A non-cooperative game approach for multiscale predictive modeling of path-dependent porous materials.

WaiChing "Steve" Sun

Department of Civil Engineering and Engineering Mechanics,

Fu Foundation School of Engineering and Applied Science,

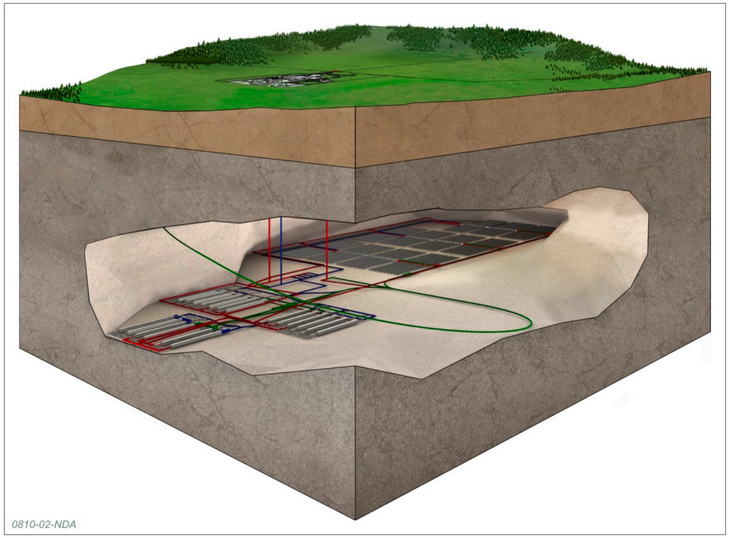
Columbia University, New York, USA.

Theoretical and Computational Poromechanics Laboratory

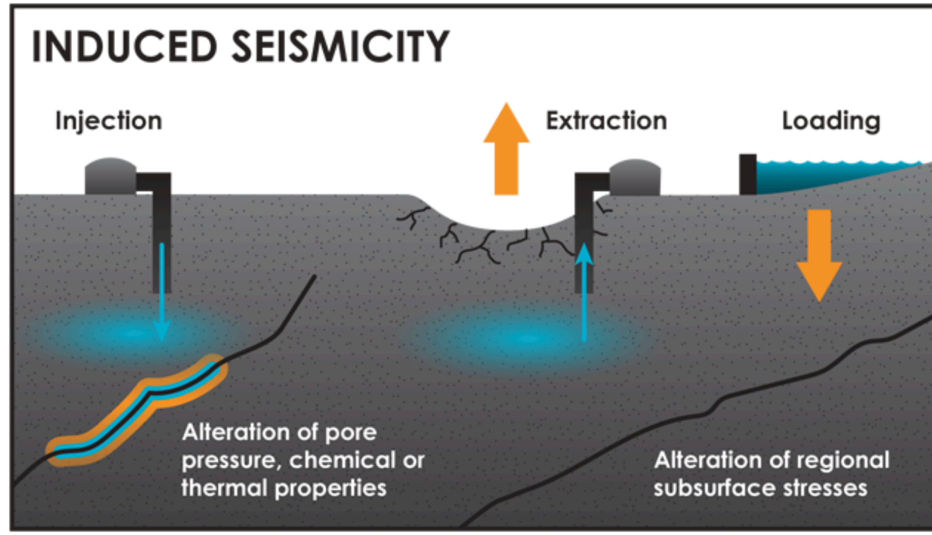
The Fu Foundation School of Engineering and Applied Science

Columbia University in the city of New York

Motivation & Background



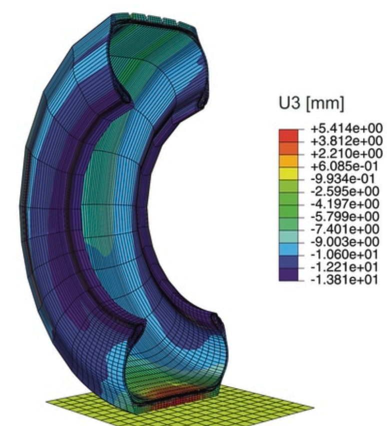
Geological disposal of nuclear waste



Induced Seismicity due to hydraulic fracture, mining, CO2 storage…etc

Artificial ground freezing

Mixing granular materials with moisture content



a)

Tire-soil interaction and off-road mobility

Why Machine Learning?

Multi-scale multi-porosity hydro-mechanical problem

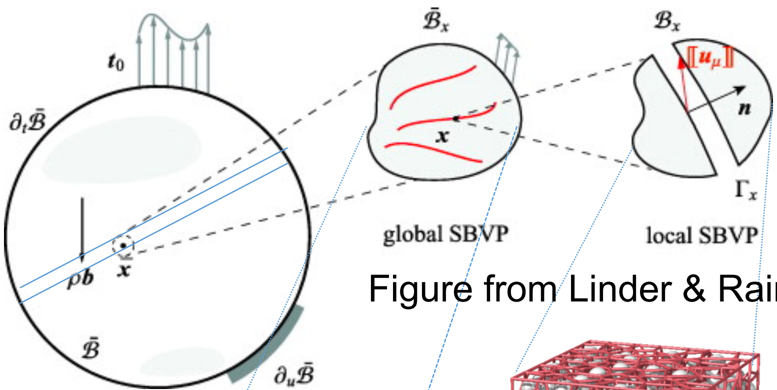
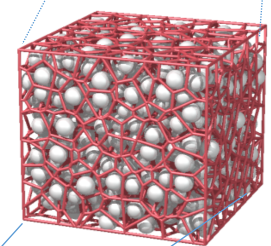


Figure from Linder & Raina, 2013

Wang & Sun, CMAME, 2018



Balance of linear momentum (cf. Borja & Choo, CMAME, 2016)

$$\nabla^X \cdot \boldsymbol{P} + \rho_0 \boldsymbol{g} = c_0(\tilde{\boldsymbol{v}}_m - \tilde{\boldsymbol{v}}_M)$$

Balance of fluid mass for macropore (fractured pore space)

$$\dot{\rho}_0^M + \nabla^X \cdot \boldsymbol{Q}_M = -c_0$$

$$\dot{\rho}_0^M = \overline{J\phi\psi\rho_f} \quad \boldsymbol{Q}_M = J\boldsymbol{F}^{-1} \cdot \boldsymbol{q}_M$$

Balance of fluid mass for micropore (pore matrix space)

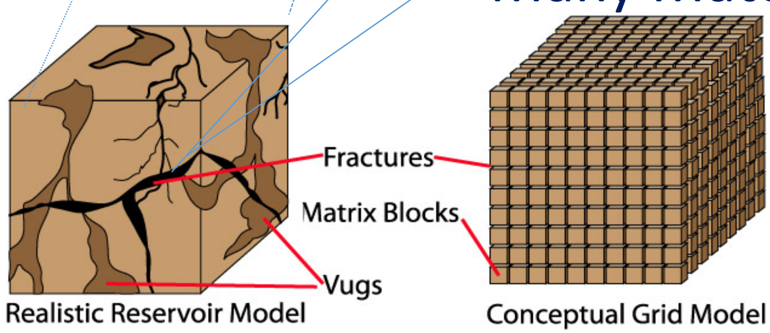
$$\dot{\rho}_0^m + \nabla^X \cdot \boldsymbol{Q}_m = c_0$$

$$\dot{\rho}_0^m = \overline{J\phi(1-\psi)\rho_f} \quad \boldsymbol{Q}_m = J\boldsymbol{F}^{-1} \cdot \boldsymbol{q}_m$$

Effective stress principle

$$\boldsymbol{\tau}' = \boldsymbol{\tau} + J\overline{p}\mathbf{1} = \boldsymbol{\tau} + J[\psi p_M + (1-\psi)p_m]\mathbf{1},$$

Many Material laws with interconnected relations!



Stress-strain & traction-separation laws

Flux - pressure gradient anisotropic relations

$$\boldsymbol{q}_M = -\rho_f \frac{\boldsymbol{k}_M}{\mu_f} \cdot (\nabla^x p_M - \rho_f \boldsymbol{g}),$$

$$\boldsymbol{q}_m = -\rho_f \frac{\boldsymbol{k}_m}{\mu_f} \cdot (\nabla^x p_m - \rho_f \boldsymbol{g}).$$

Online Multiscale homogenization
or Offline data-driven model
or Phenomenological models

Mass transfer coefficient between macropore and micropore

$$c_0 = Jc = J\frac{\bar{\alpha}}{\mu_f}(p_M - p_m).$$

Information flow in computational mecahnics solvers represented by directed graph

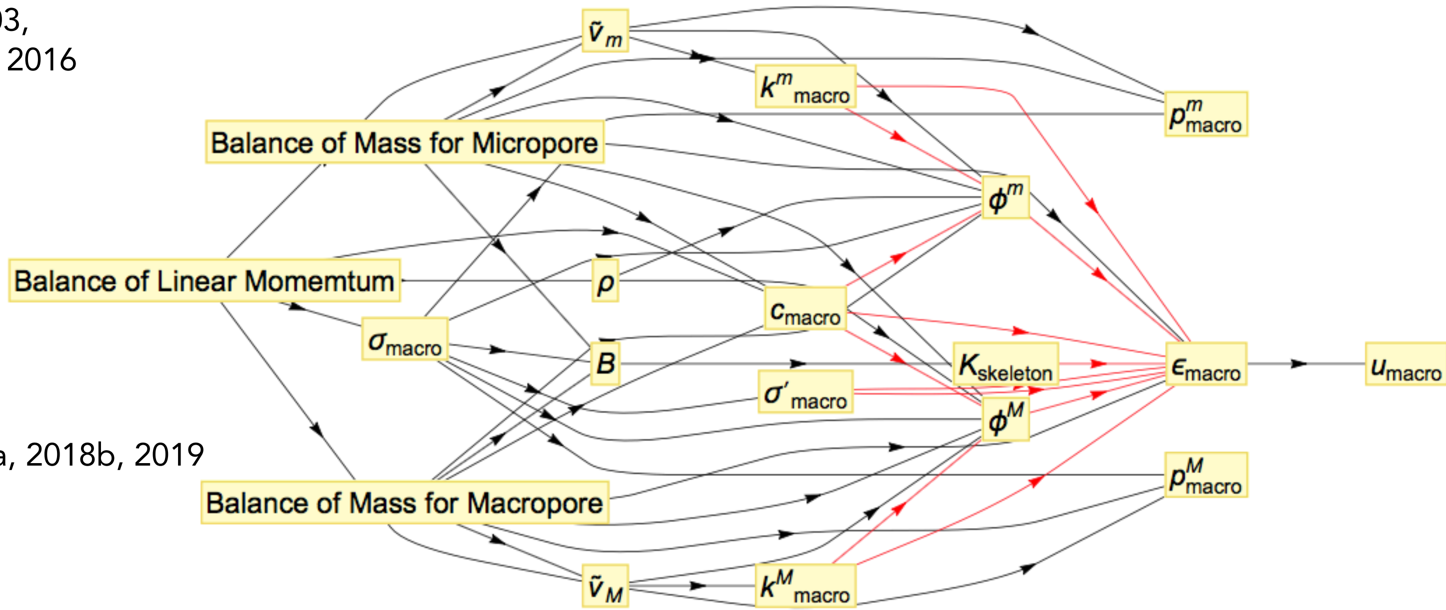
Single-physics case

Ghaboussi et al. 1991,
Lefik & Schrefler 2003,
Kirchdoerfer & Ortiz 2016

Multi-physics case

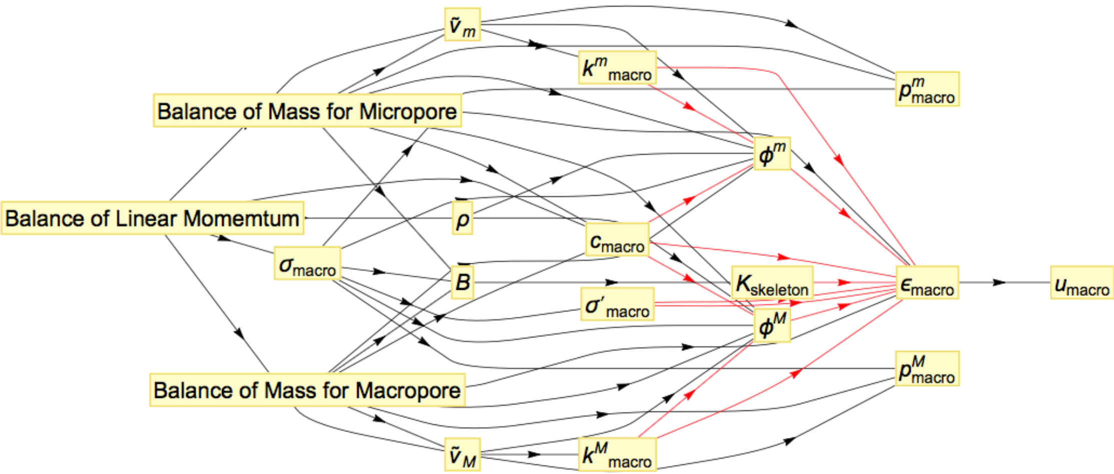
(THIS STUDY)

Wang & Sun, 2018a, 2018b, 2019

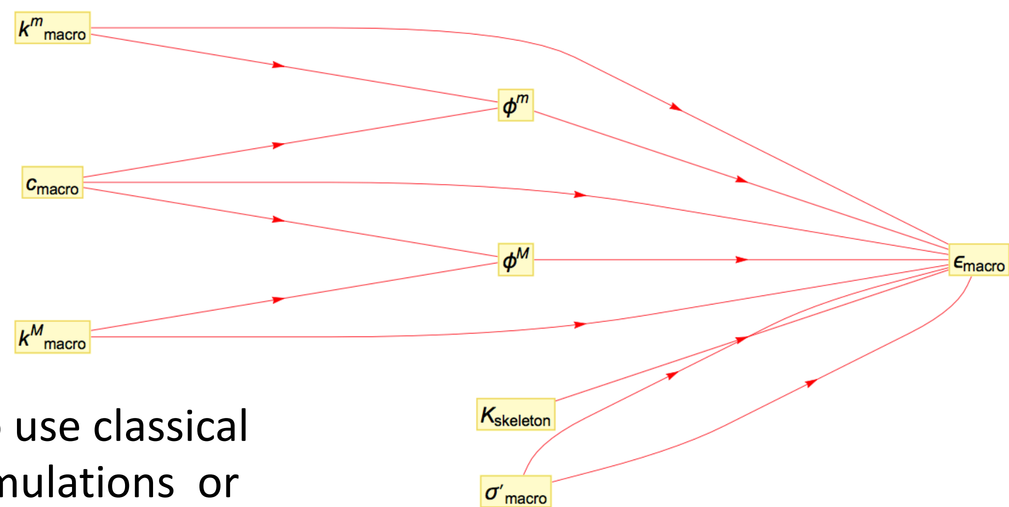


- Black arrows represent "definition" or "universal principle"
- Red arrows represent material laws
- Component-based PDE solver (cf. Sun et al. IJNAMG 2013, Sun, IJNME 2015 Salinger et al. IJMCE 2016)

Generate configurations of subgraphs

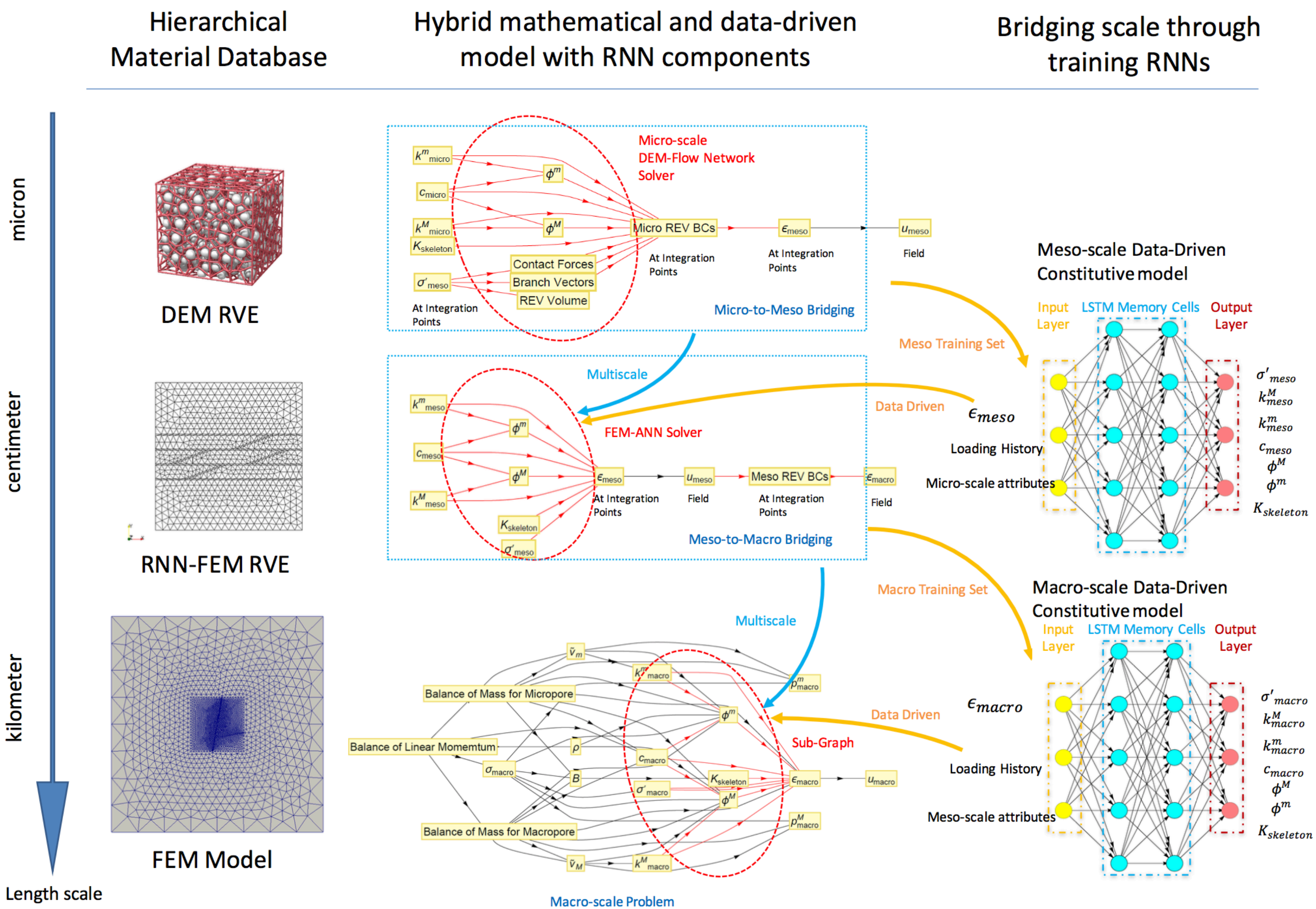


Extract subgraphs



In the subgraph, decide whether to use classical constitutive law, online subscale simulations or data-driven models for each edge

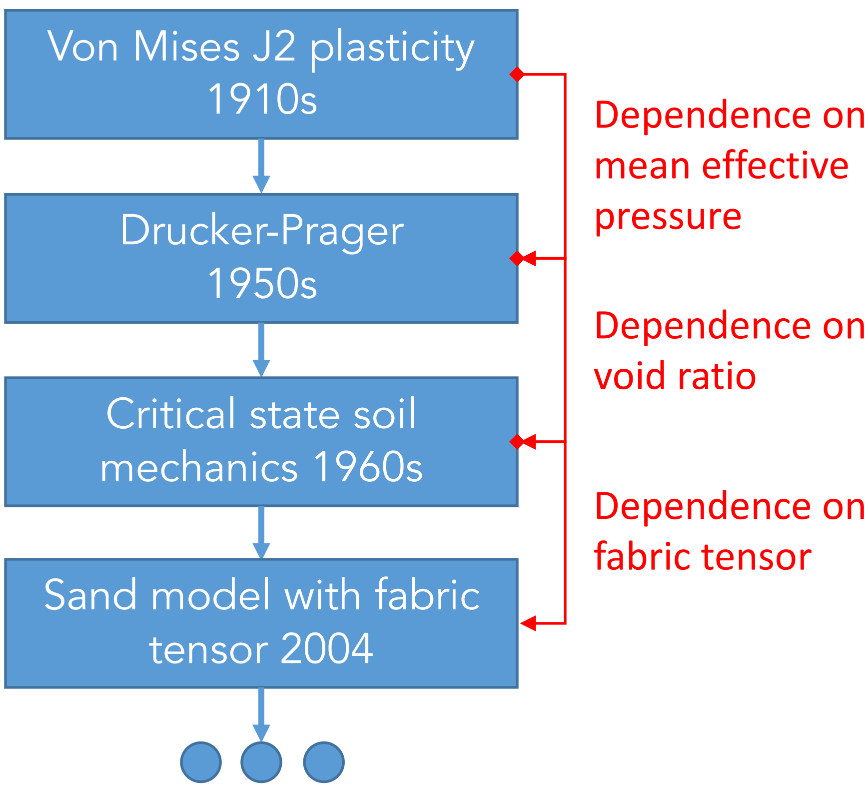
Recursive Deep Learning -- using neural network to train neural network

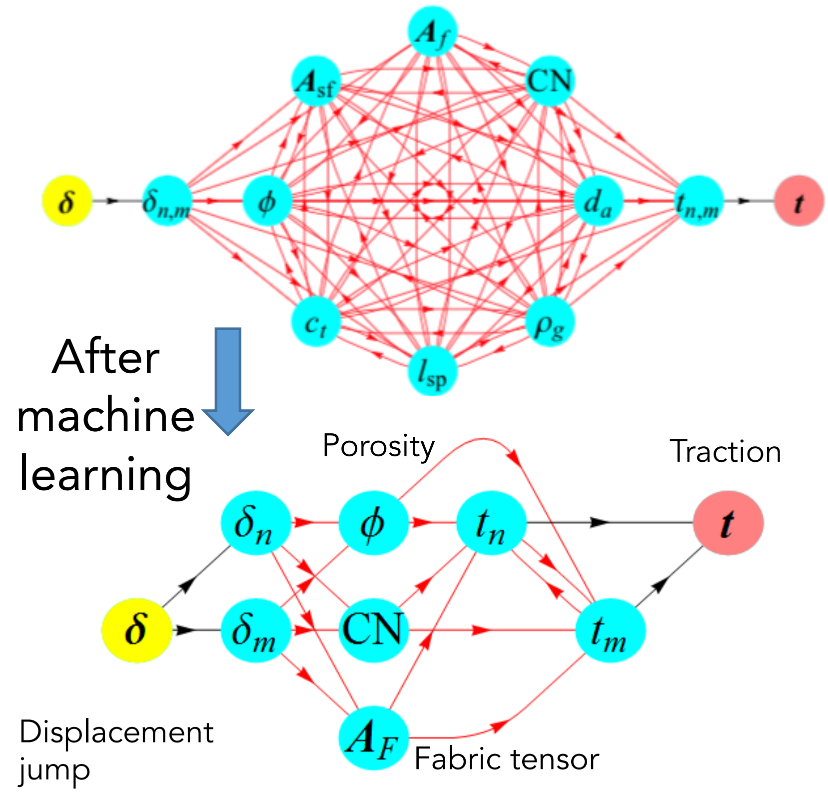


Wang & Sun, CMAME 2018

Abstraction of knowledge can be done via graph theory

How to accelerate scientific discovery using machine learning?





