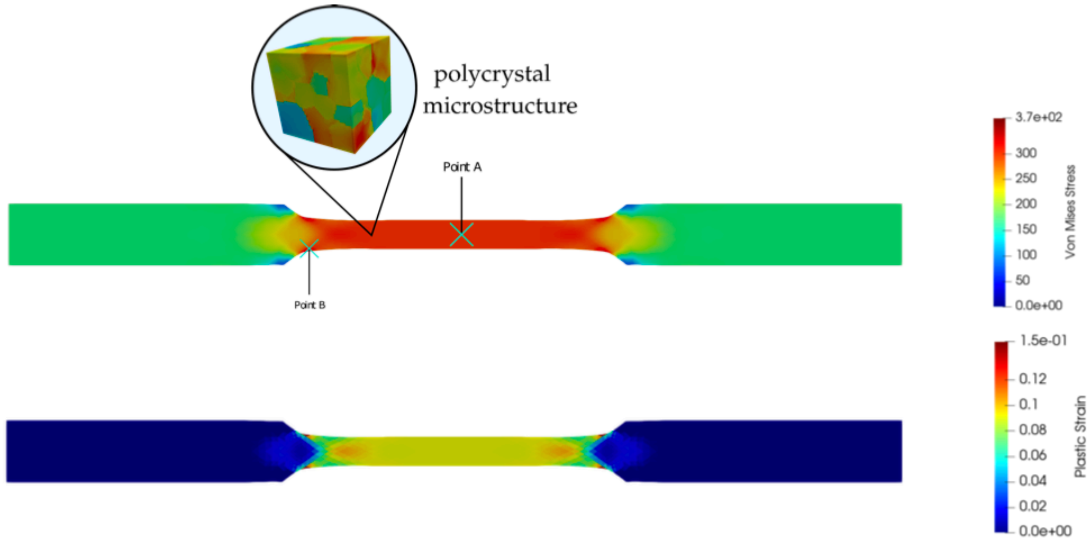
(a)



(b)

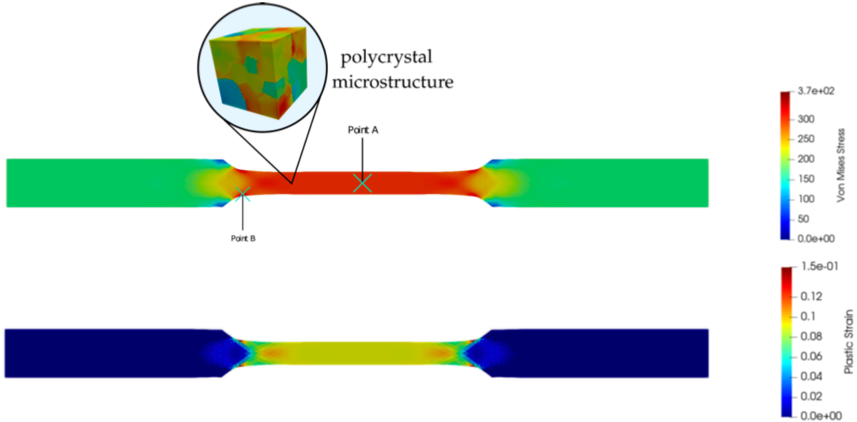
Vlassis & Sun, CMAME 2021

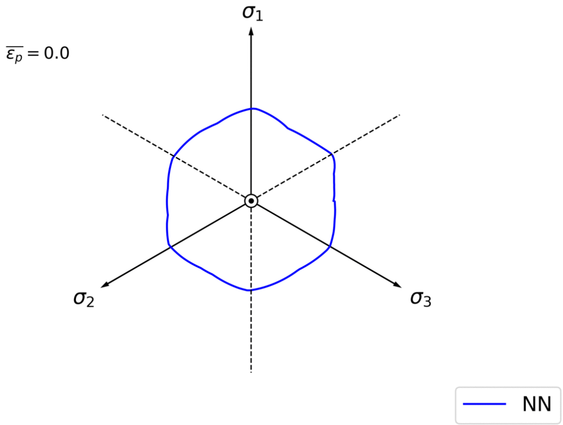
Elastoplasticity NN Framework – Polycrystal Plasticity Benchmark

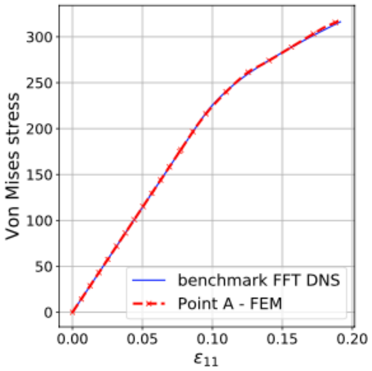


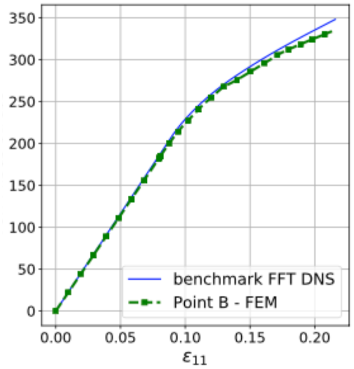
Our framework can **be readily implemented in FEM simulations**

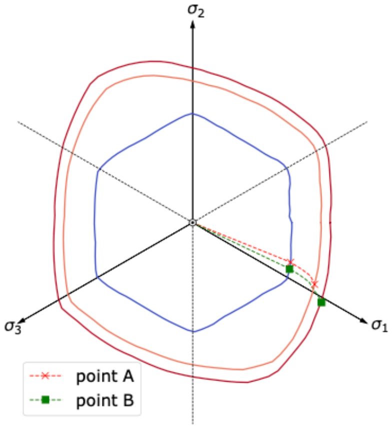
Elastoplasticity NN Framework – Polycrystal Plasticity Benchmark