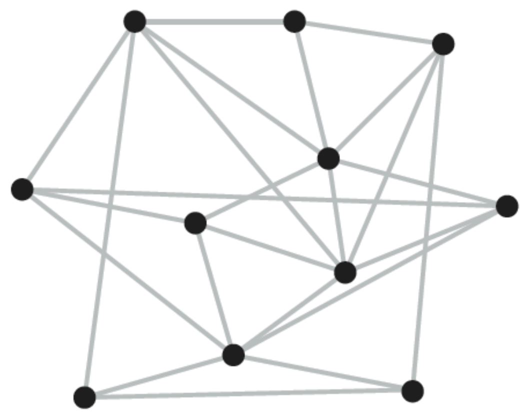
Discover new mechanisms

Each discovery relates to finding new mechanisms from data, which can be regarded as adding new nodes and new edges in the knowledge graph.

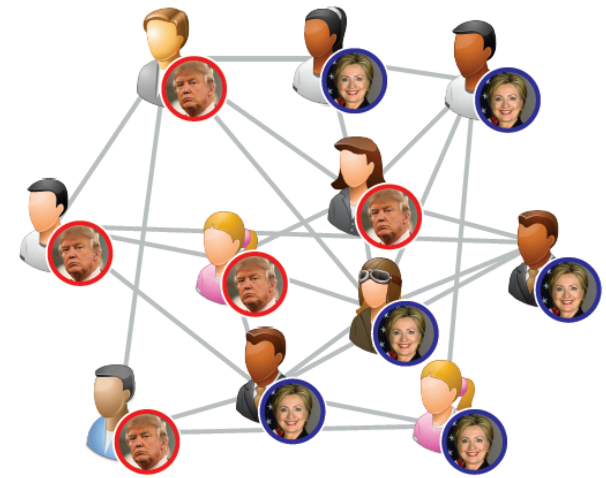
Computers can execute the scientific discovery process by playing a "**game**" of finding the optimal knowledge graph from a multi-graph of modeling possibilities through trial-and-error and policy learning.

Discovering/incorporating new ideas and descriptors not known/used in classical modeling approach

Consider **descriptors of data as the ingredients for theory**



Domain structure



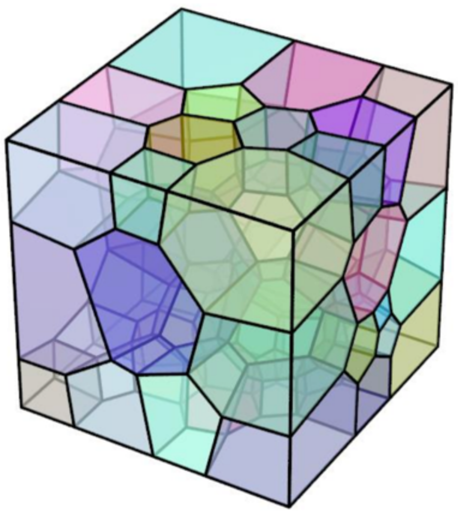
Social network

Figures from geometriclearning.com

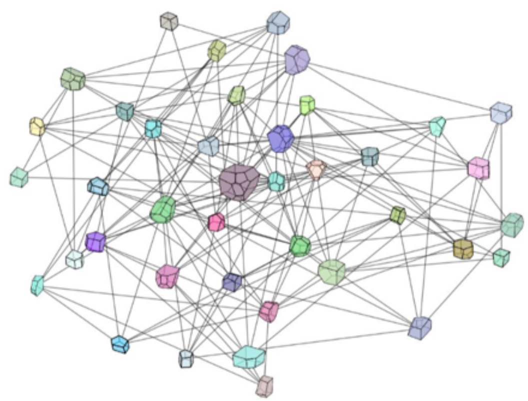
Example: Incorporating Non-Euclidean Data for Predictive Damage-Plasticity Models

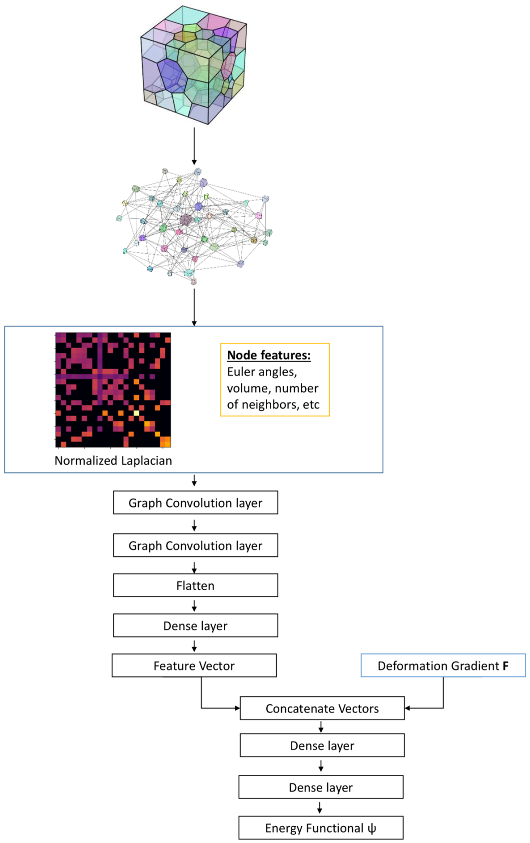
Microstructural information provides constraints that regularize the predictions

Polycrystal RVE



Weight crystal connectivity graph

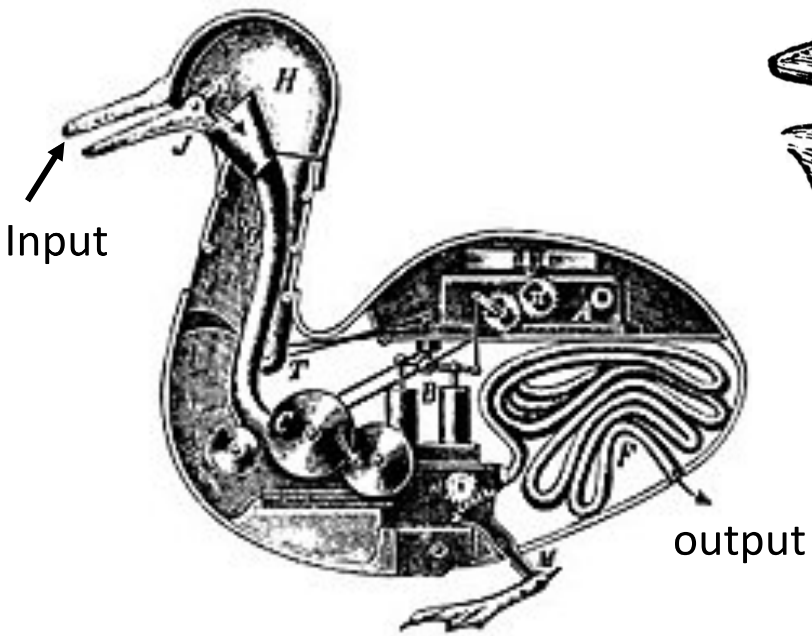




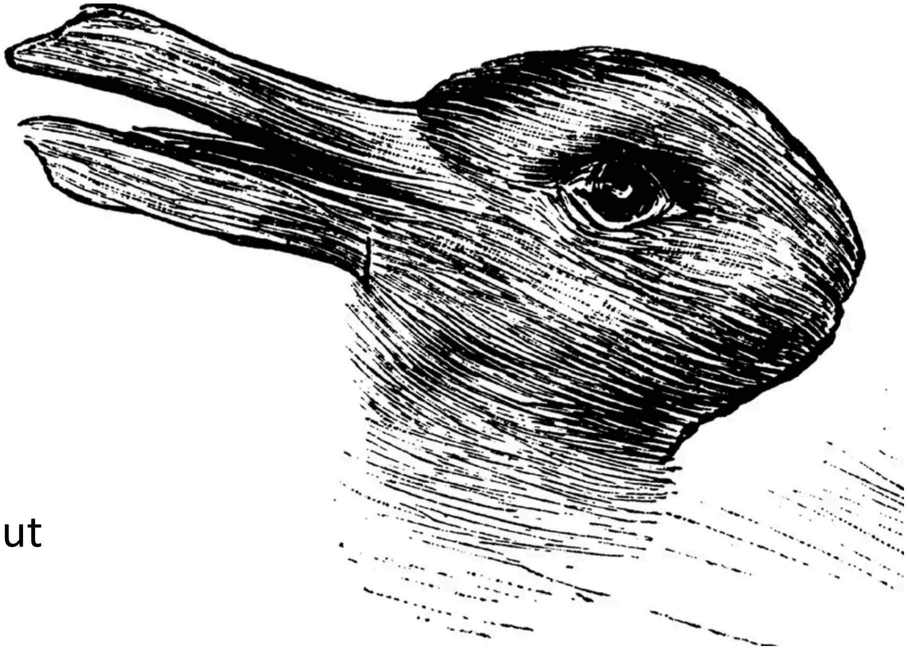
Vlassis, Ma, Sun, under review

Which Machine Learning?

Rationale of Predictions: External behaviors vs. internal properties



Canard Digérateur (1741)



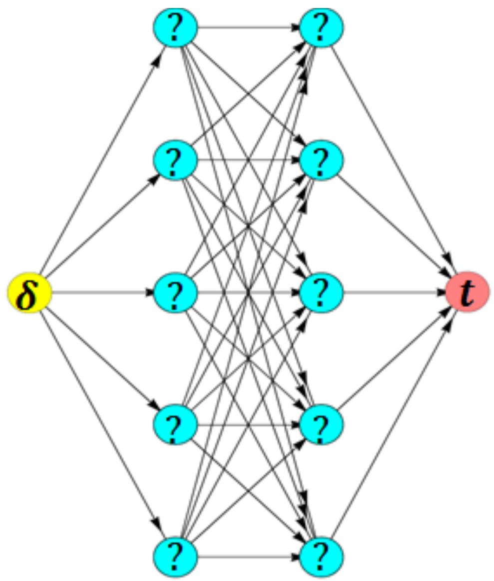
Duck-rabbit (1892)

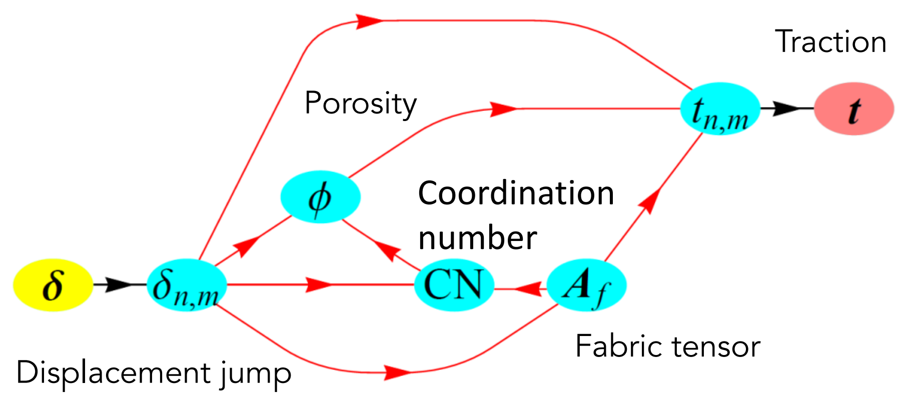
Scientific machine learning for constitutive modeling process

Machine Learning focusing on internal properties

Why?

- Machine learning is often being used as a **black box** and people need to develop trust for it. (Geotechnical engineering problems are high-regret & safety-critical)
- Small data (geomechanics experiments) versus Big data (Image Recognition)
- Leveraging **domain knowledge and constraints** in ML formulations





Black box ANN – designed to replicate ***external behaviors*** *without caring internal properties (e.g. thermodynamics…etc)*

Graph-based predictions – designed to generate knowledge represented by directed graph with the same ***internal properties*** of human thinkers.

Why game?

1. Emulating the scientific process of generating material constitutive laws as a game
2. Use directed multigraph and directed graph to represent possible theories and models (**Graph representation of knowledge**)
3. Use deep reinforcement learning to find optimal way to generate knowledge and model that best represented the data among all possibility (deep reinforcement learning)

Captured Stones

3 hours

AlphaGo Zero plays like a human beginner, forgoing long term strategy to focus on greedily capturing as many stones as possible.

Captured Stones

70 hours

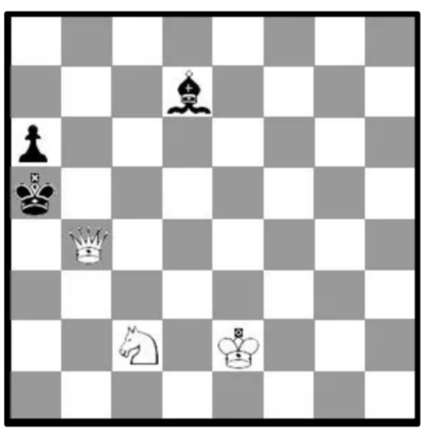
AlphaGo Zero plays at super-human level. The game is disciplined and involves multiple challenges across the board.

Ref: https://deepmind.com/blog/alphago-zero-learning-scratch/

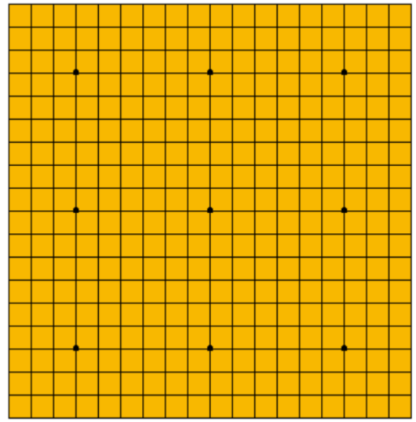
Analogy of Constitutive Modeling to Games

Chess Game

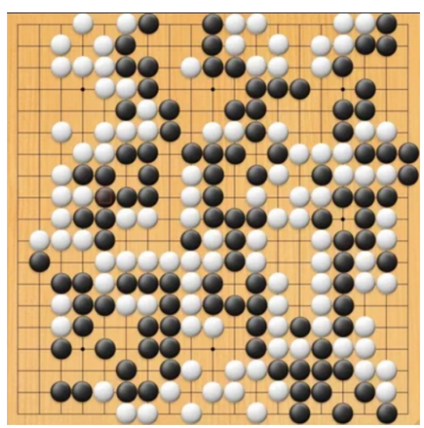
Move pieces to put the opponent's king in "checkmate"



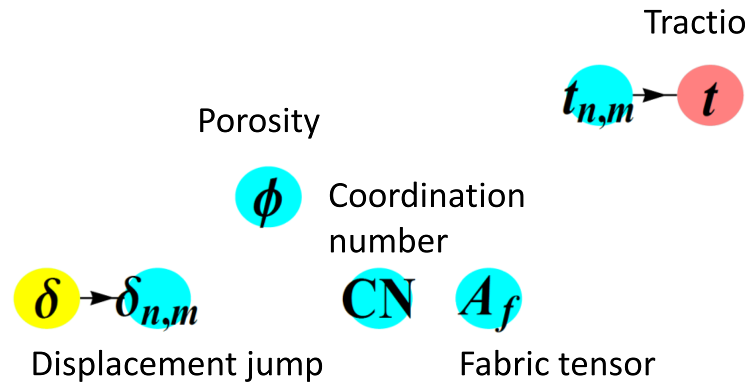
Go Game



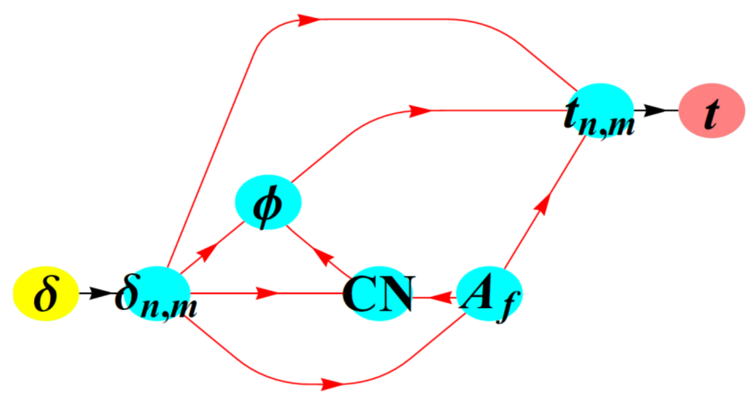
Place pieces to control more territory than your opponent



Meta-modeling Game



Connect edges to generate optimal internal information flow of constitutive models



Superhuman Performance of AI in learning the strategies of games

Alpha Go Zero

Legal game positions:
2e170
\> atoms in universe
1.6e79

https://deepmind.com/blog/alphago-zero-learning-scratch/

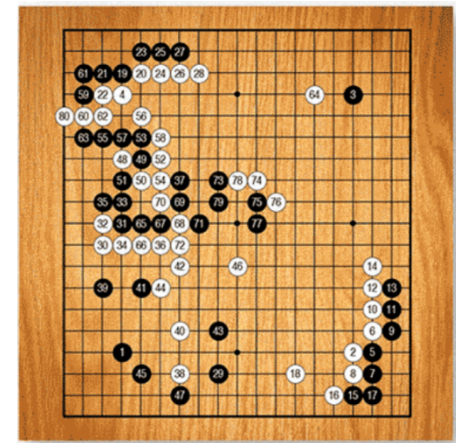
3 hours

Beginner level
with greedy plays

19 hours

Learnt the fundamentals
of Go strategies

70 hours



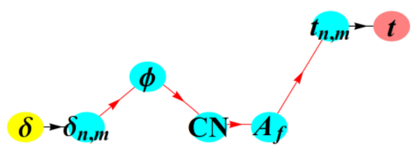
Super-human level
with disciplined plays