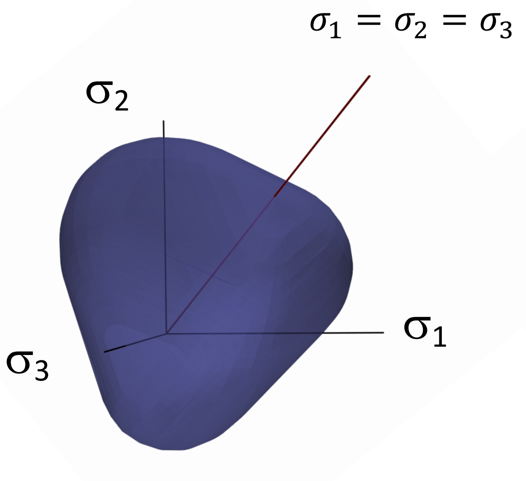


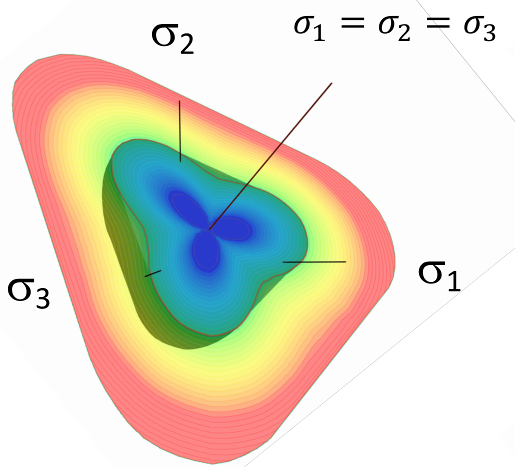


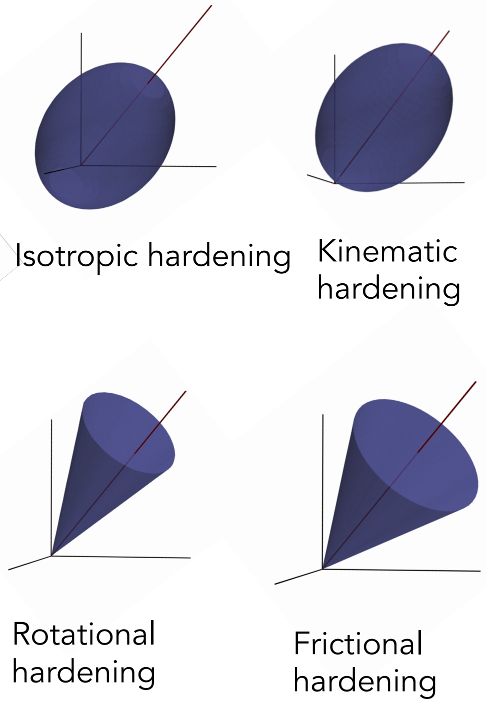


- **Polycrystal yield function** discovered from data – no need for complex yield function shape descriptors.
- Neural networks can **fully replace the elastoplastic constitutive model** – replace heavy FFT simulations at every material point.

Benchmark Study: Predicting hardening/softening mechanism for pressure-dependent materials via ONE unified level set model

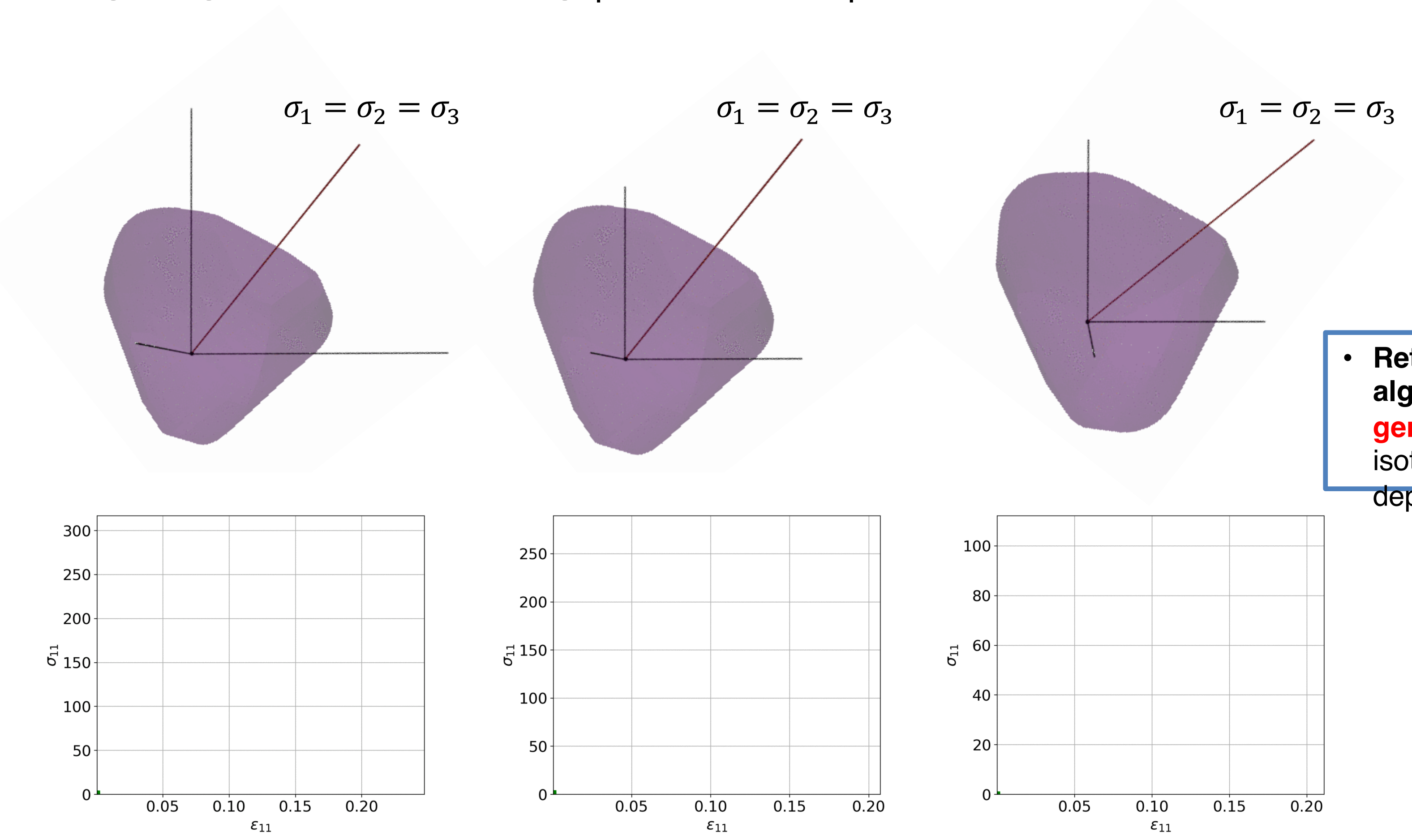






- Algorithm is readily **generalizable for pressure dependent models**
- Data-driven formulation any capture any form of hardening (isotropic, kinematic – change of **size** / **shape** / **translation** / **rotation** of yield surface in 3D stress space)

Ongoing work: Emulating pressure-dependent models

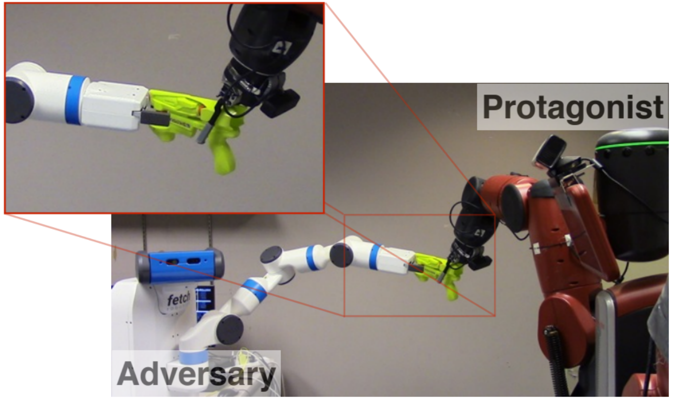


- **Return mapping algorithm** currently **generalizable** for isotropic pressure-dependent models

What data do we need to calibrate and validate the ML-generated constitutive theories?