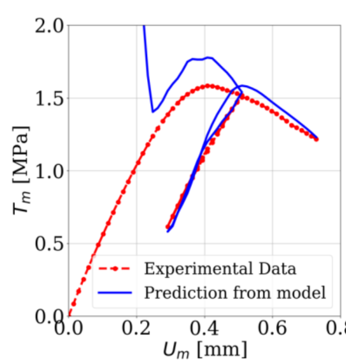
Meta modeling DRL

Legal game positions depend on the number of nodes of internal features

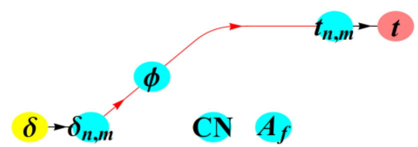
In our example:
over 2e4

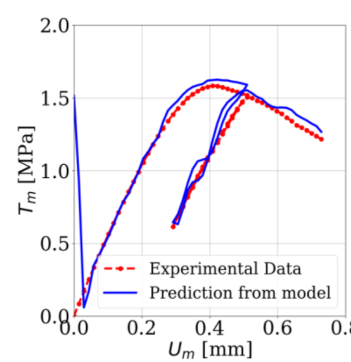
10 games



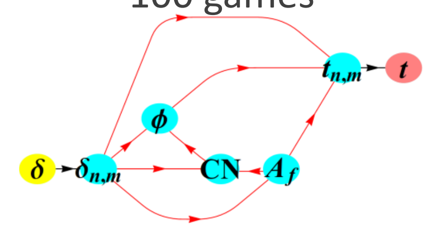


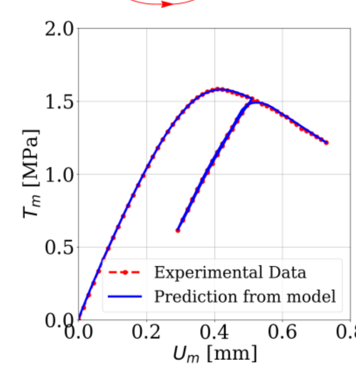
30 games





100 games

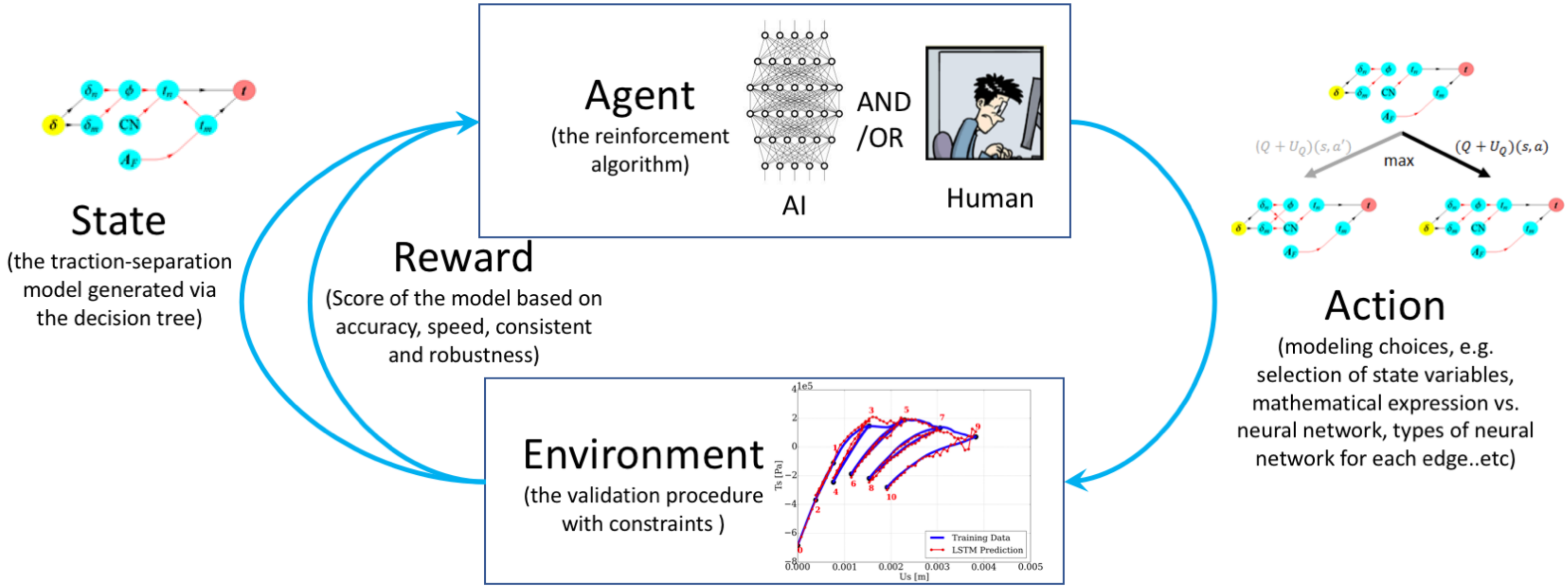




In a nutshell, ..the process of writing constitutive laws/surrogate models as a game AND this game can be played by AI or human

	Game of chess	Game of modeling writing in directed graph
Definition of game	Make a sequence of decisions to maximize the probability to win	Make a sequence of decisions to maximize the score of the constitutive model
Game board	8×8 grid	Directed graph with predefined nodes of physical quantities and edges of definition or universal principles
Game state	Configuration of chess pieces on the board	Configuration of directed graph representing the constitutive model
Game action	Move chess pieces	Select among modeling choices. For instance 1. What physical quantities are included? 2. How physical quantities are linked? 3. What are the edges between physical quantities?
Game rule	Restrictions on chess piece movements	Universal principles
Game reward	Win, draw or lose (discontinuous)	Model score (continuous)
Reward evaluation	Only available at the end	Only available at the end

Meta-modeling of traction-separation law



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 a modeler as a game* whose goal is to replicate the physics as close as possible
- Dee-Q-learning creates AI to play the game and learn from repeating generating models automatically

How to build the modeling game?

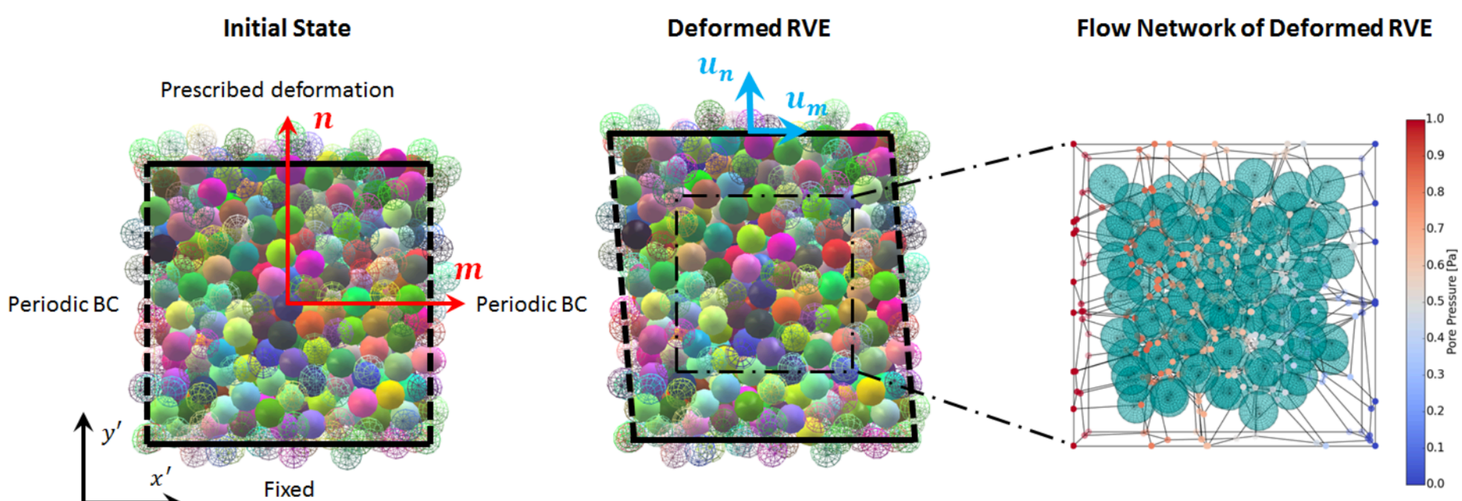
Game Environment – Data Generation: Computational homogenization of traction-separation law for strong discontinuity

Hill-Mandel Lemma for bulk volume

Solid skeleton: $\langle \boldsymbol{\sigma}' \rangle : \langle \dot{\boldsymbol{\epsilon}} \rangle = \langle \boldsymbol{\sigma}' : \dot{\boldsymbol{\epsilon}} \rangle$ Darcy's flow: $\langle \nabla p \cdot \boldsymbol{q} \rangle = \langle \nabla p \rangle \cdot \langle \boldsymbol{q} \rangle$

Hill-Mandel Lemma for interface

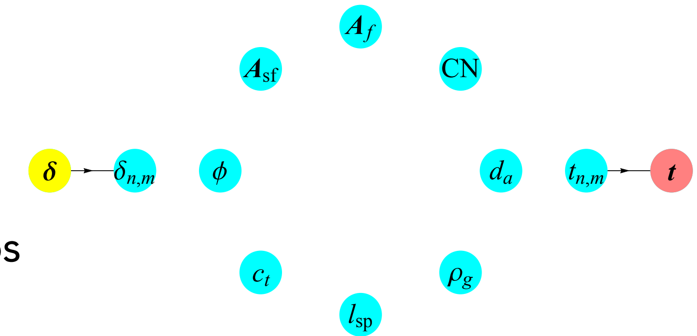
$$h_0 \langle \boldsymbol{\sigma}' : \dot{\boldsymbol{\epsilon}} \rangle = \langle \boldsymbol{T}'_\Gamma \rangle \cdot [\![\dot{\boldsymbol{u}}]\!] = \langle T_n \rangle [\![\dot{u}]\!]_n + \langle T_m \rangle [\![\dot{u}]\!]_m$$



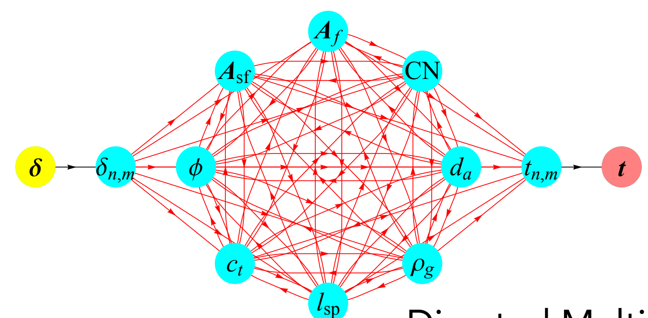
Sun, Andrade, Rudnicki, IJNME, 2011, Wang & Sun, CMAME 2016, Wang et al. IJMCE 2016, Wang & Sun, CMAME 2018

Game Board: Mechanics Knowledge Representation in Graphs, Directed Graph and Directed Multigraph

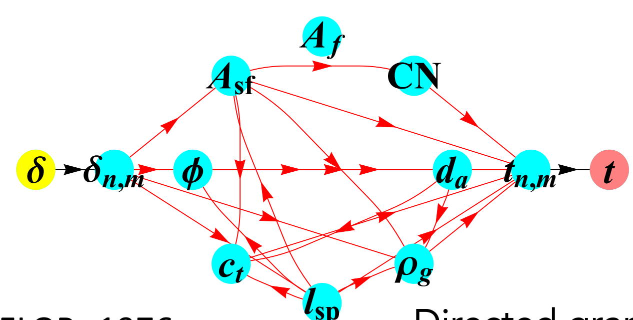
- Vertices - a measurable physical properties (permeability, thermal conductivity, force, displacement, strain..etc)
- Directed Edges – existing hierarchical relationships between two vertices (could be trained neural network or mathematical expression
- Edge Labels – the specific models used to connect two physical vertices. The model an be mathematical, neural network, support vector machine …etc
- Label Directed Multi-graph – all the possible way the vertices are connected by different combination of edges with different labels
- Directed graph – the optimal configuration of the vertices connected by edges, each with one unique labels, a subset of the directed multi-graph



Initial game state



Directed Multigraph

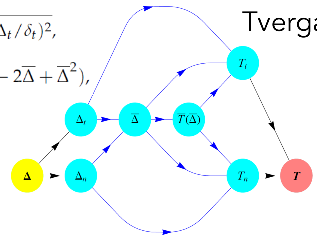


Directed graph

JF Sowa, Conceptual Graphs for a Data Base Interface, IBM J. RES. DEVELOP., 1976

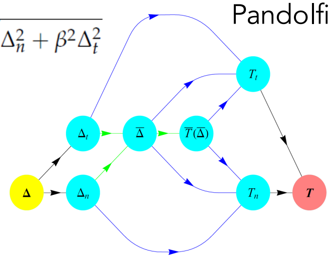
Game Board Generation: combining best moves from experts and ML edges

$$
\begin{aligned}
\overline{\Delta} &= \sqrt{(\Delta_n/\delta_n)^2 + (\Delta_t/\delta_t)^2}, \\
\overline{T}(\overline{\Delta}) &= \frac{27}{4}\sigma_{\max}\overline{\Delta}(1 - 2\overline{\Delta} + \overline{\Delta}^2), \\
T_n &= \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\frac{\Delta_n}{\delta_n}, \\
T_t &= \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\alpha\frac{\Delta_n}{\delta_t}
\end{aligned}
$$



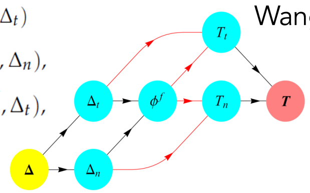
Tvergaard [1990]

$$
\begin{aligned}
\overline{\Delta} &= \tilde{\Delta}/\delta_n\,,\ \tilde{\Delta} = \sqrt{\Delta_n^2 + \beta^2\Delta_t^2} \\
\overline{T}(\overline{\Delta}) &= k\overline{\Delta} + c \\
T_n &= \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\frac{\Delta_n}{\delta_n}, \\
T_t &= \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\alpha\frac{\Delta_n}{\delta_t}
\end{aligned}
$$

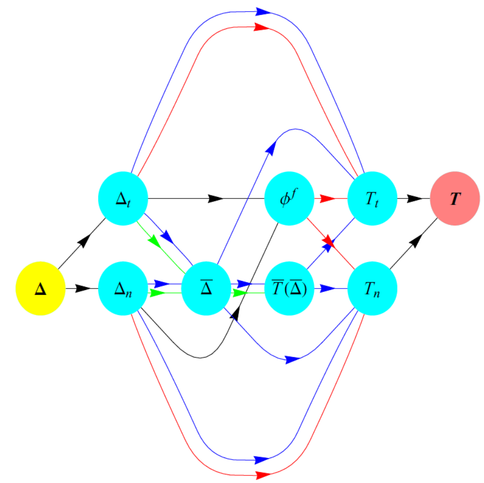


Pandolfi et al. [1999]

$$
\begin{aligned}
\phi^f &= \phi^f_0(1 + \Delta_n\Delta_t) \\
T_n &= f^{\mathrm{LSTM}}(\phi^f, \Delta_n), \\
T_t &= g^{\mathrm{LSTM}}(\phi^f, \Delta_t),
\end{aligned}
$$



Wang & Sun [2018]

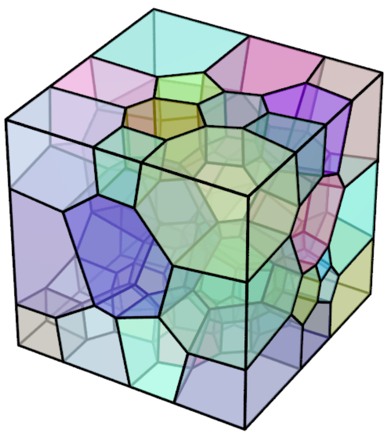


Directed multi-graph that contains all actions of three previous modelers recorded in Tvergaard, 1990, Pandolfi et al, 1990 and Wang & Sun [2018]

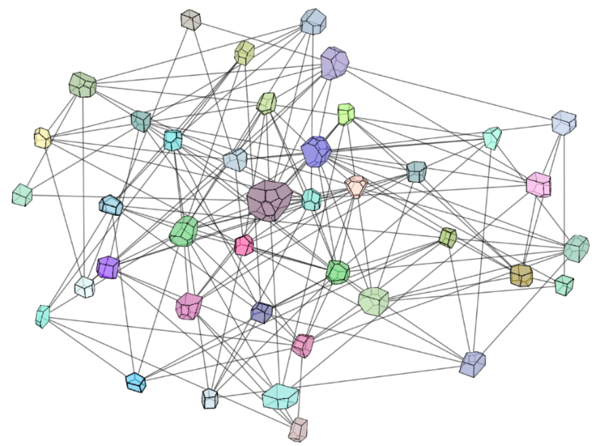
Definition A **labeled directed multi-graph** is a multi-graph with labeled vertices and edges which can be mathematically expressed as an 8-tuple $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n}_{V}, \boldsymbol{n}_{E})$ where $\mathbb{V}$ and $\mathbb{E}$ are the set of vertices and edges, $\mathbb{L}_{\mathbb{V}}$ and $\mathbb{L}_{\mathbb{E}}$ are the sets of labels for the vertices and edges, $s : \mathbb{E} \to \mathbb{V}$, $t : \mathbb{E} \to \mathbb{V}$ are the mappings that map the edge to the source and target vetrices, and $n_V : \mathbb{V} \to \mathbb{L}_{\mathbb{V}}$ and $n_E : \mathbb{E} \to \mathbb{L}_{\mathbb{E}}$ are the mapping that gives the vertices and edges the corresponding labels in $\mathbb{L}_{\mathbb{V}}$ and $\mathbb{L}_{\mathbb{E}}$ accordingly.

Adding new vertex (and physics) via Geometric Deep Learning

Poly-crystal Connectivity Graph for Anisotropic Energy Functional Prediction



Polycrystal RVE



Node-weighted undirected crystal connectivity graph

- Constitutive law generation from non-Euclidean grid data

$$W = W(\boldsymbol{F}, \mathbb{G}) \ , : \boldsymbol{P} = \frac{\partial W}{\partial \boldsymbol{F}}$$

$$\mathbb{G} = (\mathbb{V}, \mathbb{E})$$

Vertices (grain)

Edges (grain contacts)

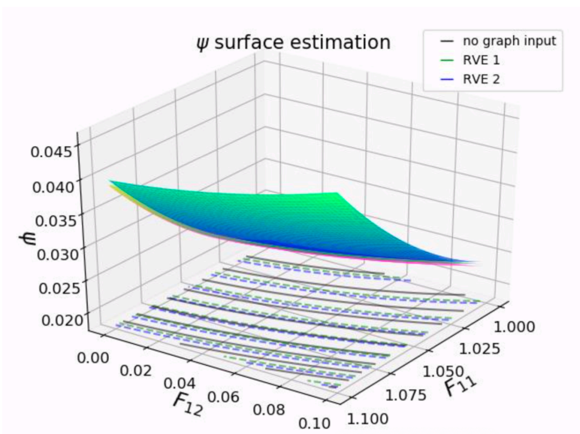
<u>Node weights:</u> crystal orientation, volume, number of neighbors, number of faces, etc.

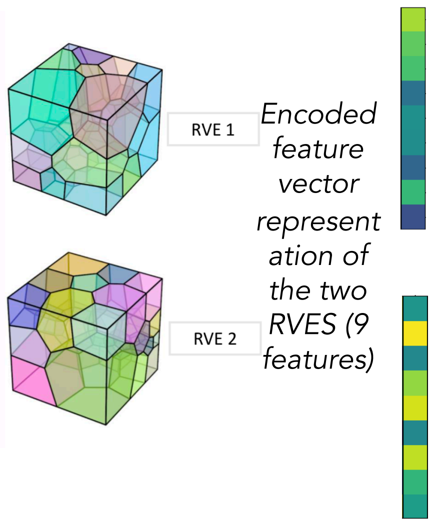
<u>Edge weights:</u> area of contact, angle of contact, etc.