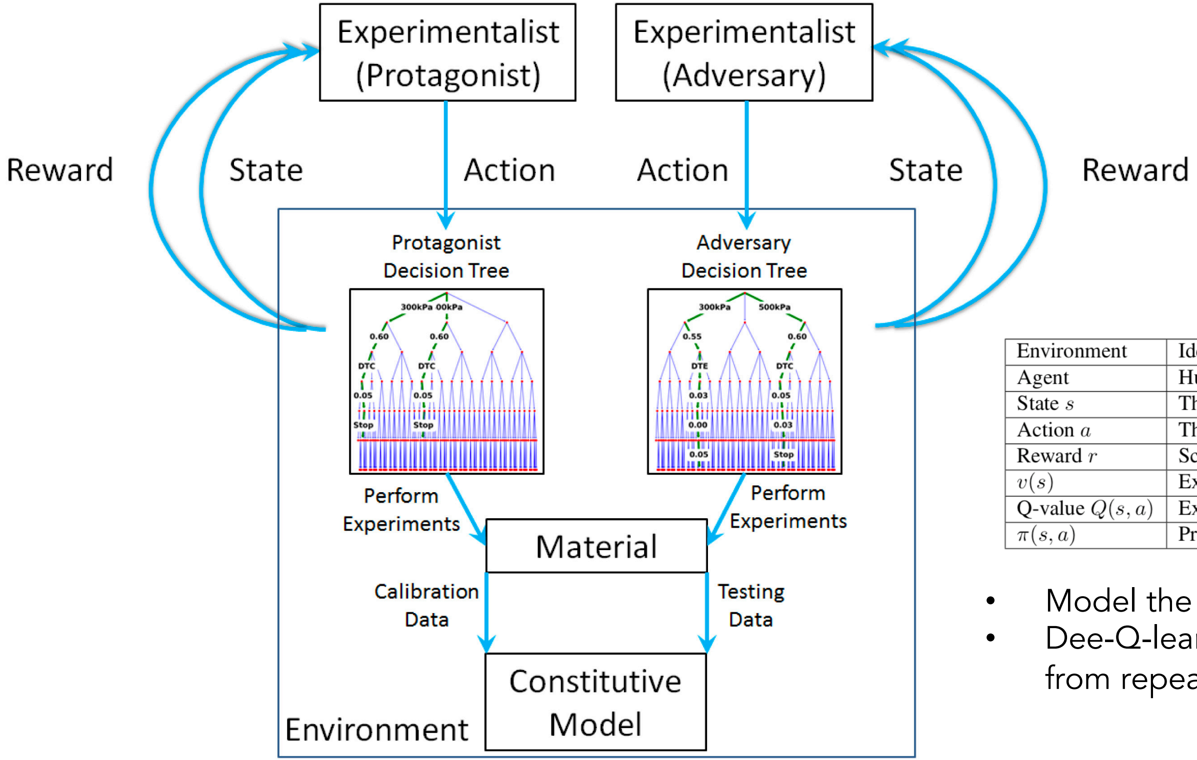
Adversarial deep reinforcement Learning

Example of adversarial learning:
Adversarial framework for effective self-supervised learning on grasp policy in robotics



Pinto, Lerrel, James Davidson, and Abhinav Gupta. "Supervision via competition: Robot adversaries for learning tasks." *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

Two-Agent non-cooperative game for validation/falsifying models



Environment	Idealized multigraph for constitutive models validated against unseen data
Agent	Human or AI
State s	The generated constitutive laws
Action a	The decisions that lead to the generation of constitutive laws
Reward r	Score (objective function) of the constitutive model
$v(s)$	Expected model score of state s
Q-value $Q(s, a)$	Expected model score from taking action a at state s
$\pi(s, a)$	Probability of taking action a at state s

- Model the action of two experimentalists as a game
- Dee-Q-learning creates AI to play the game and learn from repeating generating models automatically

Wang, K., Sun, W., & Du, Q. (2020). A non-cooperative meta-modeling game for automated third-party calibrating, validating and falsifying constitutive laws with parallelized adversarial attacks. Computer Methods in Applied Mechanics and Engineering, 373, 113514.

Two-Agent non-cooperative game for validation/falsifying models

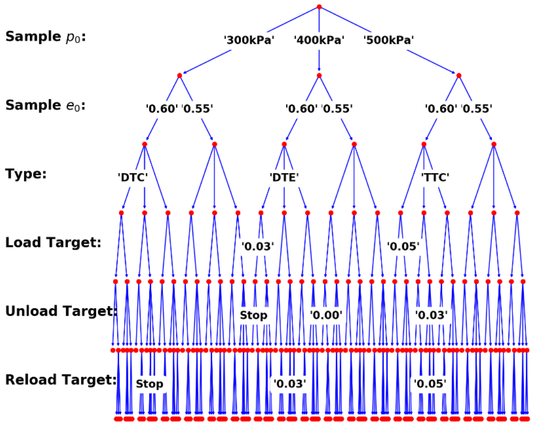
1. A non-cooperative game is run for two agents
   - Agent 1 is tasked with generating new experimental data to calibrate a model.
   - Agent 2 try to undermine the calibration effort of Agent 1 by finding the tests that maximize the calibration errors

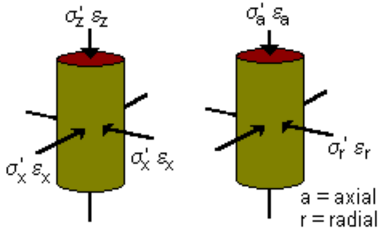
2. Material: DEM Representative Volume Element

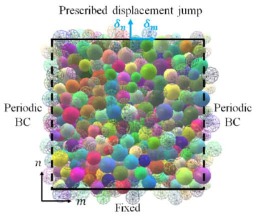
3. Game choices for experimentalist and adversary agents:
   1. Triaxial Compression/Extension/True Triaxial Tests (DTC, DTE, TTC)

   1. Loading/Unloading/Reloading Paths

1. The same decision Tree available for the agents





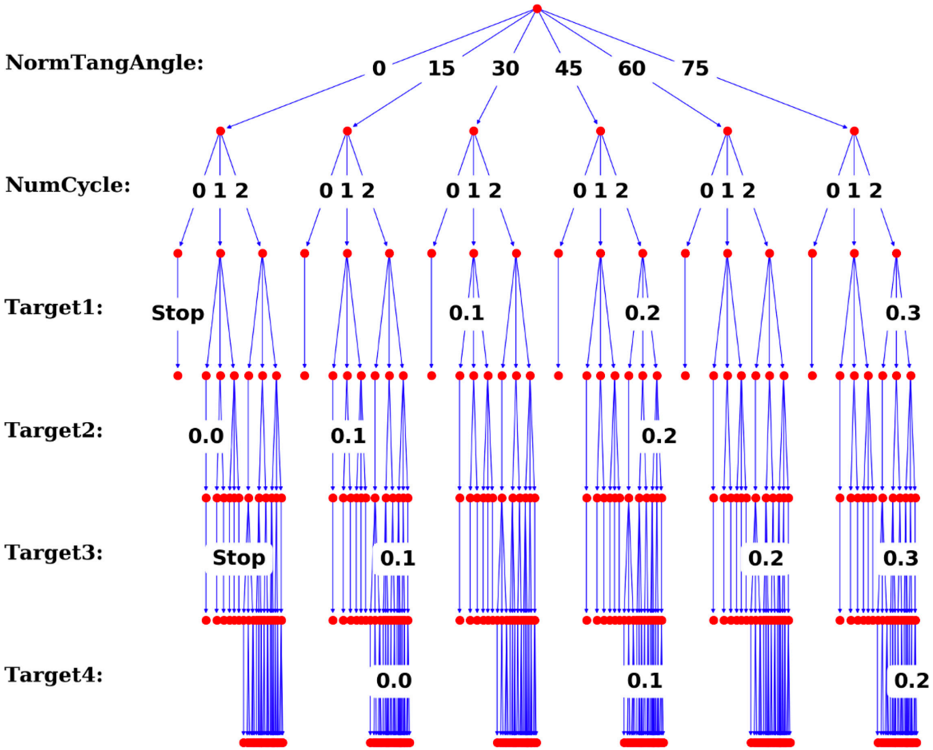


A non-cooperative meta-modeling game for automated third-party training, validating, and falsifying constitutive laws with adversarial attacks, CMAME, 2020.

Game Action for the non-cooperative game – Making choices for an experiment

Example: Traction-Separation Law

$\mathbb{TC}$ test conditions	Choices
$\mathbb{TC}_1$ = 'NormTangAngle'	{'0', '15', '30', '45', '60', '75'}
$\mathbb{TC}_2$ = 'NumCycle'	{'0', '1', '2'}
$\mathbb{TC}_3$ = 'Target1'	{'NaN', '0.1', '0.2', '0.3'}
$\mathbb{TC}_4$ = 'Target2'	{'NaN', '0.0', '0.1', '0.2'}
$\mathbb{TC}_5$ = 'Target3'	{'NaN', '0.1', '0.2', '0.3'}
$\mathbb{TC}_6$ = 'Target4'	{'NaN', '0.0', '0.1', '0.2'}



A non-cooperative meta-modeling game for automated third-party training, validating, and falsifying constitutive laws with adversarial attacks, CMAME, 2020.

Game reward for the non-cooperative game

- Efficiency Index (Nash-Sutcliffe)

$$E_{NS}^{j} = 1 - \frac{\sum_{i=1}^{N_{data}} |\overline{Y}_i^{\text{data}} - \overline{Y}_i^{\text{model}}|^j}{\sum_{i=1}^{N_{data}} |\overline{Y}_i^{\text{data}} - \text{mean}(\overline{Y}^{\text{data}})|^j} \in (-\infty, 1.0].$$

1 – perfect match
0 – model as accurate as using the mean of observed data
< 0 when using the mean is more accurate than using the model

- Re-scaling the efficiency index to range -1 to 1

$$\text{SCORE}_{\text{protagonist or adversary}} = 2 * \frac{\min(\max(E_{NS}^{1}, E_{NS}^{min}), E_{NS}^{max}) - 0.5 * (E_{NS}^{min} + E_{NS}^{max})}{E_{NS}^{max} - E_{NS}^{min}},$$

- Competing reward

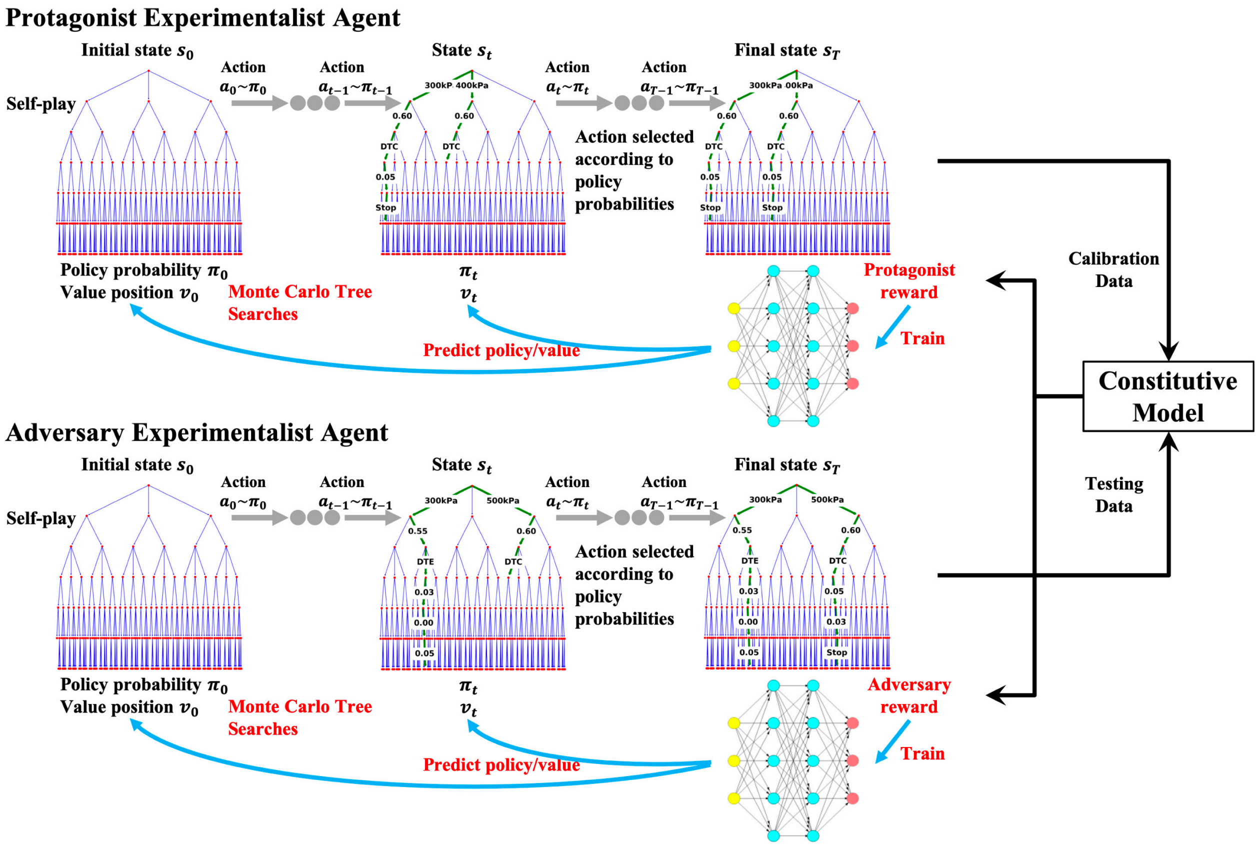
Decay coefficient Cut-off value Historically lowest EI

$$\text{Reward}_{\text{protagonist}} = -1 + (\text{SCORE}_{\text{protagonist}} + 1) * \exp\left[-\alpha_{\text{SCORE}} * \max(E_{NS}^{min} - \min(\{E_{NS}^{1}\}), 0)\right].$$