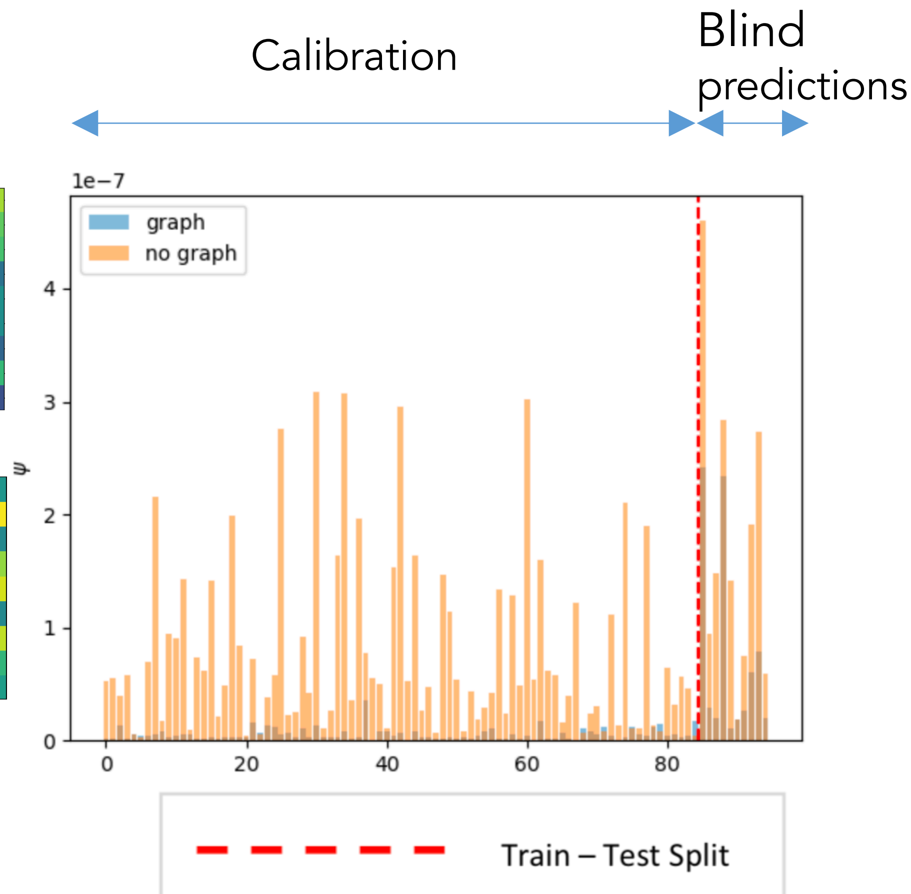
<u>Why switch from Euclidean to Non-Euclidean space:</u>

- Data structures crafted meaningfully with domain expertise / interpretable
- Euclidean grid data (eg. images) → ambiguity of interpreted features
- Eliminate grid resolution dependency → computational efficiency

Adding new vertex: Graph Data – Weighted undirected graph







- Generally superior accuracy for blind prediction AND calibration with graph data
- Most important graph node feature: crystal orientation (Euler angles)

Game Reward: Objective function with k-fold cross-validation

- Example Score system:
  - 0.4 weight on **accuracy** of the predictions
  - 0.4 weight on **consistency** in replication of training data and in forward prediction
  - 0.2 weight on *model execution time*

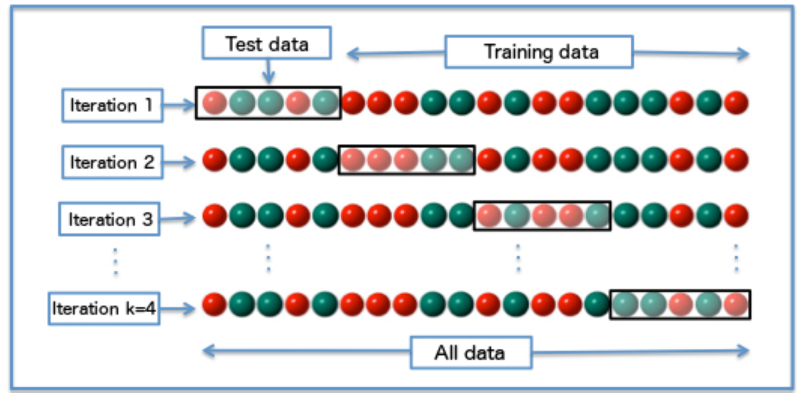
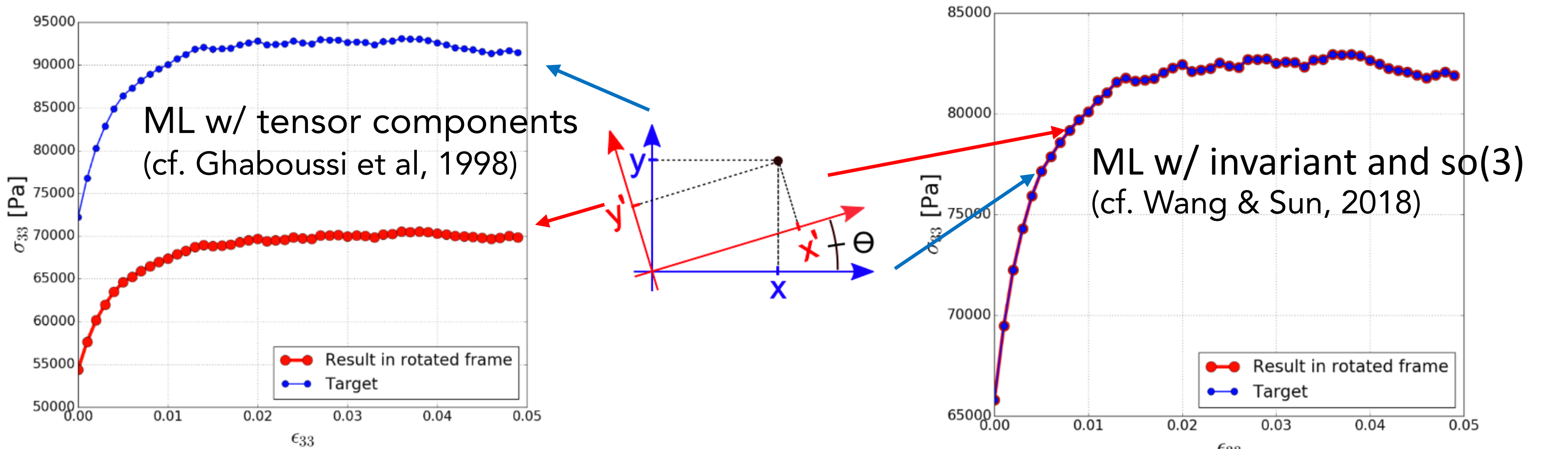


Figure from wikipedia

Instants of constitutive laws are considered as directed graphs. Given a dataset that contains the time history information of n types of data labeled by $l_i \in \mathbb{L}_{\mathbb{V}}$ and the labeled direct graph defined by the 8-tuple $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{n}_V, \boldsymbol{n}_E)$, and objective function SCORE and constraints to enforce universal principles. Find an subgraph $\mathbb{G}'$ of $\mathbb{G}$ consists of vertices $\boldsymbol{V} \in \mathbb{V}^s \subseteq \mathbb{V}$ and edges $\boldsymbol{E} \in \mathbb{E}^s \subseteq \mathbb{E}$ such that 1) $\mathbb{G}'$ is a directed acyclic graph, 2) a score metric is maximized under a set of m constraints $f_i(l_1, l_2, \ldots, l_n) = 0, i = 1, \ldots, m$ where , i.e.,

$$\begin{aligned} \underset{l_i}{\text{maximize}} \quad & \text{SCORE}(l_1, l_2, \ldots, l_n) \\ \text{subject to} \quad & f_i(i_i) = 0, \ i = 1, \ldots, m. \end{aligned} \tag{17}$$

Game Rules -- where Mechanics human knowledge is used (e.g. material; frame indifference)



Tensor component as input lead to lack of objectivity (prediction depends on observer)

Tensor invariant as input lead to lack of objectivity (prediction independent of observer)

Remedy 1: we proposed – use invariants and parametrize rotations, i.e. modify the directed graph (RIGHT RIGHT)

$$\mathbf{c} = \sum_{A=1}^{3}\sum_{B=1}^{3} a_{AB}\, m^{(A)} \otimes m^{(B)} + \sum_{A=1}^{3} \sigma_A \omega^{(A)},$$

$$\delta\tau = \sum_{A=1}^{3} \delta\tau_A m^{(A)} + \sum_{A=1}^{3} \sum_{B \neq A} \Omega_{AB}(\tau_B - \tau_A)\, m^{(AB)},$$

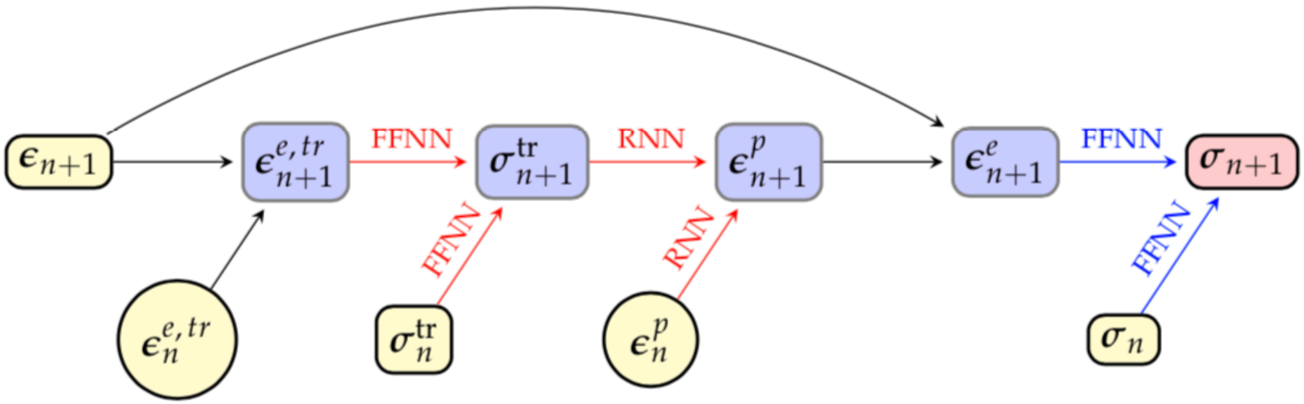
$$\mathbf{R}(\boldsymbol{\Psi}) = \sum_{k=0}^{\infty} \frac{\widetilde{\boldsymbol{\Psi}}^k}{k!}$$

$$t \mapsto \mathbf{R} \in \mathrm{SO}(3)$$

$$\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\Delta \widetilde{\boldsymbol{\Psi}}_n]$$

Remedy 2: Get more data with rotated frame (cf. Lefik & Schrelfer 2003)

Game Rules: Mechanics Principles (e.g. material; frame indifference)



elasto-plasticity
($DG2$)

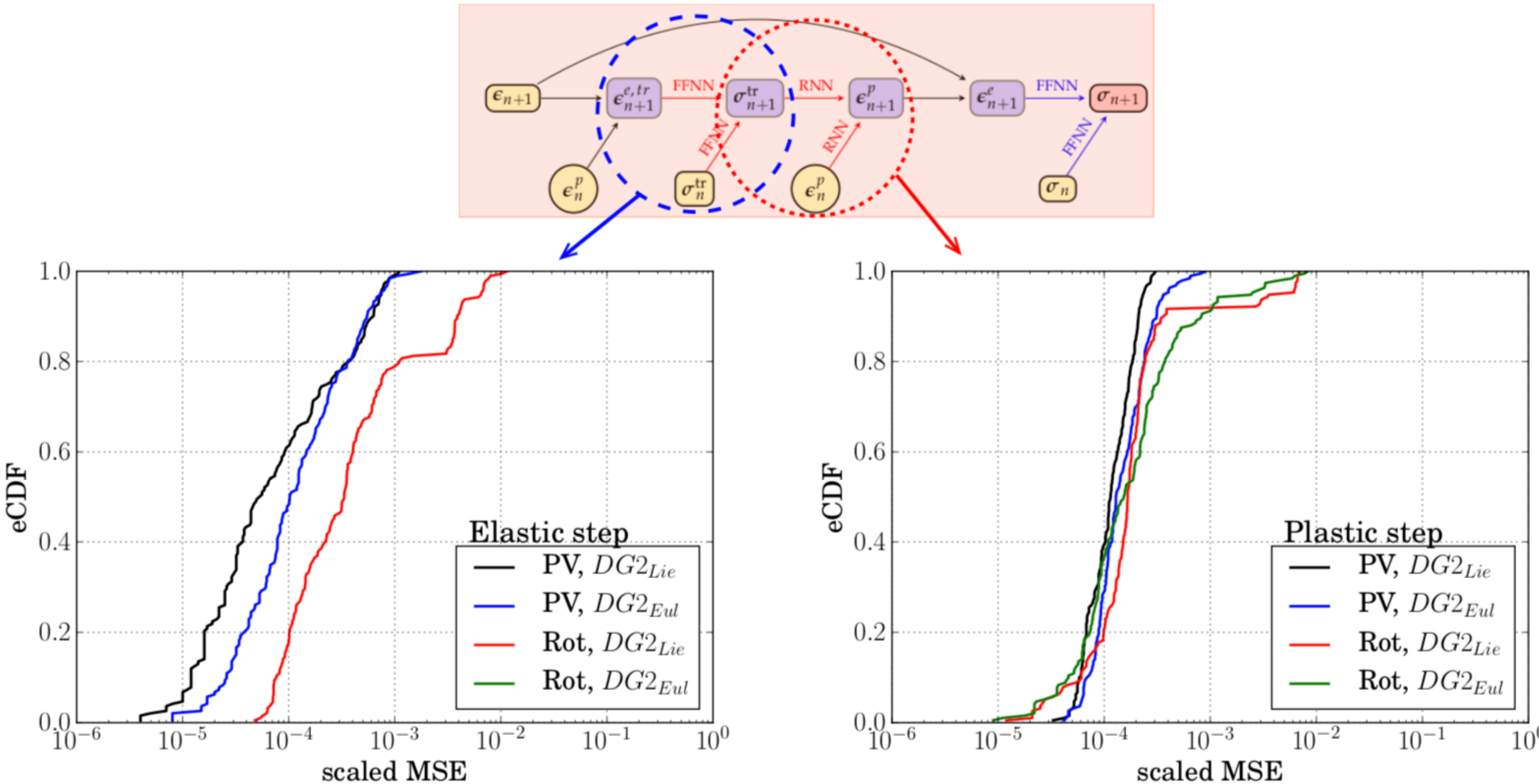
Euler Angle $$\phi_{Eu} = \sqrt{d(\varphi_1,\varphi_2)^2 + d(\theta_1,\theta_2)^2 + d(\psi_1,\psi_2)^2}\,.$$

Property Test $$\phi_{DI} = \|\boldsymbol{I} - \boldsymbol{R}_1\boldsymbol{R}_2^T\|_F = \sqrt{2\left[3 - \operatorname{tr}(\boldsymbol{R}_1\boldsymbol{R}_2^T)\right]}\,.$$

Lie algebra $$\phi_{Lie} := \|\log(\boldsymbol{R}_1\boldsymbol{R}_2^T)\| = \|\log(\boldsymbol{R}_1) - \log(\boldsymbol{R}_2)\| = \|\boldsymbol{W}_1 - \boldsymbol{W}_2\|\,,$$

where $$\boldsymbol{W} = \log\boldsymbol{R} = \begin{cases} \boldsymbol{0} & \text{if } \Theta = 0 \\ \dfrac{\Theta}{2\sin\Theta}(\boldsymbol{R} - \boldsymbol{R}^T) & \text{if } \Theta \in \,]0,\pi[\\ \pm\pi\,\tilde{\boldsymbol{v}} & \text{if } \Theta = \pi\,, \end{cases}$$

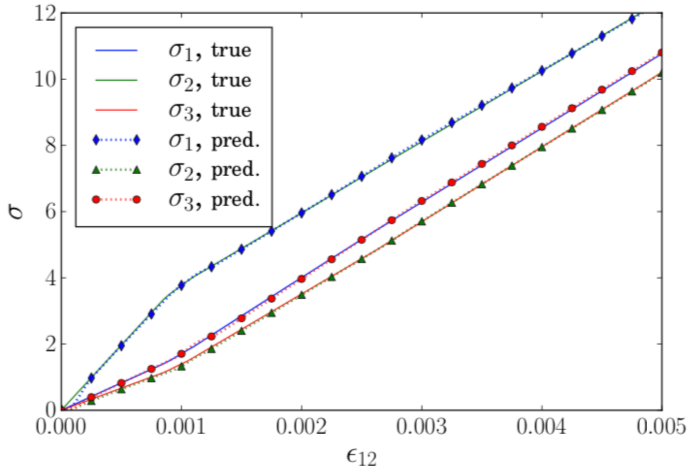
Game Rules: Mechanics Principles (e.g. material; frame indifference)

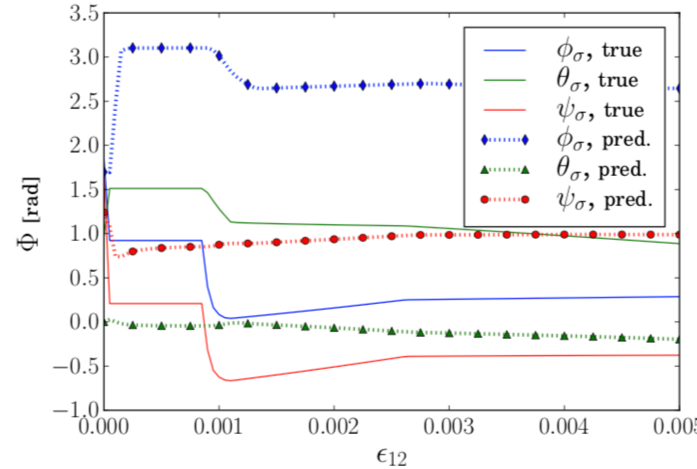


FCC Crystal plasticity Example

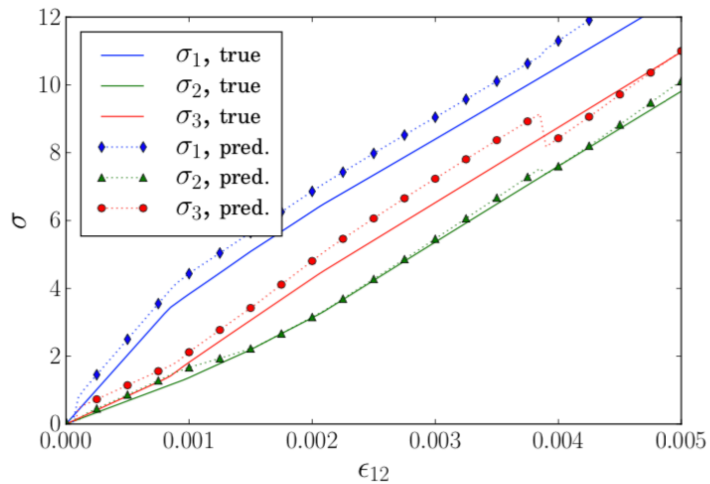
Game Rules: Mechanics Principles (e.g. material; frame indifference)

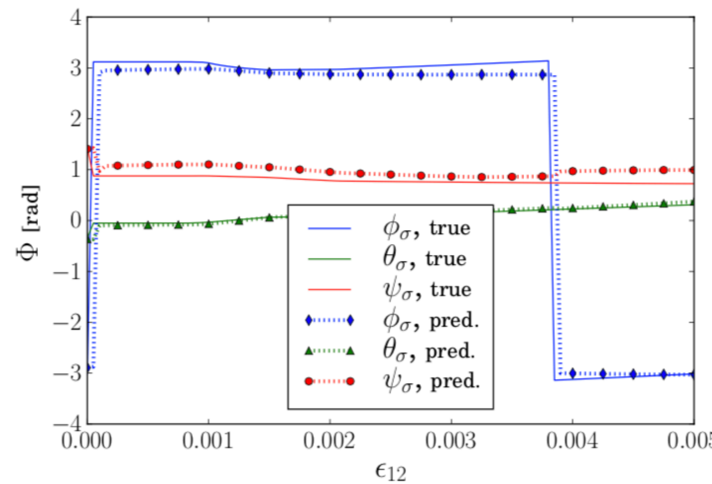
Component-based training





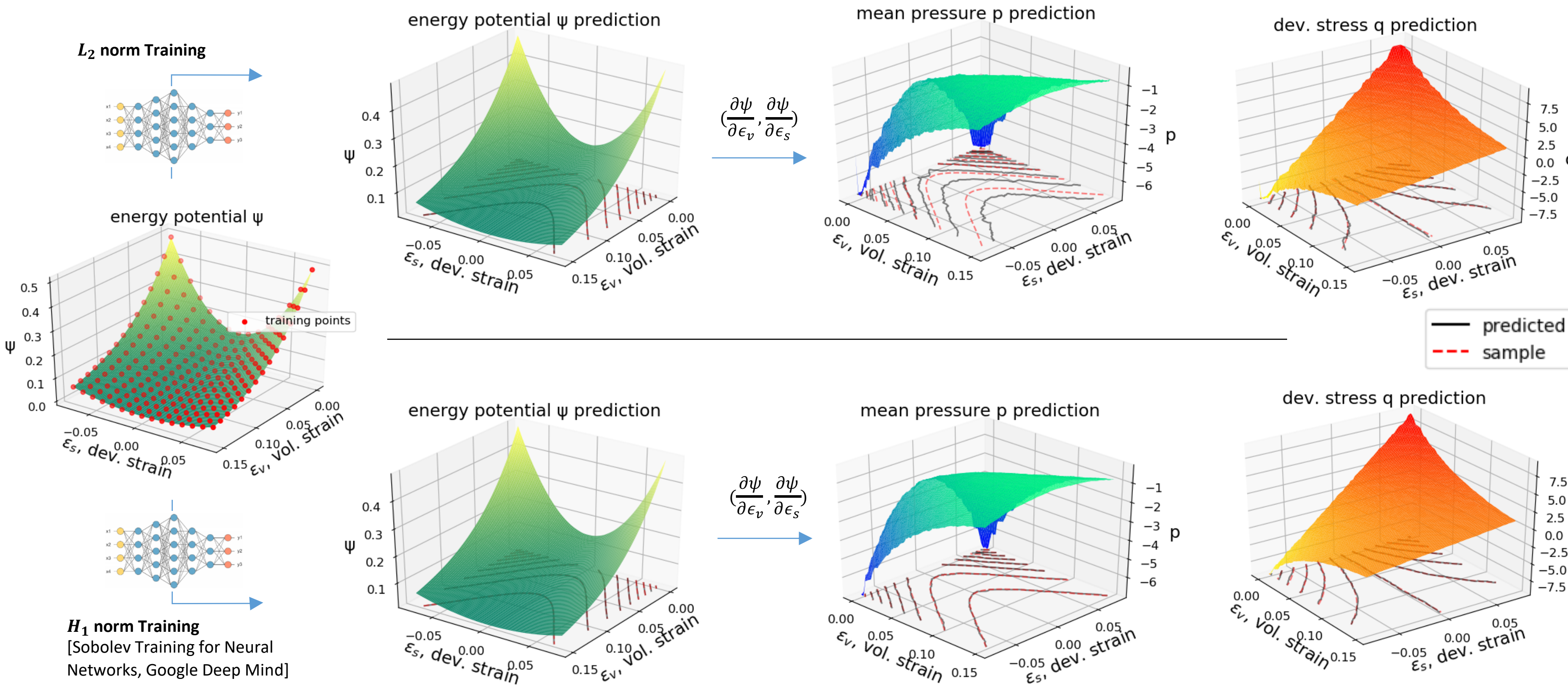
Lie-algebra training





FCC Crystal plasticity Example

Game Rule: Convexity and smoothness of elastic stored energy

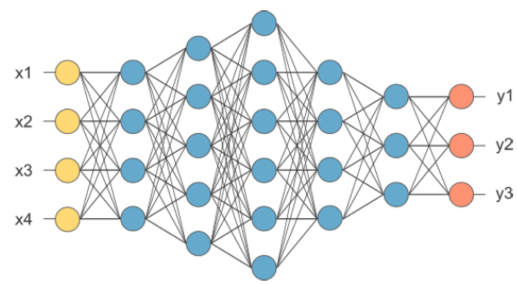


$$L_2\ norm\text{: } \min \frac{1}{N}\sum_{i=1}^{N} \left\|\psi_{true}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}}) - \psi_{pred}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}})\right\|_2^2$$

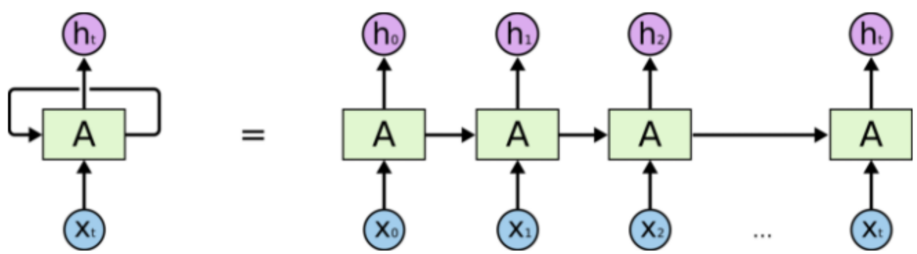
$$H_1\ norm\text{: } \min \frac{1}{N}\sum_{i=1}^{N} \left\|\psi_{true}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}}) - \psi_{pred}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}})\right\|_2^2 + \left\|\frac{\partial\psi_{true}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}})}{\partial\boldsymbol{\epsilon}^{\boldsymbol{e}}} - \frac{\partial\psi_{pred}^{e}(\boldsymbol{\epsilon}^{\boldsymbol{e}})}{\partial\boldsymbol{\epsilon}^{\boldsymbol{e}}}\right\|_2^2$$

Game Move (Example): Neural network models for connecting information flow

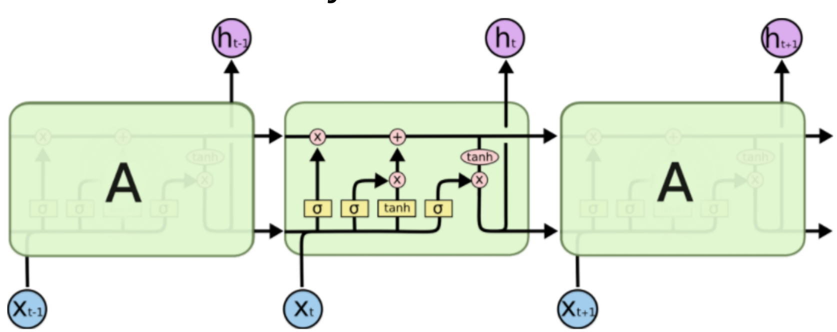
Multilayer perceptron



Recurrent neural networks



Long-short term memory



The repeating module in an LSTM contains four interacting layers.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[J Ghaboussi et al. 1991]

[M Lefik and BA Schrefler. 2003]

Treating path-dependent behavior is non-trivial

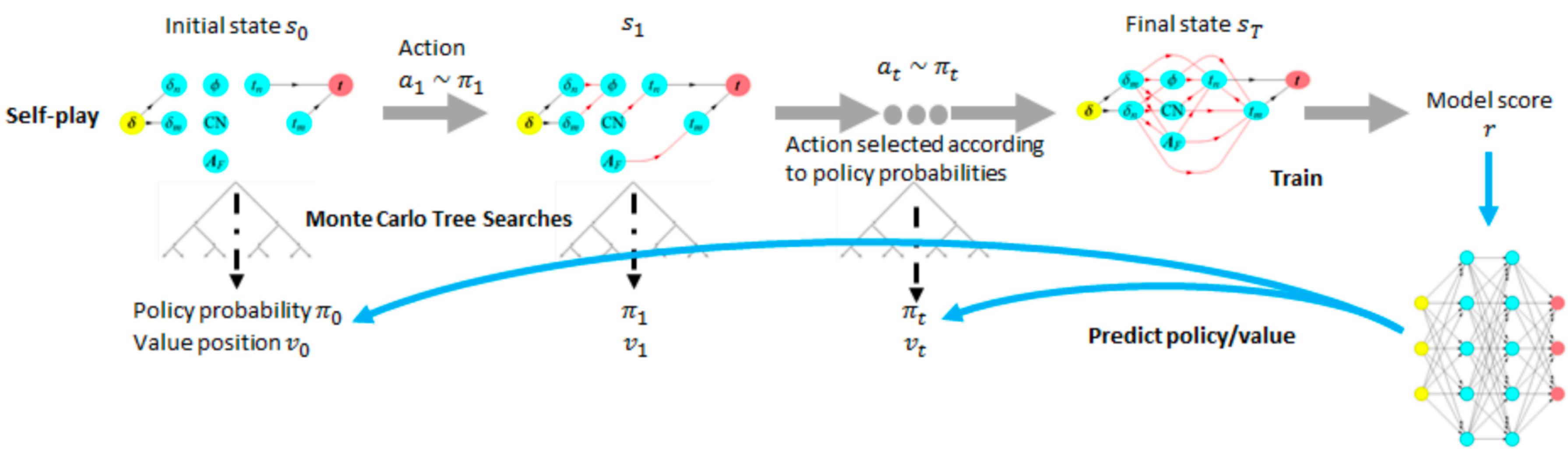
[Zhu JH et al. 1998]

- Capable of memorizing deformation history
- Gradient vanishes in long term memory

This work

- Overcoming gradient vanishing or exploding issues
- Circumventing over-fitting with dropout layers

Game Playing: Improvement of predictions through self-playing



- Self-play reinforcement learning of traction-separation law.
- In each "play", reward is assessed, then the reward for each action is estimated.
- If we know the true "reward" of each action, we can determine the optimal action sequence that yields the best model.

Game Learning: Improvement of predictions through self-playing: Monte Carlo Tree Search

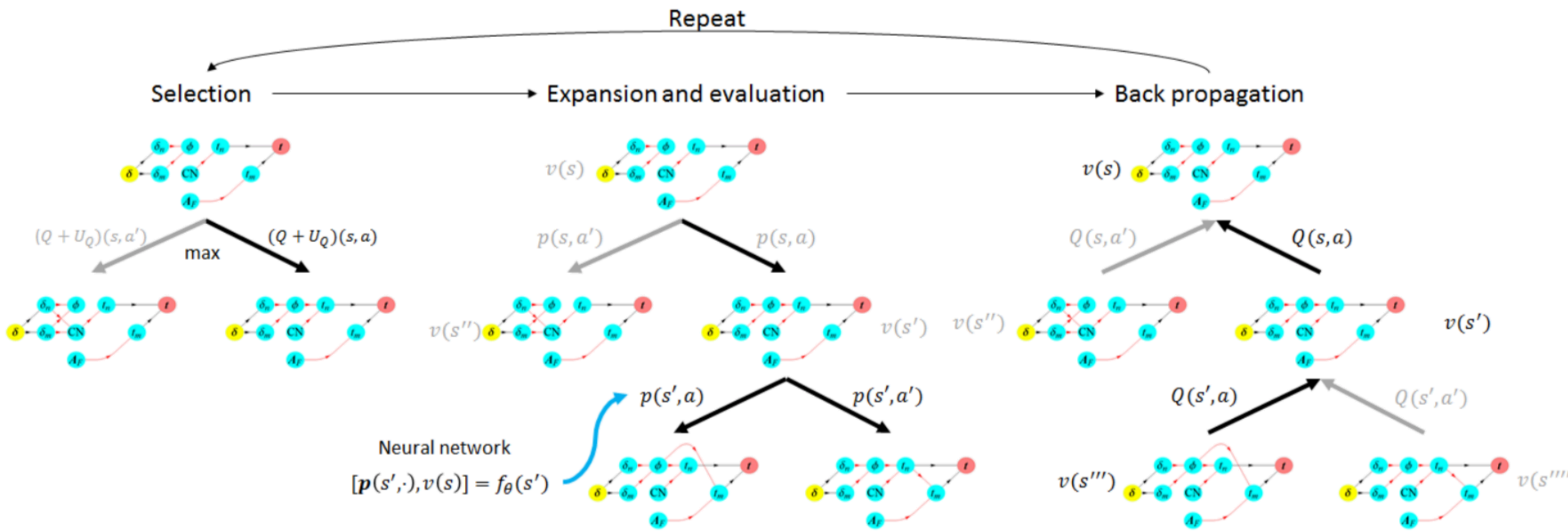


Figure 5: *Actual snapshot of Monte Carlo Tree Search (MCTS) in a game of constitutive models (figure design borrowed from [68]). A sequence of actions are selected from the root state s, each maximizing the upper confidence bound $Q(s,a) + U_Q(s,a)$. The leaf node s_L is expanded and its policy probabilities and position value are evaluated from the neural network $\mathbf{p}(s^L)$ and $v(s^L) = f_\theta(s^L)$. The action values Q in the tree are updated from the evaluation of the leaf node. Finally search probabilities $\boldsymbol{\pi}$ are returned to guide the next action in self-play*

Results?

Training Example 1: Training traction-separation law from DEM simulations

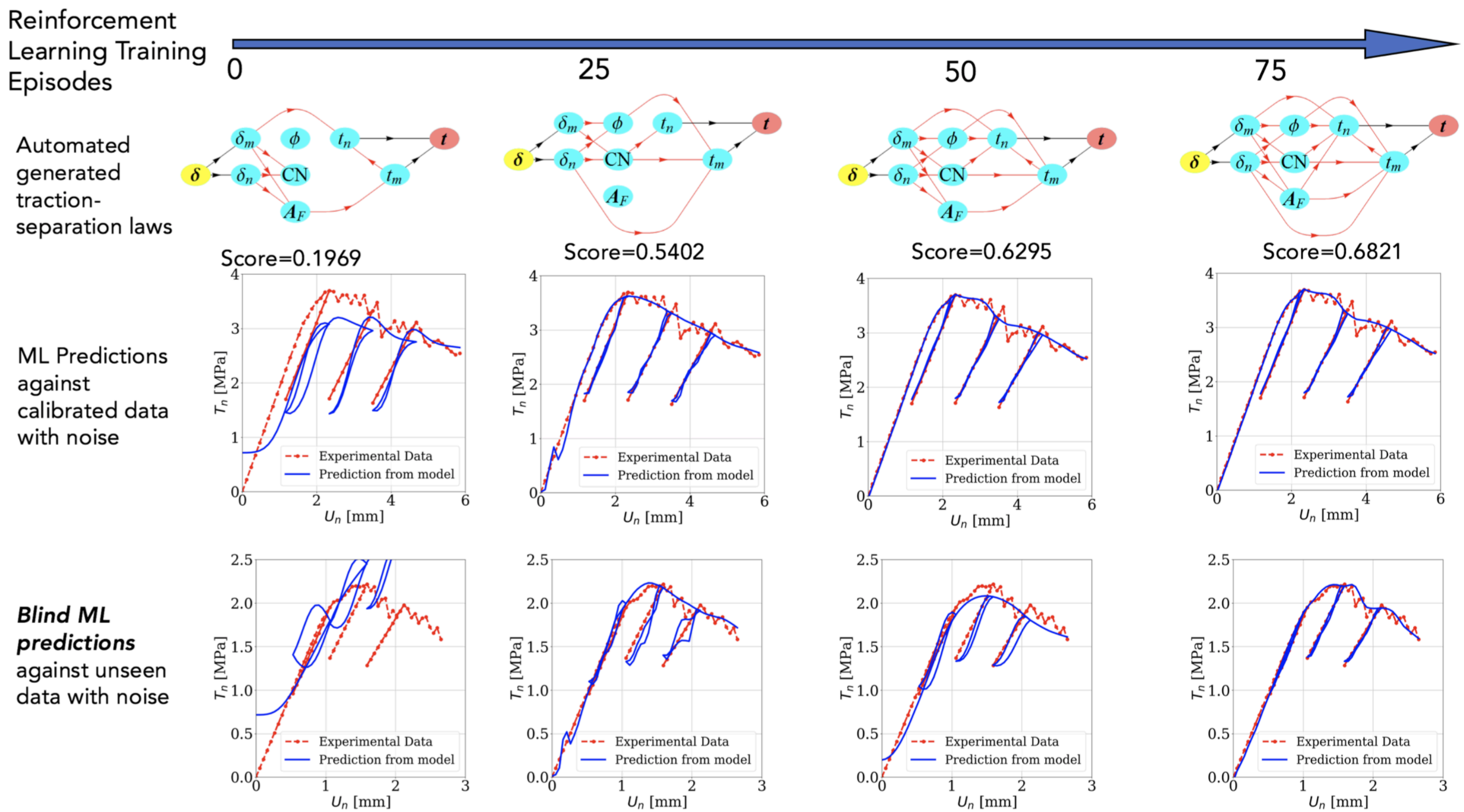
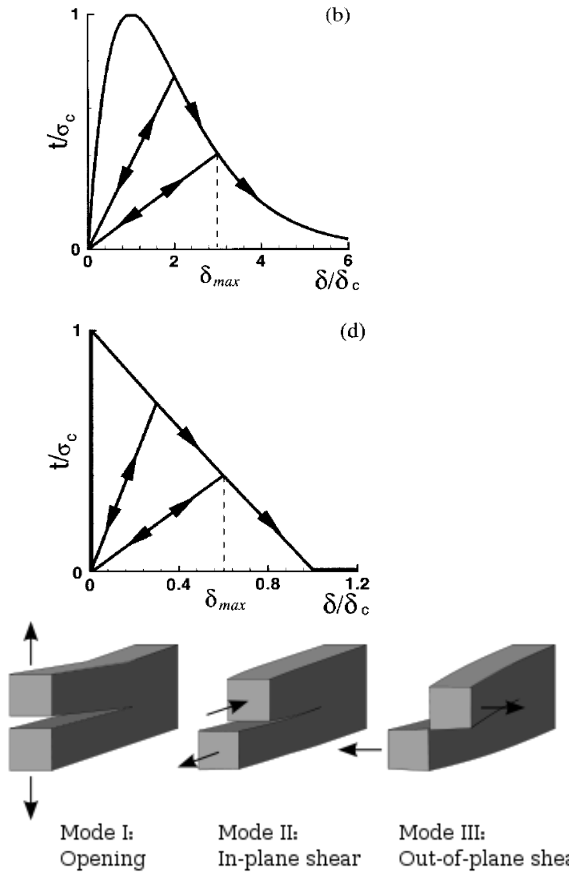


Figure 6: *Improved calibration and blind prediction scores throughout the training. As time progresses, the AI learn to write models with increasingly precise predictions. After 75 episodes (i.e. 75 different constitutive laws are built, both the calibration exercises and blind predictions (blue) are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

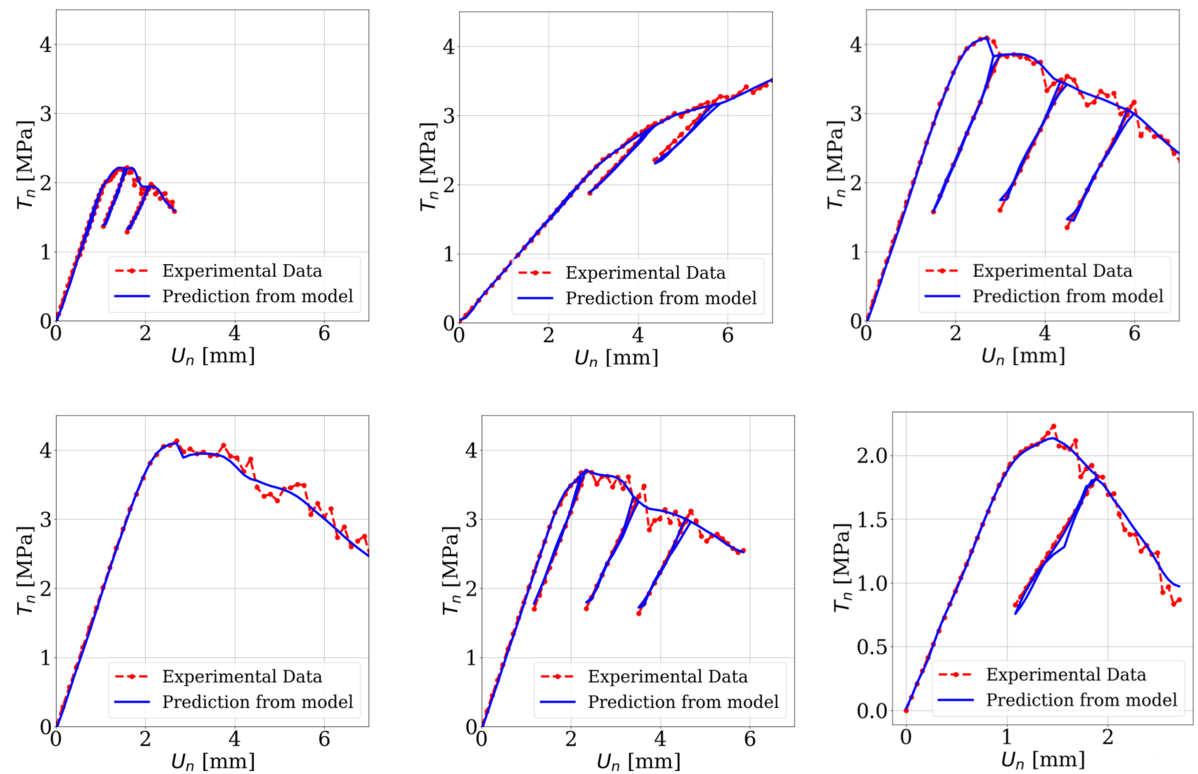
Numerical example: self-learned knowledge of cyclic traction-separation law

Hand-crafted TS law



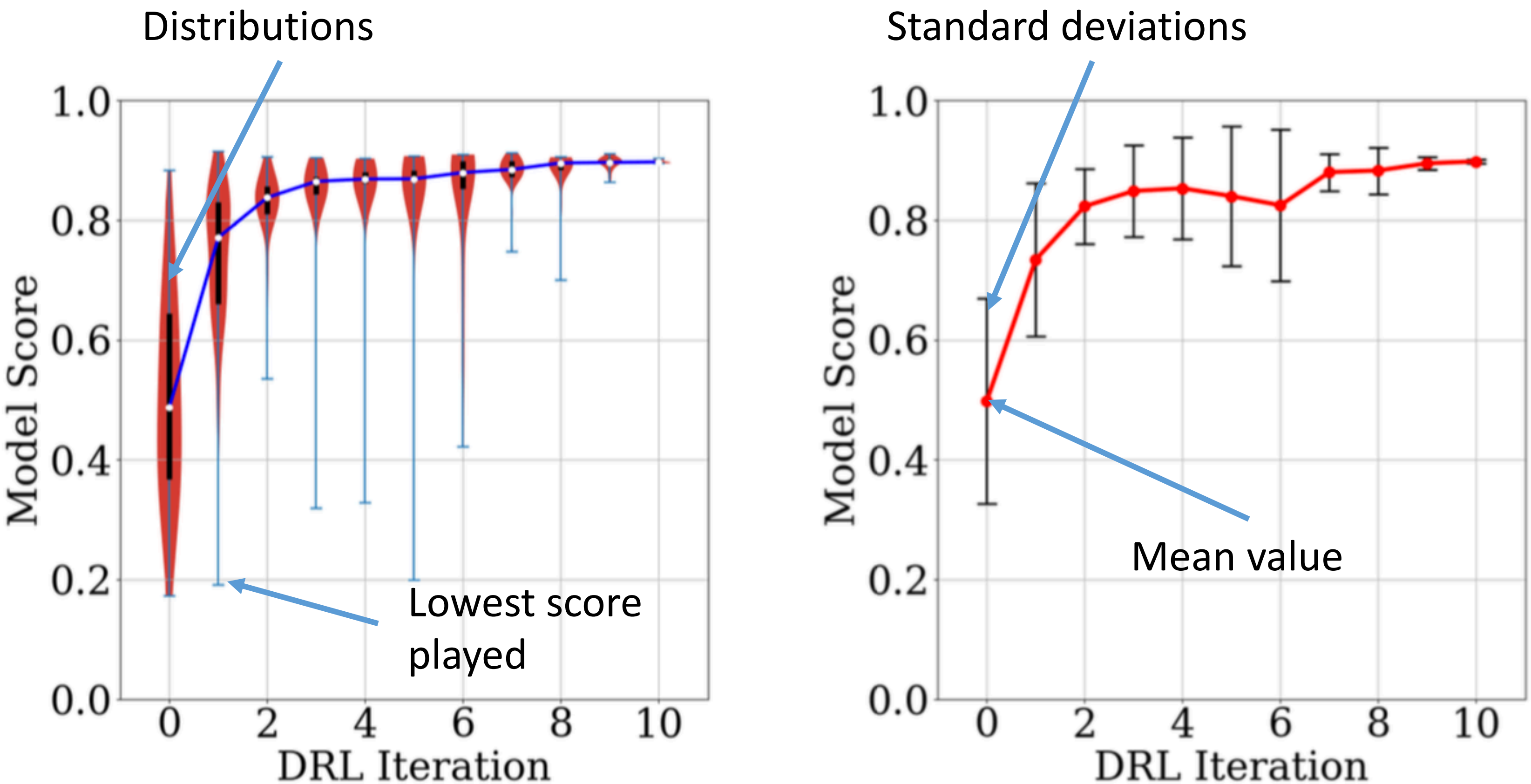
Hand-crafted cohesive laws reviewed in [M Ortiz, A Pandolfi, 1999]

AI-generated knowledge graph TS law



Self-reinforcement-learned cohesive laws blind-validated against cyclic data

Performance of AI over self-learning sessions



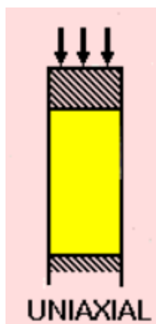
1.(a) Violin plots of the density distribution of model scores **(b)** Mean value and ± standard deviation of model score in each DRL iteration in each DRL iteration

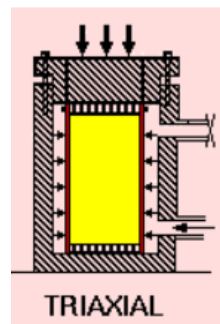
How much data do we need?