$$\text{Reward}_{\text{adversary}} = -\text{SCORE}_{\text{adversary}}.$$

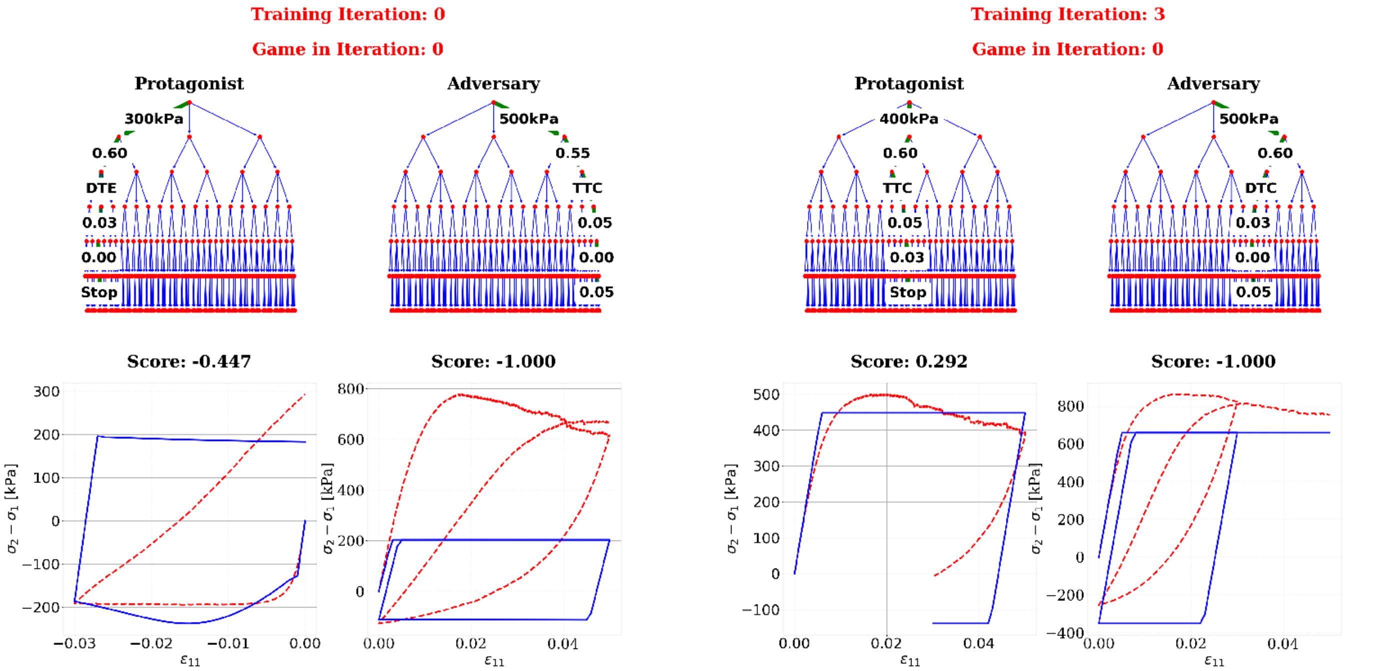
A non-cooperative meta-modeling game for automated third-party training, validating, and falsifying constitutive laws with adversarial attacks, CMAME, 2020.

Game Play for the non-cooperative game



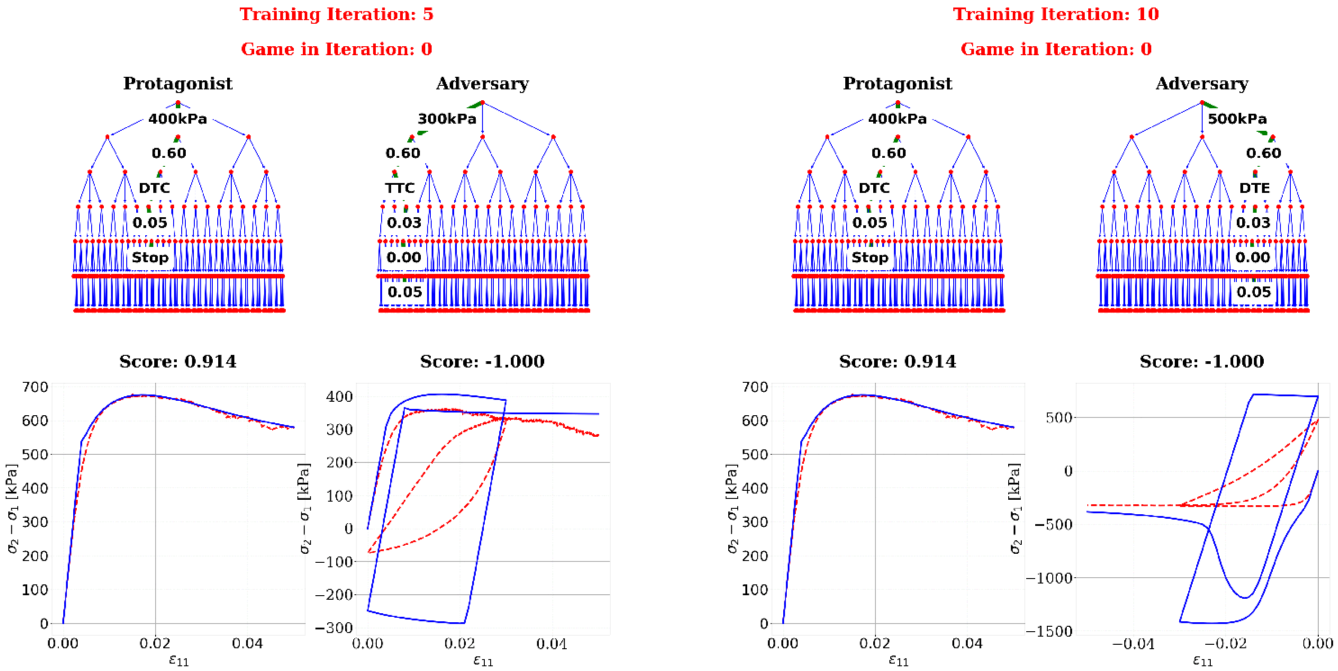
A non-cooperative meta-modeling game for automated third-party training, validating, and falsifying constitutive laws with adversarial attacks, CMAME, 2020.

Reinforcement learning performance of the experimentalist/adversary game (Drucker-Prager)



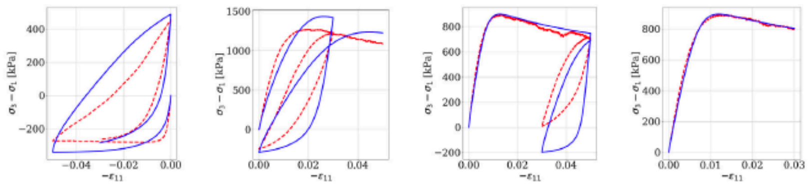
Initially, both agents are exploring the parametric space and attempt to improve their estimated Q values through interacting with each others.

Reinforcement learning performance of the experimentalist/adversary game (Drucker Prager)



As the game progress, both agents have generated sufficient knowledge such that the Q table converges – the calibration agent tells you the strength of the models and where the Drucker-Prager model scores the best (monotonic triaxial compression), while the adversarial agent found that the DP model is not suitable to predict cyclic responses of DEM RVE.

Reinforcement learning performance of the experimentalist/adversary game (Bounding surface plasticity model)

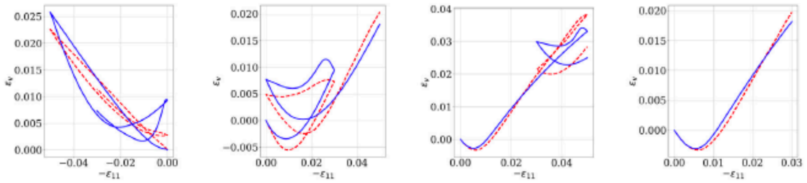


(a) Iteration 0, Episode 24, Defense Game Score: 0.162

(b) Iteration 3, Episode 33, Defense Game Score: 0.626

(c) Iteration 6, Episode 10, Defense Game Score: 0.687

(d) Iteration 10, Episode 30, Defense Game Score: 0.912



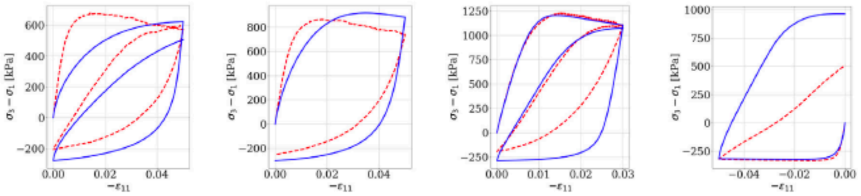
(e) Iteration 0, Episode 24, Defense Game Score: 0.162