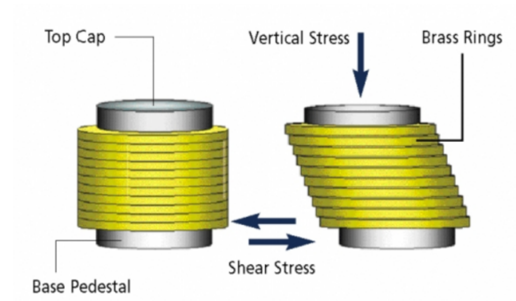
Two-agents to play the meta-modeling game collaboratively

Data Agent or experimentalist

- Game board: All experiment choices: (uniaxial, biaxial, simple shear, …)
- Game actions: choose the tests to be conducted for model calibrations
- Game goal:
  1. Maximize the final model score (global goal, need to be checked by the subsequent Model Game)
  2. Minimize the total number of tests (local goal)



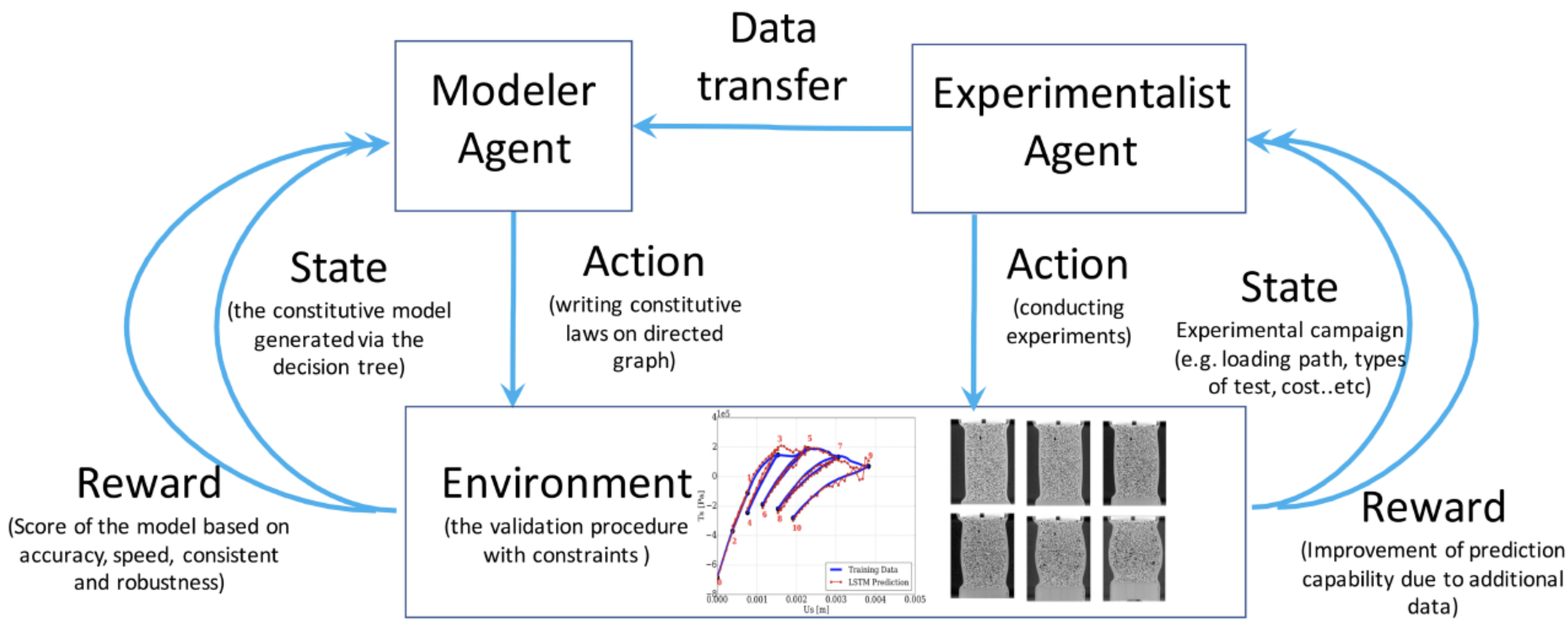




Model Agent or modeler (identical to the previous single agent)

- Game board: All modeling choices: (mathematical, ANN, …)
- Game actions: choose the modeling edges to connect the physical quantities
- Game goal:
  1. Maximize the final model score

Two-agent game: data collections and meta-modeling



- *Both the modeler and the experimentalist* has a common goal of replicating the physics as close as possible.
- The experimentalist also has its local goal of minimizing the experiments but needs to work **collaboratively** with the modeler to achieve the common goal.
- **Multi-agent Multi-objective Deep-Q-learning** creates AI to play the Data and Model games and learn from repeating generating models automatically.
- The game stops when there is no more additional reward for new action.

Markov decision process for data collection and meta-modeling

Data Agent

Initial state s_0 — Set of experiments [T1]

Action $a_0 = T3$ →

State s_1 — Set of experiments [T1, T3] — Reward r_1

Action $a_1 = T5$ →

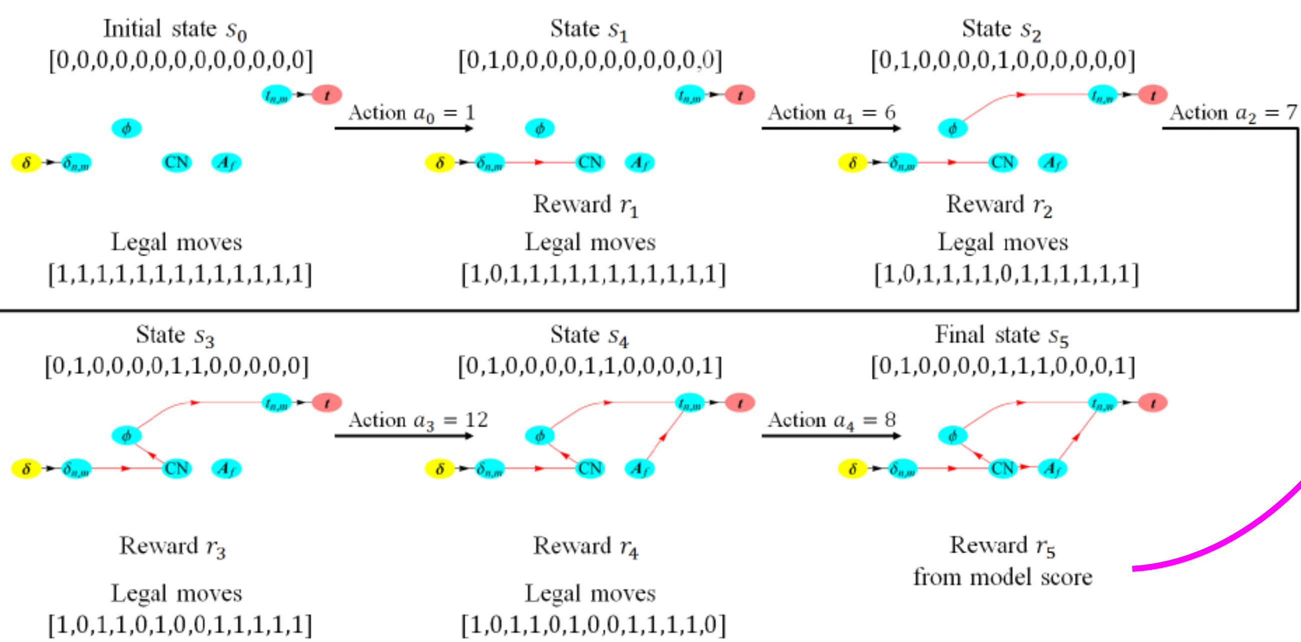
State s_2 — Set of experiments [T1, T3, T5] — Reward r_2

Action $a_2 = T8$ →

Final state s_3 — Set of experiments [T1, T3, T5, T8] — Reward r_3

Model Agent (identical to the previous single agent)

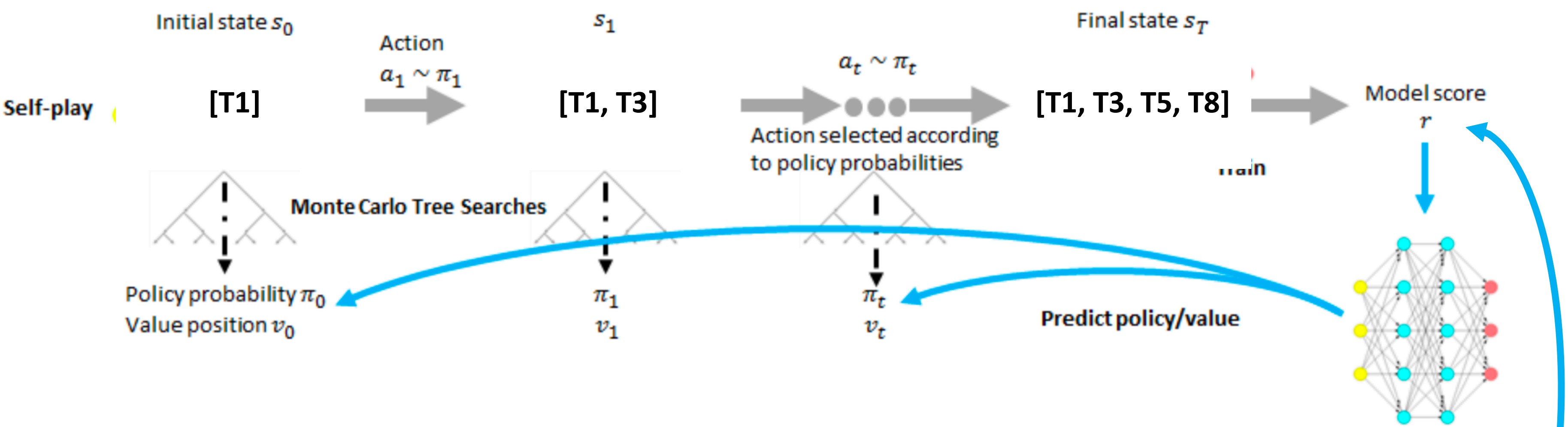
Construct calibration data



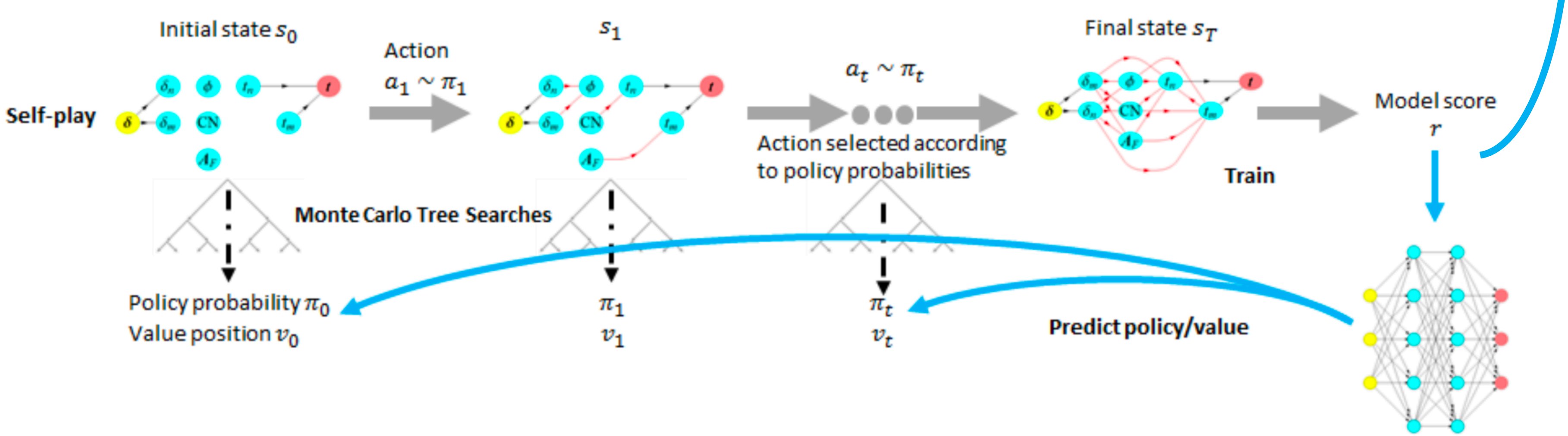
Global Reward

Self-play reinforcement learning of both Data Agent and Model Agent

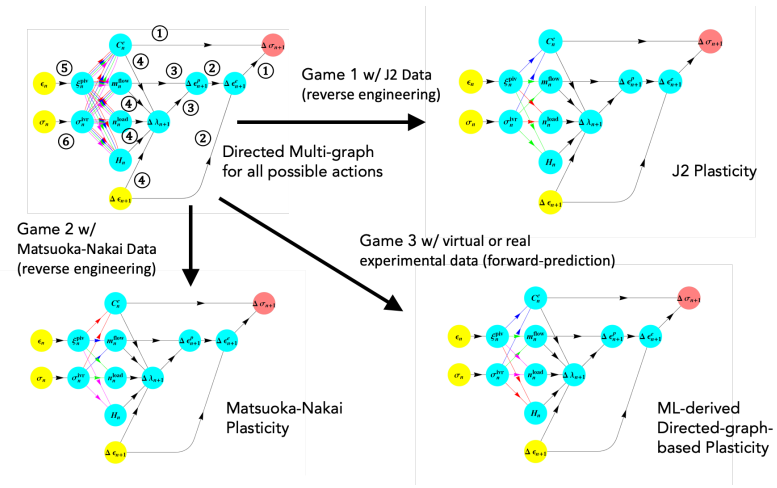
Data Agent

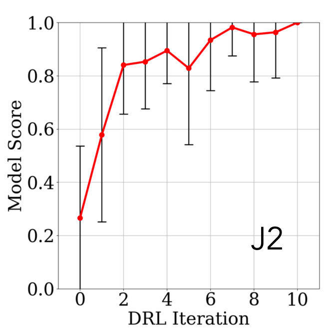


Model Agent (identical to the previous single agent)

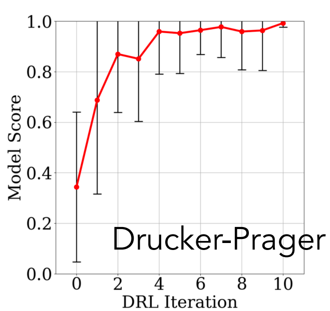


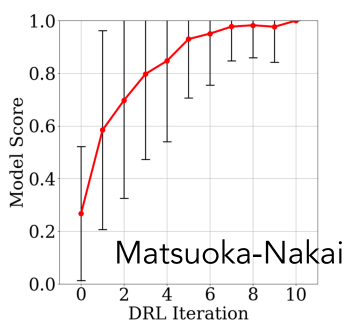
Numerical Example 2:
Reverse engineering constitutive laws

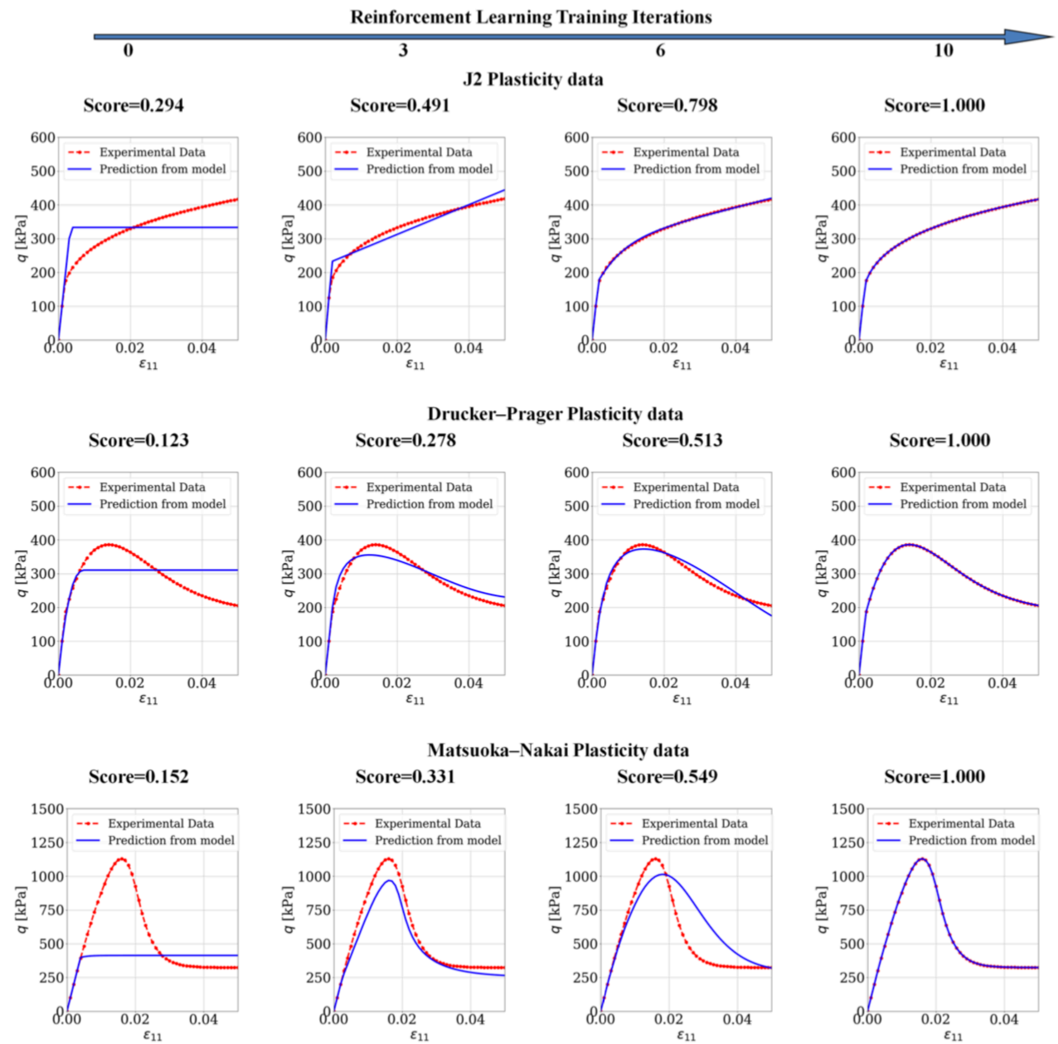




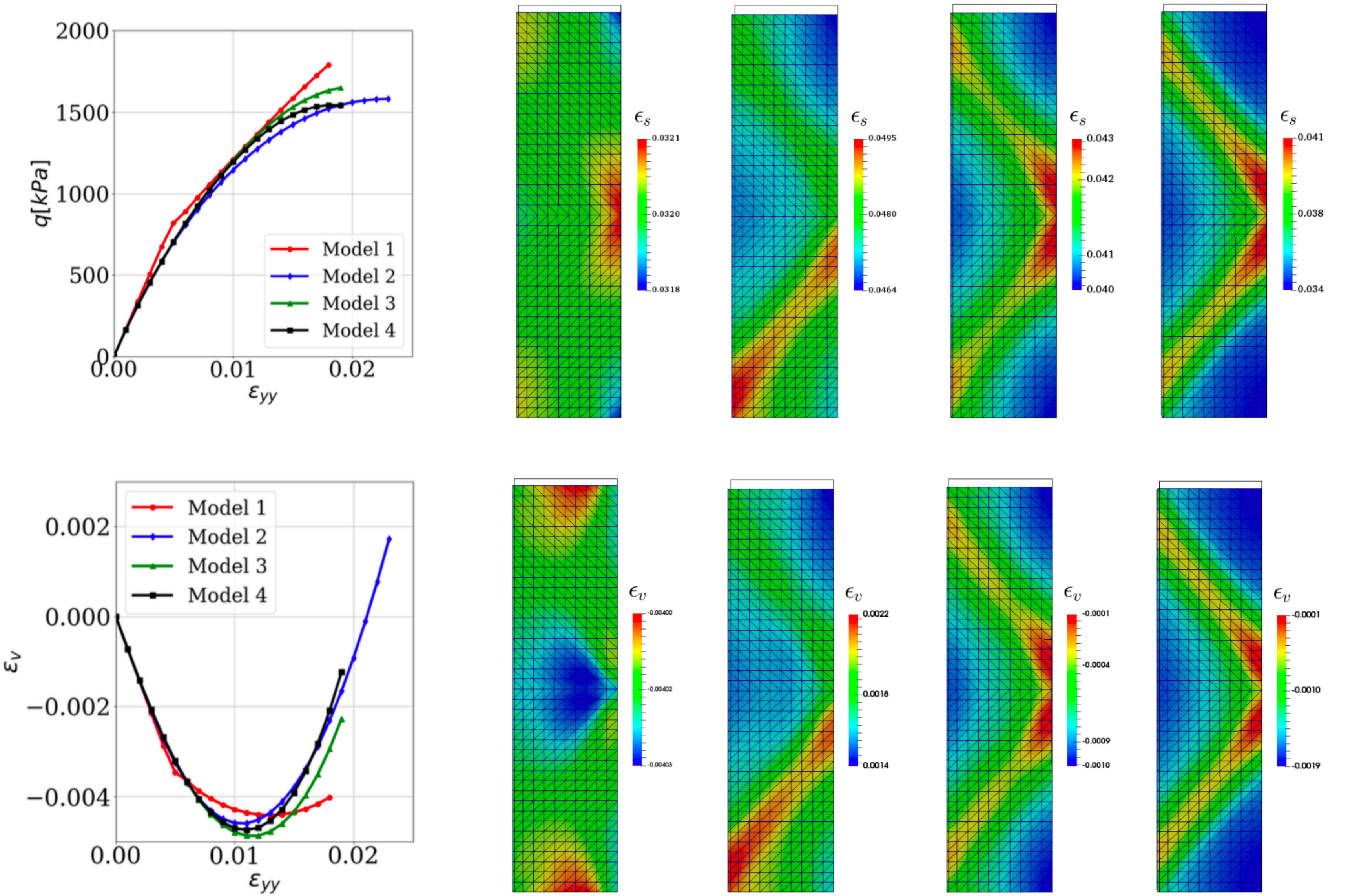
Goal: given a type of data and check whether the AI can generate the right constitutive law