(f) Iteration 3, Episode 33, Defense Game Score: 0.626

(g) Iteration 6, Episode 10, Defense Game Score: 0.687

(h) Iteration 10, Episode 30, Defense Game Score: 0.912

Defense experimentalist + model calibrator

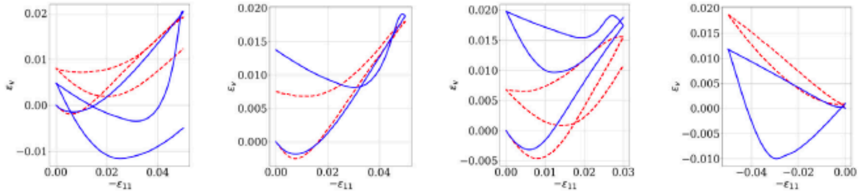


(a) Iteration 0, Episode 24, Attack Game Score: 0.206

(b) Iteration 3, Episode 33, Attack Game Score: 0.323

(c) Iteration 6, Episode 10, Attack Game Score: 0.295

(d) Iteration 10, Episode 30, Attack Game Score: -0.290



(e) Iteration 0, Episode 24, Attack Game Score: 0.206

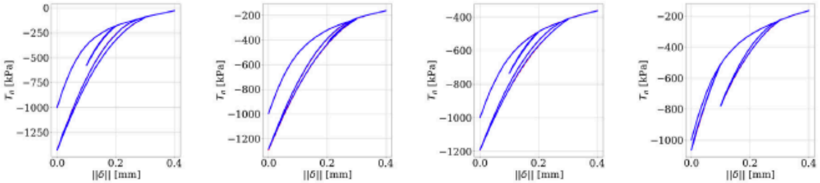
(f) Iteration 3, Episode 33, Attack Game Score: 0.323

(g) Iteration 6, Episode 10, Attack Game Score: 0.295

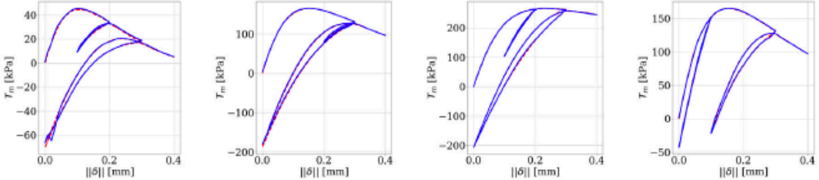
(h) Iteration 10, Episode 30, Attack Game Score: -0.290

Attack experimentalist

Reinforcement learning performance of the experimentalist/adversary game (ML Traction- separation model)

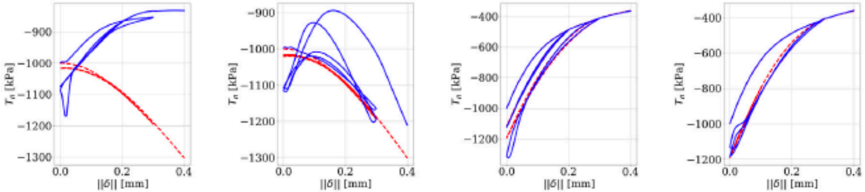


(a) Iteration 0, Episode 11, Defense Game Score: -0.992
(b) Iteration 3, Episode 7, Defense Game Score: -0.331
(c) Iteration 6, Episode 31, Defense Game Score: 0.269
(d) Iteration 10, Episode 20, Defense Game Score: 0.879

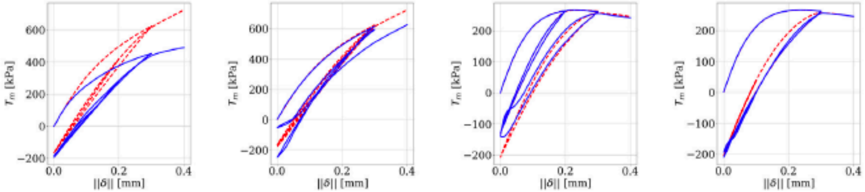


(e) Iteration 0, Episode 11, Defense Game Score: -0.992
(f) Iteration 3, Episode 7, Defense Game Score: -0.331
(g) Iteration 6, Episode 31, Defense Game Score: 0.269
(h) Iteration 10, Episode 20, Defense Game Score: 0.879

Defense experimentalist + model calibrator



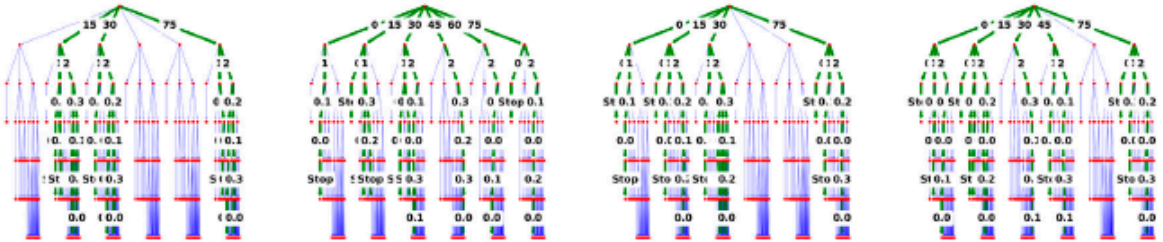
(a) Iteration 0, Episode 11, Attack Game Score: -0.085
(b) Iteration 3, Episode 7, Attack Game Score: -0.127
(c) Iteration 6, Episode 31, Attack Game Score: 0.540
(d) Iteration 10, Episode 20, Attack Game Score: 0.443



(e) Iteration 0, Episode 11, Attack Game Score: -0.085
(f) Iteration 3, Episode 7, Attack Game Score: -0.127
(g) Iteration 6, Episode 31, Attack Game Score: 0.540
(h) Iteration 10, Episode 20, Attack Game Score: 0.443

Attack experimentalist

Evolution of the estimated policy value



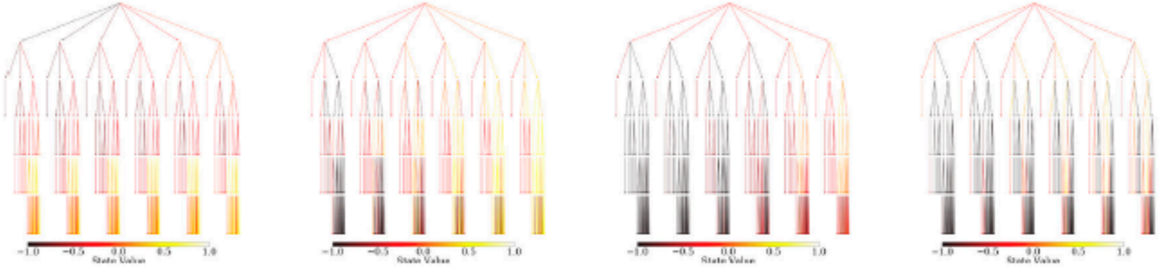
(a) Iteration 0, Episode 11, Defense Game Score: -0.992

(b) Iteration 3, Episode 7, Defense Game Score: -0.331

(c) Iteration 6, Episode 31, Defense Game Score: 0.269

(d) Iteration 10, Episode 20, Defense Game Score: 0.879

Fig. 21. Examples of paths (experiments) in the decision trees selected by the protagonist during the DRL training iterations for the traction–separation model.