Validation exercise 3: Blind predictions

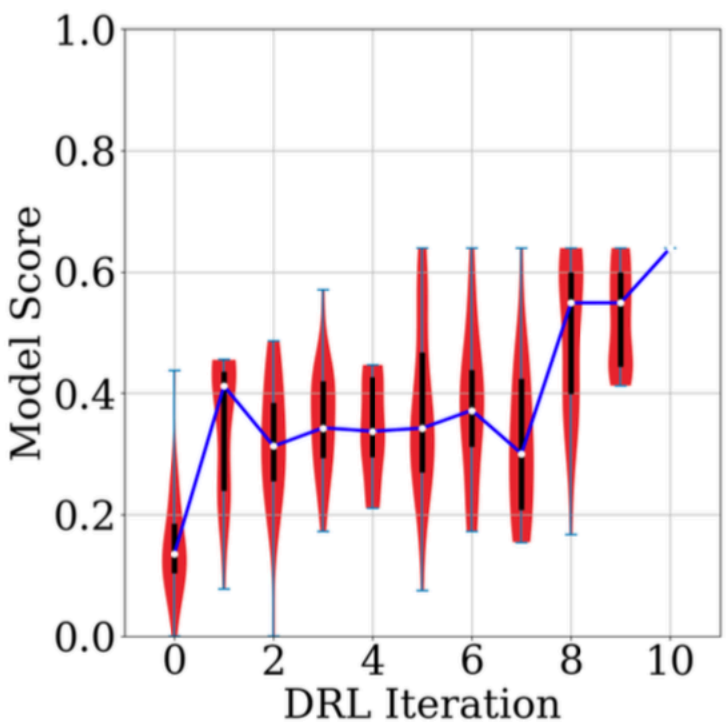


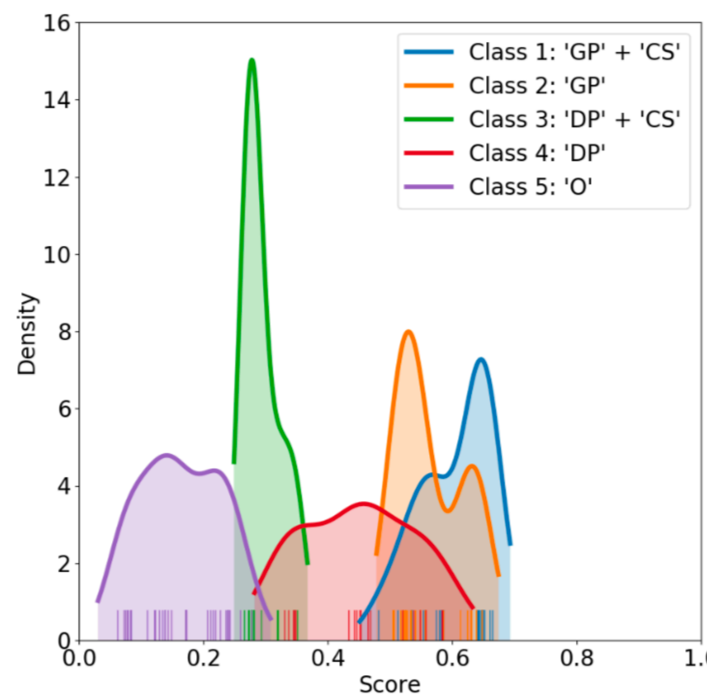
1st iteration | 5th iteration | 8th iteration | Benchmark

How does AI perform compared to human players/modelers/experimentalists?

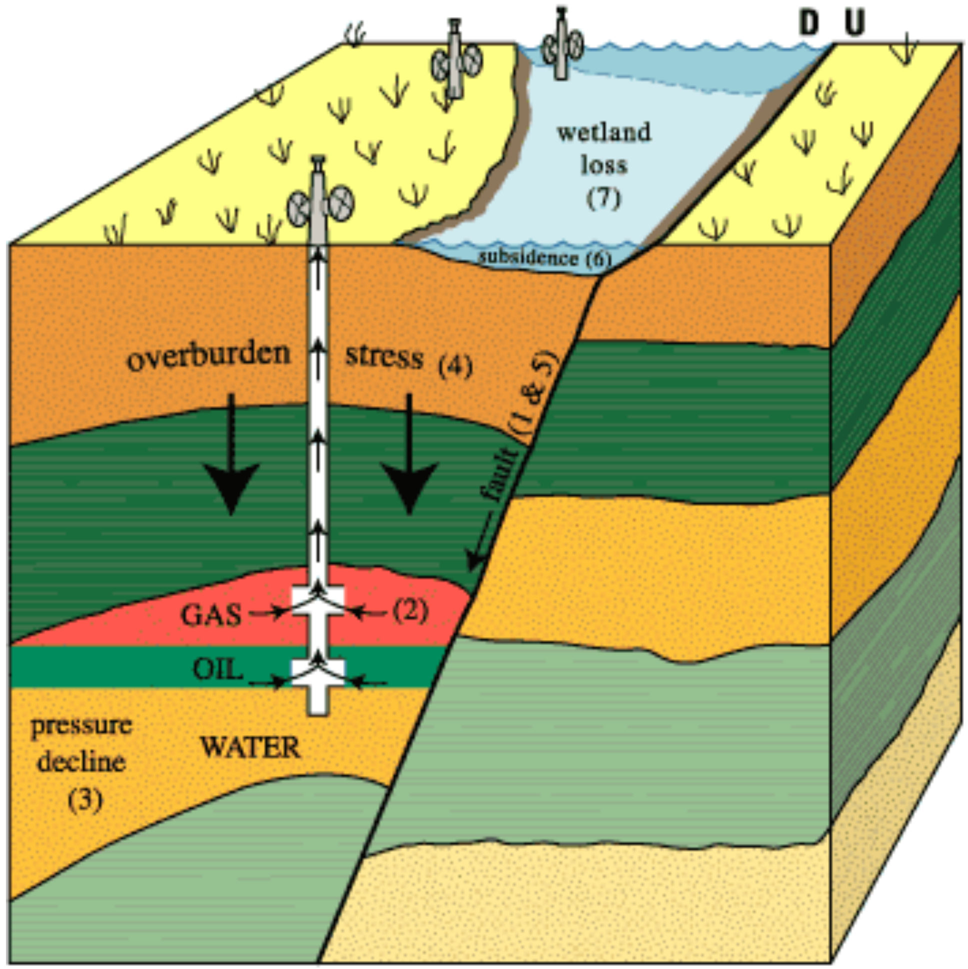
Post-game analysis: Performance in blind predictions (soil critical state plasticity)

Model Class	Number of Models	Mean Score	Standard deviation	Generalized Plasticity 'GP'	Critical State 'CS'	Classical pressure dependent elasto-plasticity 'DP'	Others 'O'
1	22	0.603	0.054	✓	✓		
2	25	0.565	0.051	✓			
3	13	0.295	0.028		✓	✓	
4	19	0.450	0.086			✓	
5	33	0.163	0.063				✓

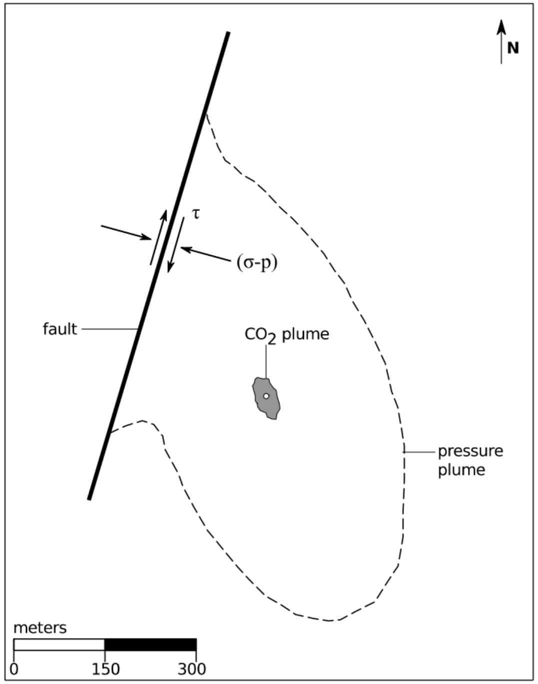




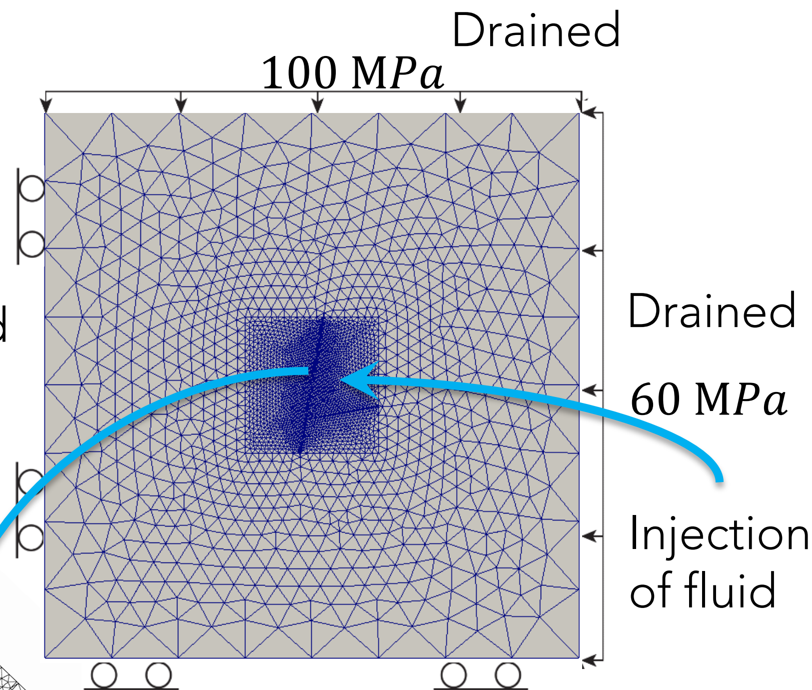
Numerical Example: Reactivation of dual-porosity fault



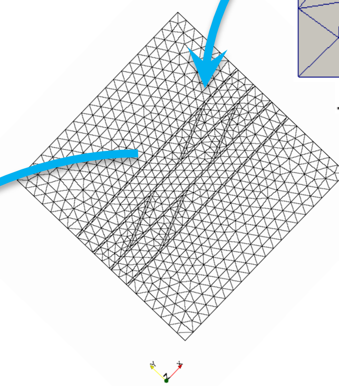
Application 1: Reactivation of dual-porosity fault



Field applications

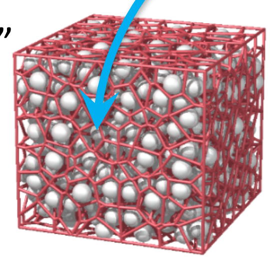


Idealized macroscopic dual-porosity problem with no build-in constitutive law



Assumed strain embedded strong dis-contunity problem serves as ”trainer” for Macro-Meso ANN that generates macroscopic dual porosity responses

DEM-network model are serve as “trainner” for Meso-micro ANN that generates the responses of joints and micro-fracture

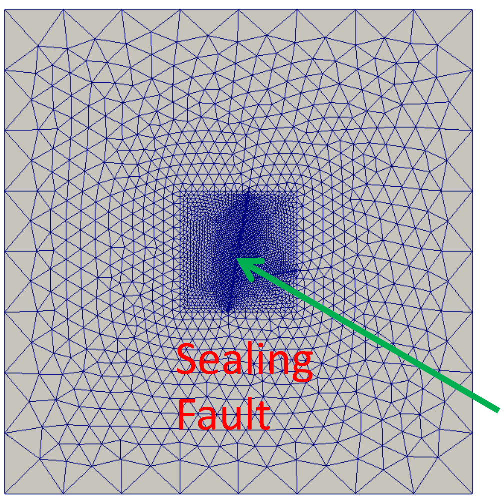


Wang & Sun, CMAME, 2018, 2019a, 2019b

From multi-scale simulation to multiscale training (small strain) – reactivation of sealing fault

Macro-scale simulation with off-line trained material models

$100\ \mathrm{M}Pa$ Drained



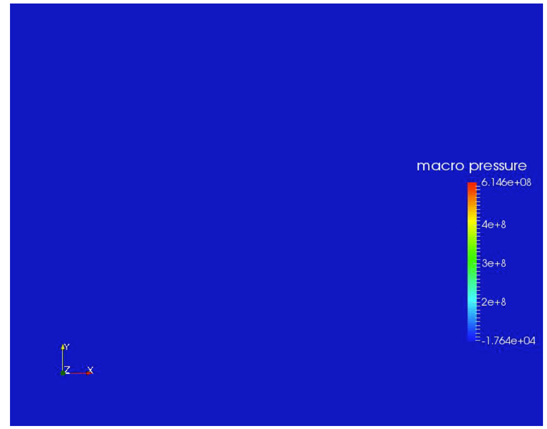
Drained

$60\ \mathrm{M}Pa$

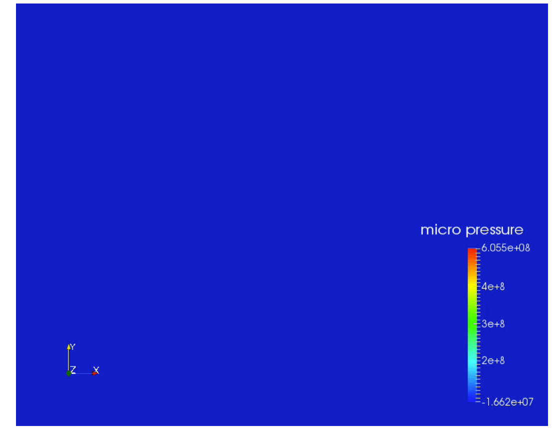
Injection of fluid

Drained

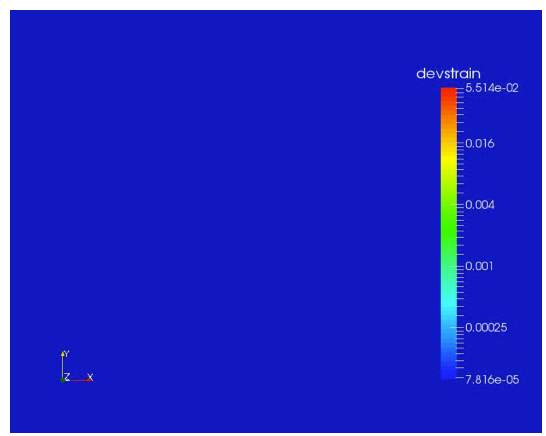
- Constitutive laws of the embedded strong discontinuities generated from training against RNN- DEM data (or meso-scale test data if available)



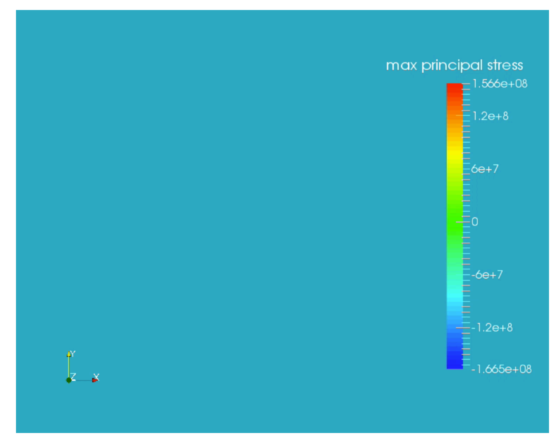
Macro-pore pressure



Micro-pore pressure



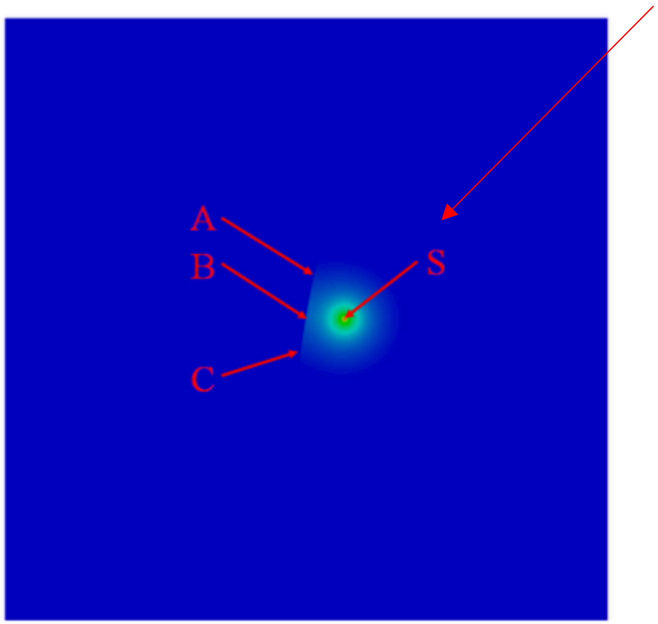
Deviatoric strain

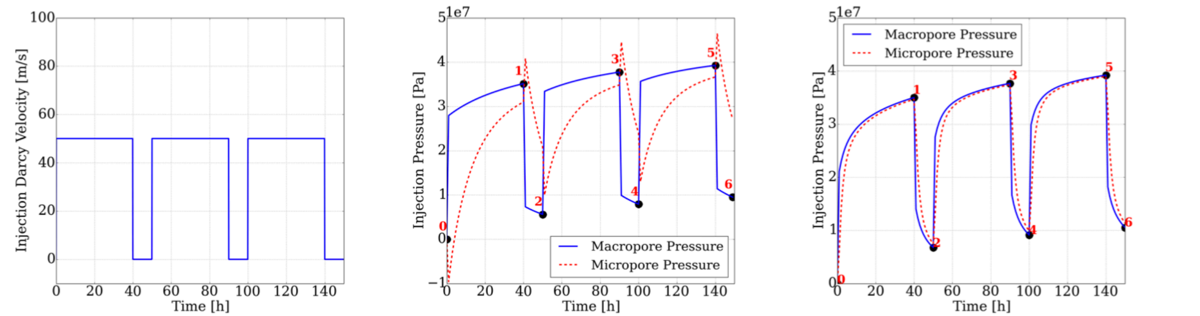


Maximum principal stress

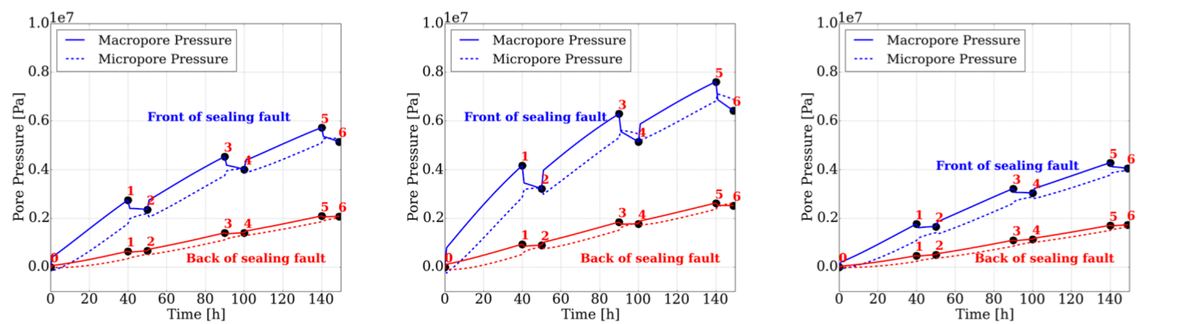
From multi-scale simulation to multiscale training (small strain) – reactivation of sealing fault

Macro-scale simulation with off-line trained material models

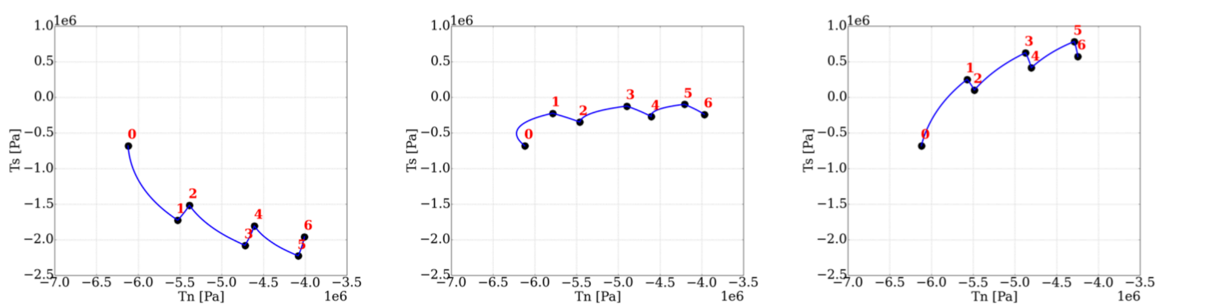




(a) **(b)** Low transfer between pores **(c)** High transfer between pores



(a) A **(b)** B **(c)** C



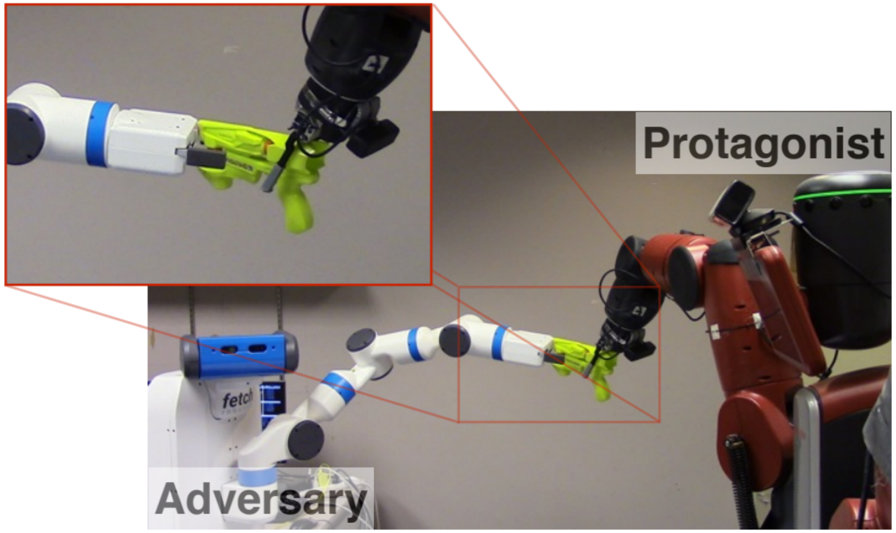
(a) A **(b)** B **(c)** C

Wang & Sun, CMAME, 2018

Future work?

Adversarial deep reinforcement Learning

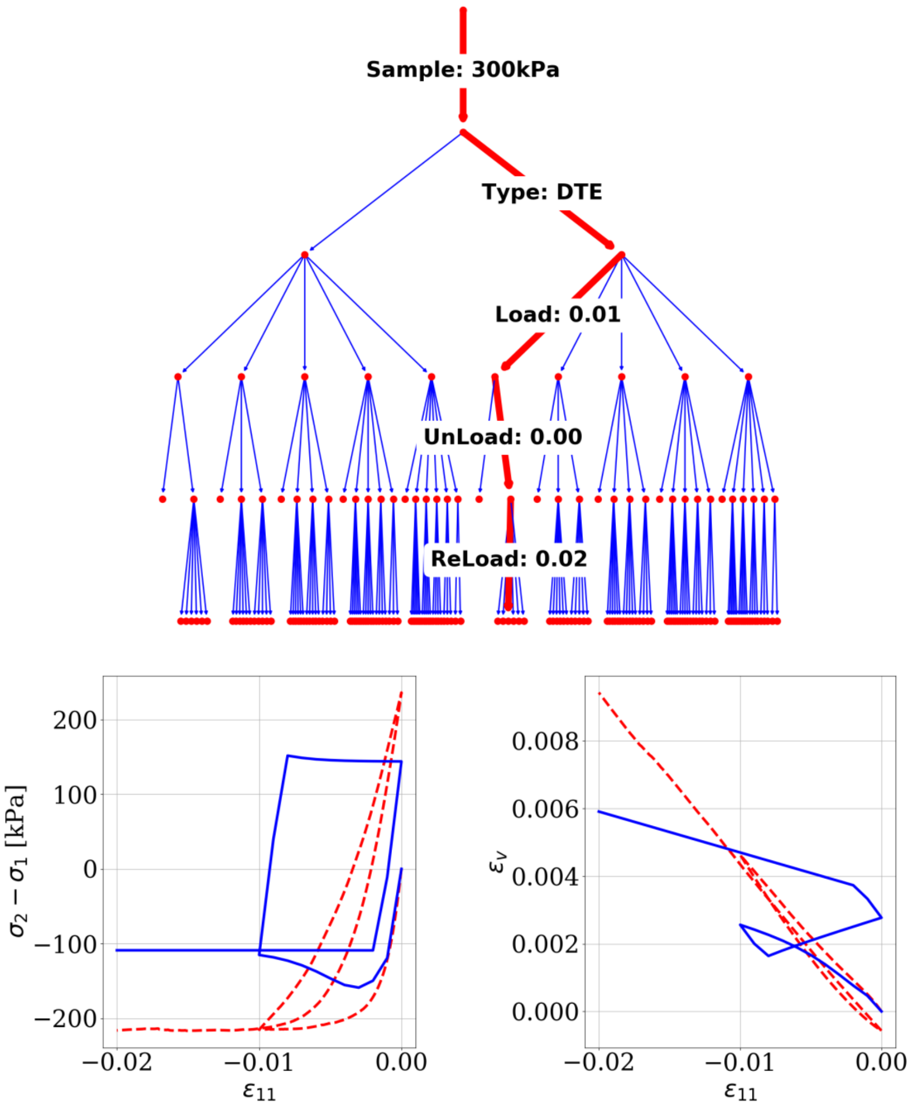
Example of adversarial learning:
Adversarial framework for effective self-supervised learning on grasp policy in robotics



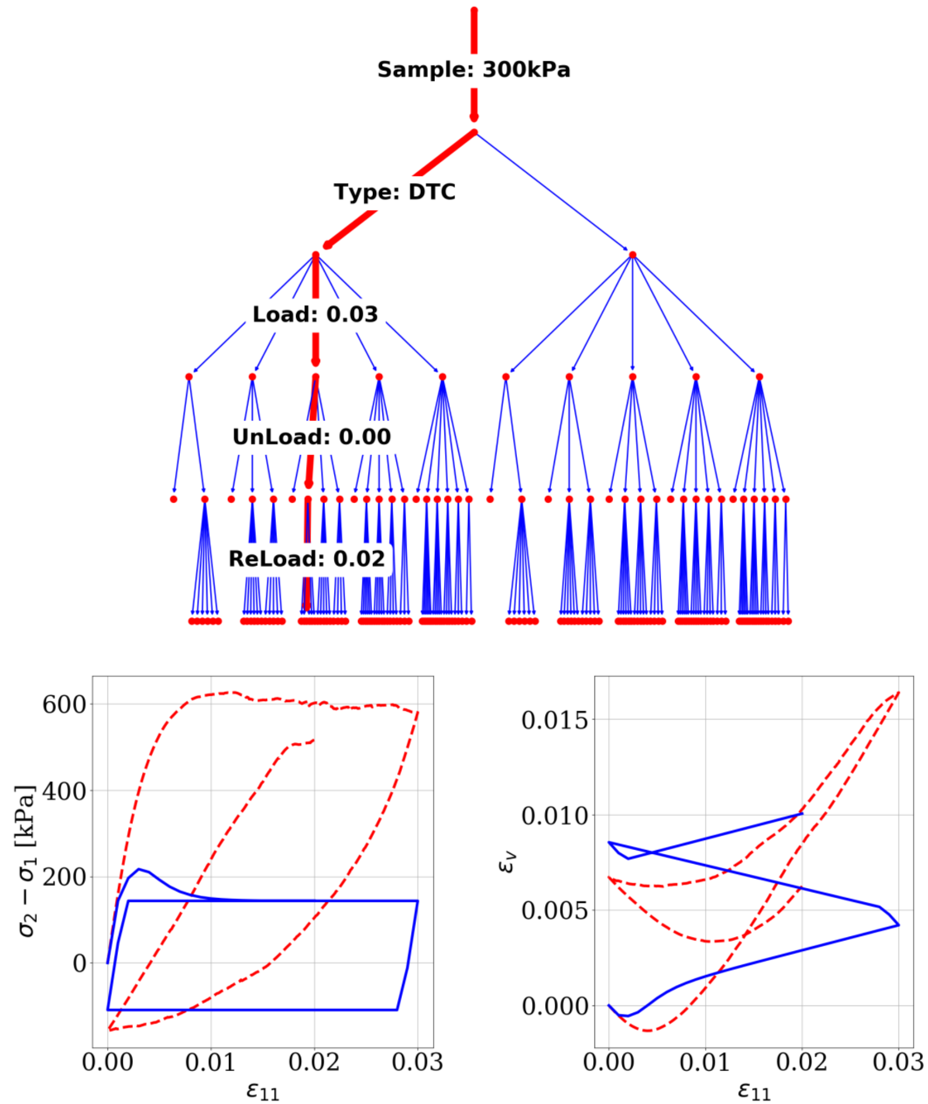
Pinto, Lerrel, James Davidson, and Abhinav Gupta. "Supervision via competition: Robot adversaries for learning tasks." *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

Experimentalist/Adversary Game Training Iteration 0

Decision Tree of Experimentalist Agent



Decision Tree of Adversarial Agent

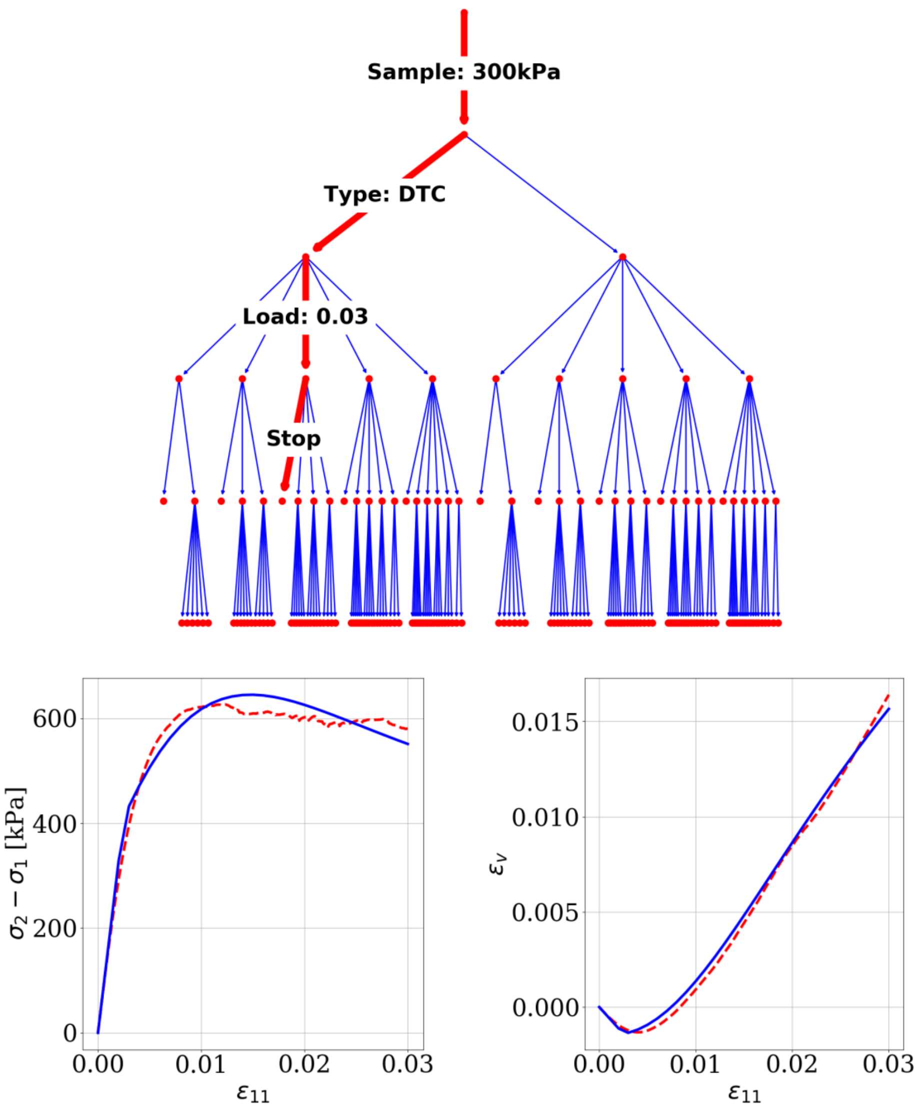


Blue: Prediction from DP model

Red: Data from DEM simulations

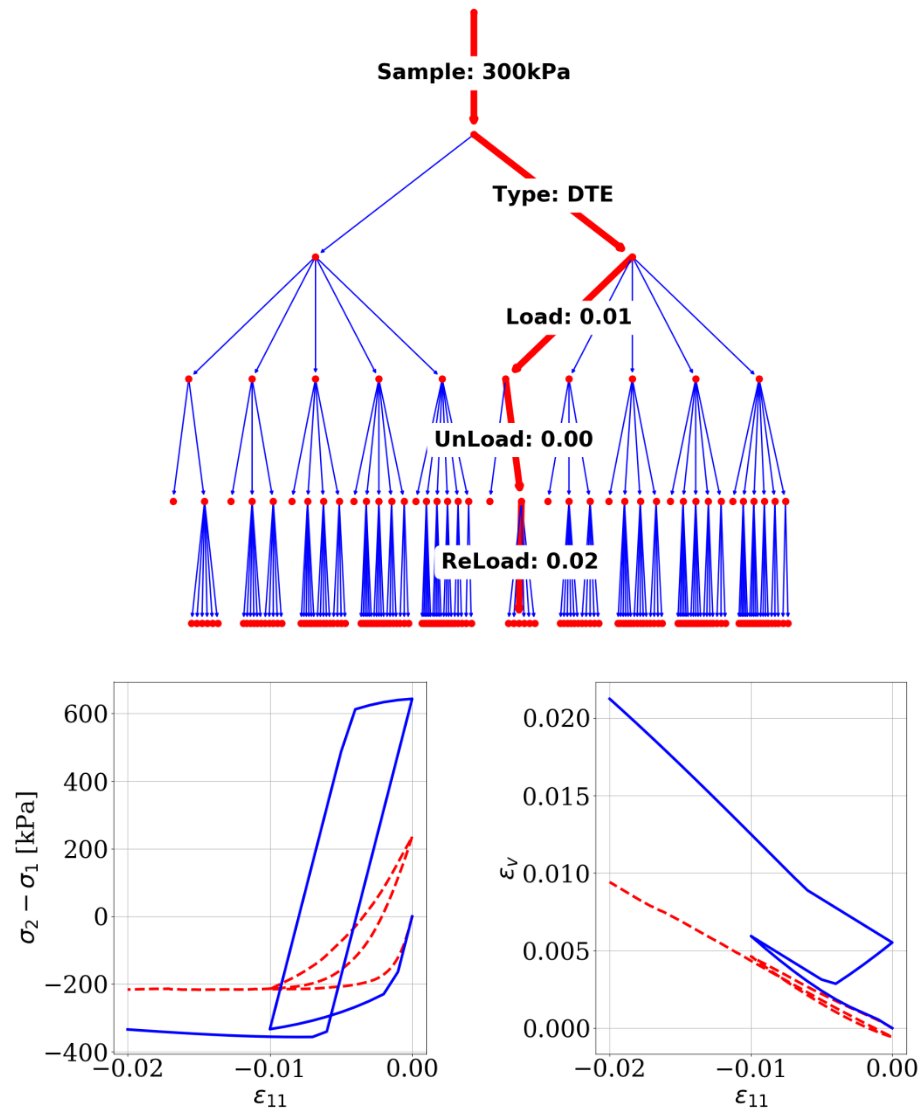
Experimentalist/Adversary Game Training Iteration 6

Decision Tree of Experimentalist Agent



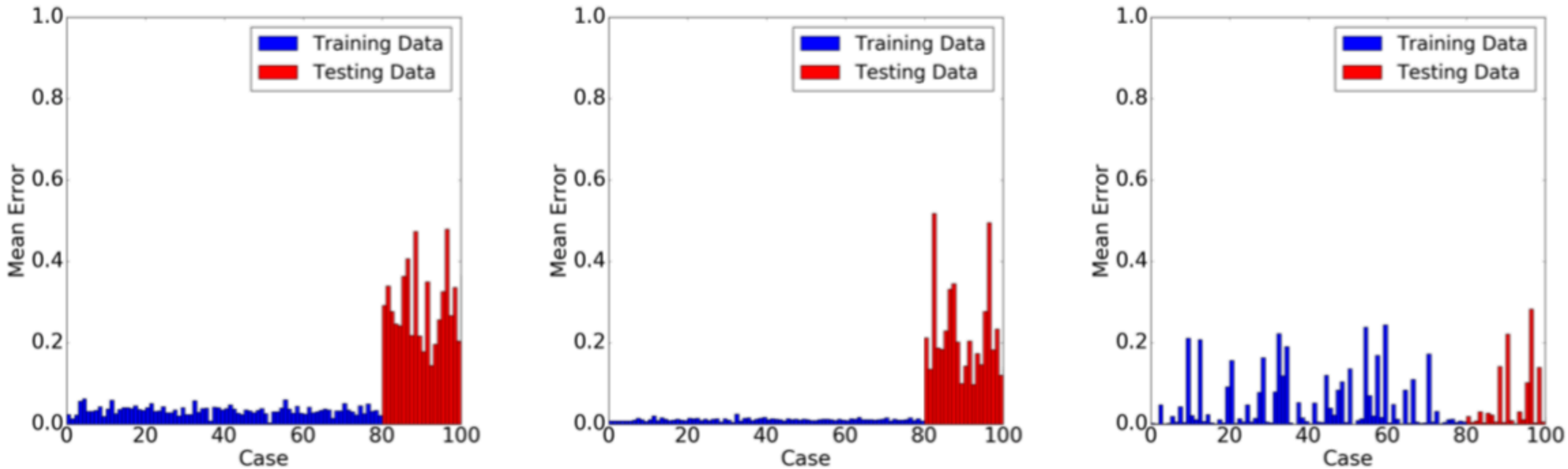
Blue: Prediction from DP model

Decision Tree of Adversarial Agent



Red: Data from DEM simulations

Final Remarks: Blind Prediction vs. calibration – overfitting vs. underfitting



(a) (b) (c)

1st IACM Special Interest Conference on Machine Learning and Digital Twins for Computational Science and Engineering Conference 2021

iacm

Conference co-chairs:
Wing Kam Liu (Northwestern)
JS Chen (UC San Diego)
George Karniadakis (Brown)
Charbel Farhat (Stanford)
Francisco Chinesta (ParisTech)
WaiChing Sun (Columbia)

Conference tracks: Digital Twins / Big Data and Machine Learning / Advanced Manufacture and Design / Multiscale Materials and Engineering System / Bio-systems, Medial Device and ML-enhanced diagnostics / Reduced-order modeling for fluid, solids and structures / Computer graphics, gaming and ML-specific hardware, Tensor Processing Unit and TensorCore / Geosystem, geostatistics and petroleum engineering/ Education, outreach, short courses, funding opportunity panels and public lectures

WCCM Paris Short Course on graph-based machine learning with open source codes

iacm

14th WCCM
& ECCOMAS Congress 2020

Paris, 19-24 July 2020

ECCOMAS

Reference

1.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.

2.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.

3.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, doi:,10.1007/s00466-019-01723-1, 2019.