(a) Iteration 0

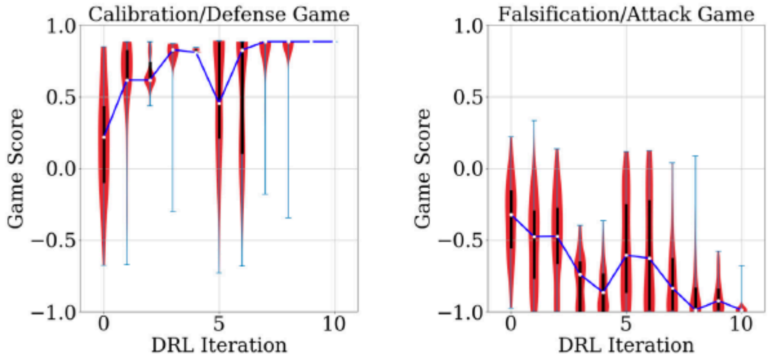
(b) Iteration 3

(c) Iteration 6

(d) Iteration 10

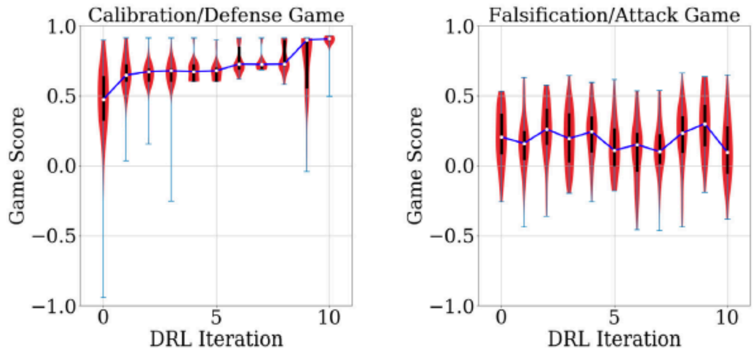
Fig. 22. Examples of Q-values of all possible states in the experimental decision tree estimated by the protagonist's policy/value network f_θ during the DRL training iterations for the traction–separation model.

Results of competitions



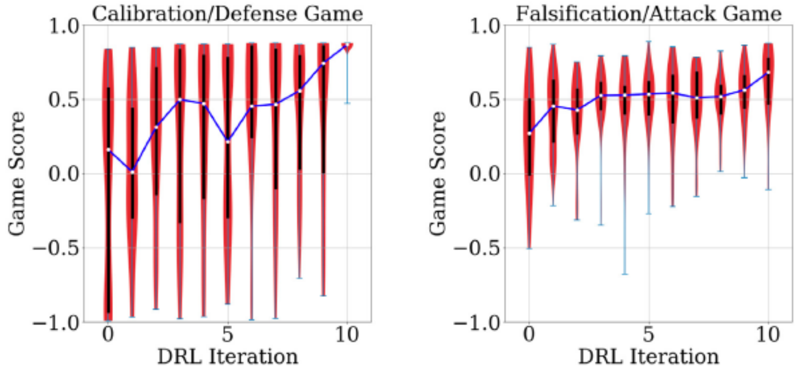
(a) Protagonist (b) Adversary

Drucker-Prager Model



(a) Protagonist (b) Adversary

Bounding-surface critical state plasticity



ML-generated traction separation law

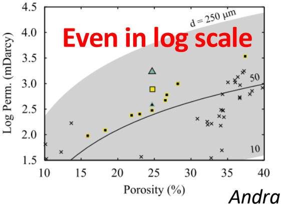
Related work 2: An accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data (with Bahador Bahmani)

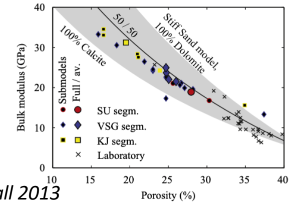
Some challenges in the data-driven model free approach:

- **Data hungry** (less assumption higher need for data)
- **Scalability** (online search over many data-points)

Observation:

- Solid constitutive models: high fidelity
- **Fluid** constitutive models: **low fidelity**



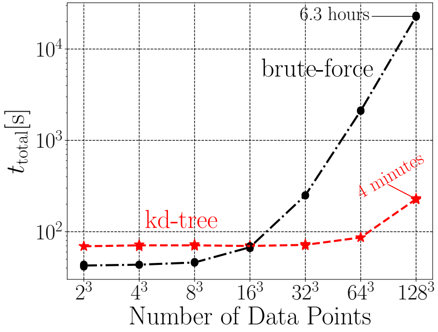


Andra et. all 2013

Our contribution:

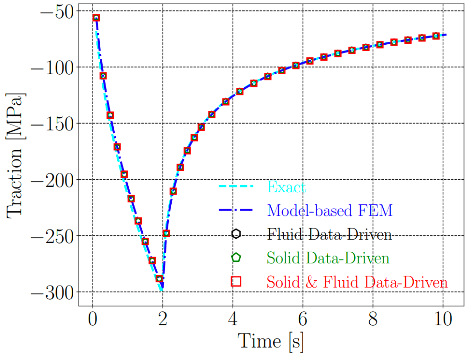
- We introduce for the first time a coupled multiphysics model-free formulation based on a non-parametric data-driven algorithm for poroelasticity applications
- Two hybrid formulations are developed in cases where model (human-written or surrogate) performs satisfactory. (**Data efficient**)
- We introduce a simple projection that maps the energy metric space onto the Euclidean metric space. Using this treatment, high dimensional data can be efficiently stored in tree data structures for fast nearest neighbor search. (**Scalable**)

Exponentially faster model-free algorithm

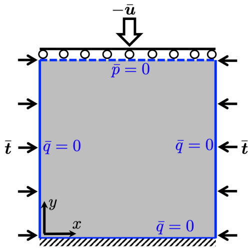


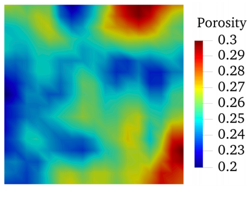
Two order of magnitude!

Validation of the developed model-free formulations



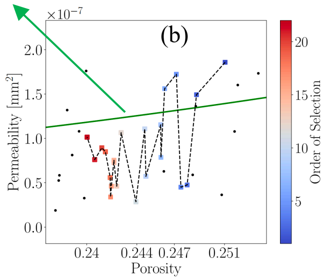
Hybrid model free solution:
Nonlinear heterogenous problem with real data from FFT simulation and sandstone 3D images

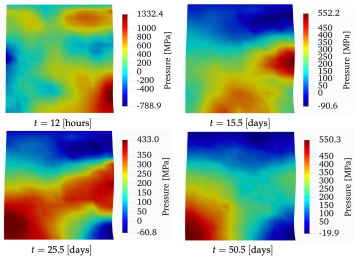




Initial heterogeneous porosity

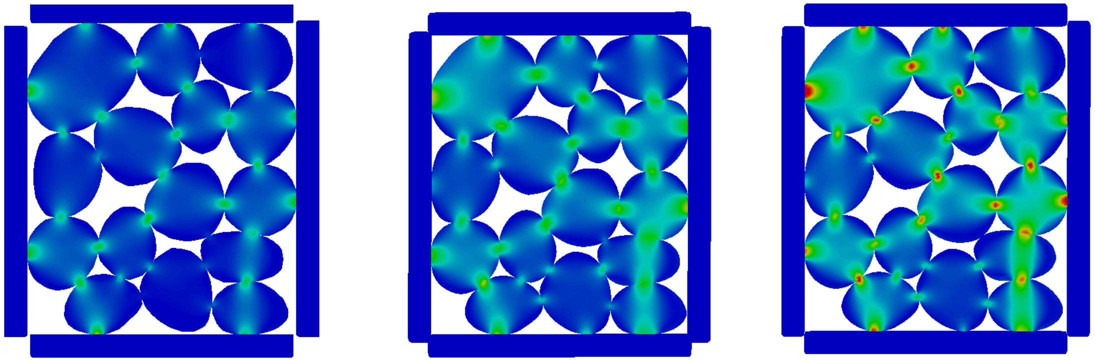
Kozeny-Carman fitted

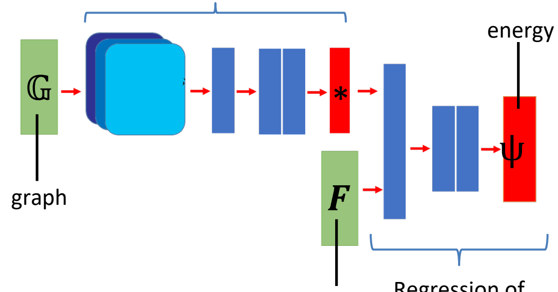


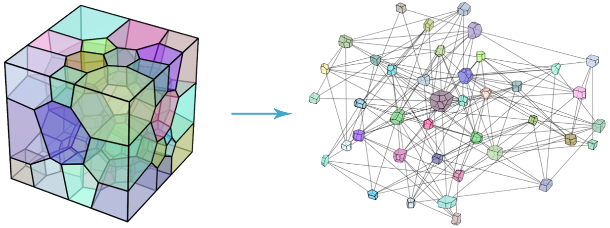


Future Work: Geometric learning for evolving connectivity graphs

➢ Stress evolutions under various loadings (grain scale)







Creating low-dimensional representation graph to represent microstructures from voxel images

Vlassis, Ma & Sun, CMAME 2020

(For convolutional neural network on voxel images, see Frankel et al, CMS 2019)

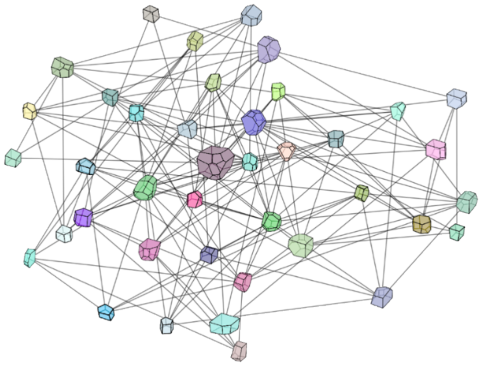
C. Liu", **W.C. Sun**, ILS-MPM: an unbiased implicit level-set-based material point method for frictional particulate contact mechanics of deformable particles, *Computer Methods in Applied Mechanics and Engineering*, , doi:10.1016/j.cma.2020.113168, 2020.

Concluding Remarks:

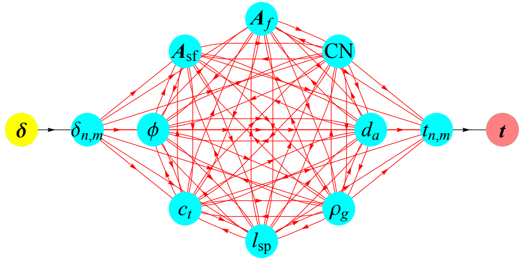
We examine some potential applications of undirected weighted graph and directed graph for computational mechanics, in particular, we introduce ways to

1. Generalized the modeling process of elasto-plasticity problems.

2. Write, validate and falsify a constitutive law represented by directed graph via non-cooperative game.



Undirected weighted graph



Directed Multi-graph

Acknowledgements

- Army Research Office: Young Investigator Program Award
- Air Force Office of Scientific Research, Young Investigator Program Award
- National Science Foundation: EAR-1516300 and CMMI-1445033, NSF CAREER
- Sandia National Laboratories
- US Department of Energy Office of Nuclear Energy
- Columbia University

Reference

1. K. Wang, W.C. Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, *Computer Methods in Applied Mechanics and Engineering,* 334(1):337-380, doi:10.1016/j.cma.2018.01.036, 2018. [PDF][Bibtex]

2. K. Wang, W.C. Sun, An updated Lagrangian LBM-DEM-FEM coupling model for dual-permeability porous media with embedded discontinuities, *Computer Methods in Applied Mechanics and Engineering*, 344:276-305, doi:10.1016/j.cma.2018.09.034, 2019.

3. K. Wang, W.C. Sun, Meta-modeling game for deriving theory-consistent, micro-structure-based traction-separation laws via deep reinforcement learning, *Computer Methods in Applied Mechanics and Engineering,* 346:216-241, doi:10.1016/j.cma.2018.11.026, 2019. [PDF][Bibtex][Data]

4. K. Wang, W.C. Sun, Q. Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation, *Computational Mechanics*, special issue for Data-Driven Modeling and Simulations: Theory, Methods and Applications, 64(2):67–499, doi:,10.1007/s00466-019-01723-1, 2019. [PDF]

5. Y. Heider, K. Wang, W.C. Sun, SO(3)-invariance of graph-based deep neural network for anisotropic elastoplastic materials, *Computer Methods in Applied Mechanics and Engineering,* 363:112875, doi:10.1016/j.cma.2020.112875, 2020. [PDF]

6. C. Liu, W.C. Sun, ILS-MPM: an unbiased implicit level-set-based material point method for frictional particulate contact mechanics of deformable particles, *Computer Methods in Applied Mechanics and Engineering,* accepted, 2020. [PDF]

7. R. Ma, W.C. Sun, Computational thermomechanics for crystalline rock. Part II: chemo-damage-plasticity and healing in strongly anisotropic polycrystals, *Computer Methods in Applied Mechanics and Engineering,* doi:10.1016/j.cma.2020.113184, 2020. [PDF]

8. K. Wang, W.C. Sun, Q. Du, A non-cooperative meta-modeling game for automated third-party training, validating, and falsifying constitutive laws with adversarial attacks, Computer Methods in Applied Mechanics and Engineering, doi:10.1016/j.cma.2020.113514, 2020. [Video]

9. N. Vlassis, R. Ma, W.C. Sun, Geometric deep learning for computational mechanics Part I: Anisotropic Hyperelasticity, Computer Methods in Applied Mechanics and Engineering, 2020

10. A. Fuchs, Y. Heider, K. Wang, W.C. Sun, M. Kaliske, DNN^2: A hyper-parameter reinforcement learning game for self-design neural network elasto-plastic constitutive laws, under review.

11. N. Vlassis, W.C. Sun, Sobolev training of thermodynamic-informed neural network for smoothed elasto-plasticity models with level set hardening, Computer Methods in Applied Mechanics and Engineering, accepted, 2021. [Video]

ICAM conference: mmldt.eng.ucsd.edu

PHOTO: DARCY PETERKA

Thank You!

More information can be found at

www.poromehanics.org