A Multiscale Meta-Modeling Game For Fluid-infiltrating Porous Media

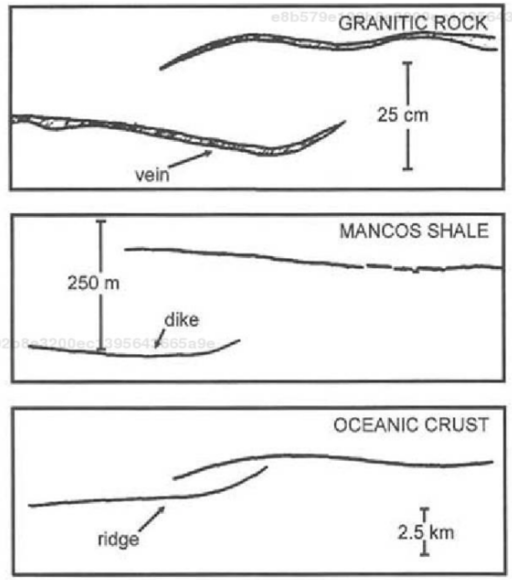
Kun Wang[a], WaiChing Sun[a], Qiang Du[b]

Department of Civil Engineering and Engineering Mechanics[a]
Department of Applied Physics and Applied Mathematics, Data Science Institute[b]
Fu Foundation School of Engineering and Applied Science,
Columbia University, New York, USA

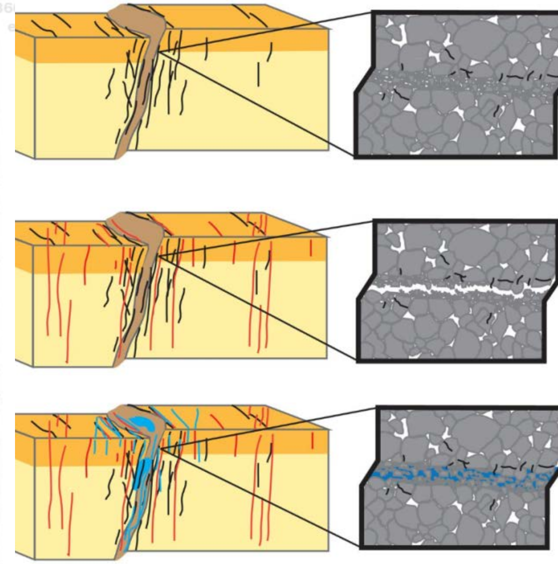
NYSDS, Columbia University, 2019

Columbia | Engineering
The Fu Foundation School of Engineering and Applied Science

Multi fracture sizes in geomechanics (multiple length & time scales)

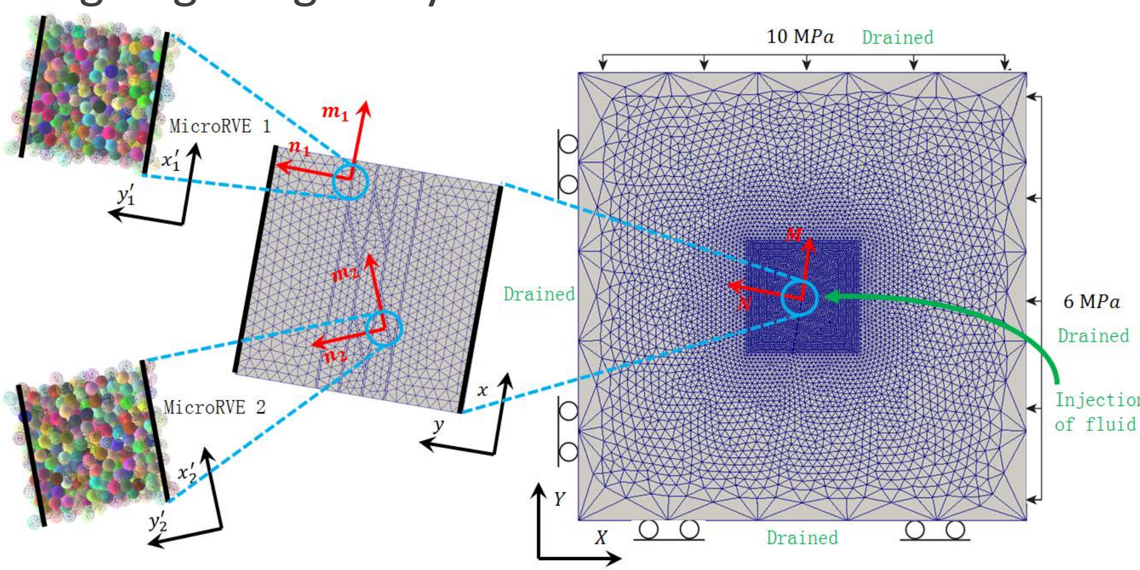


[Pollard & Aydin, 1984]



[Skurtveit et al., 2015]

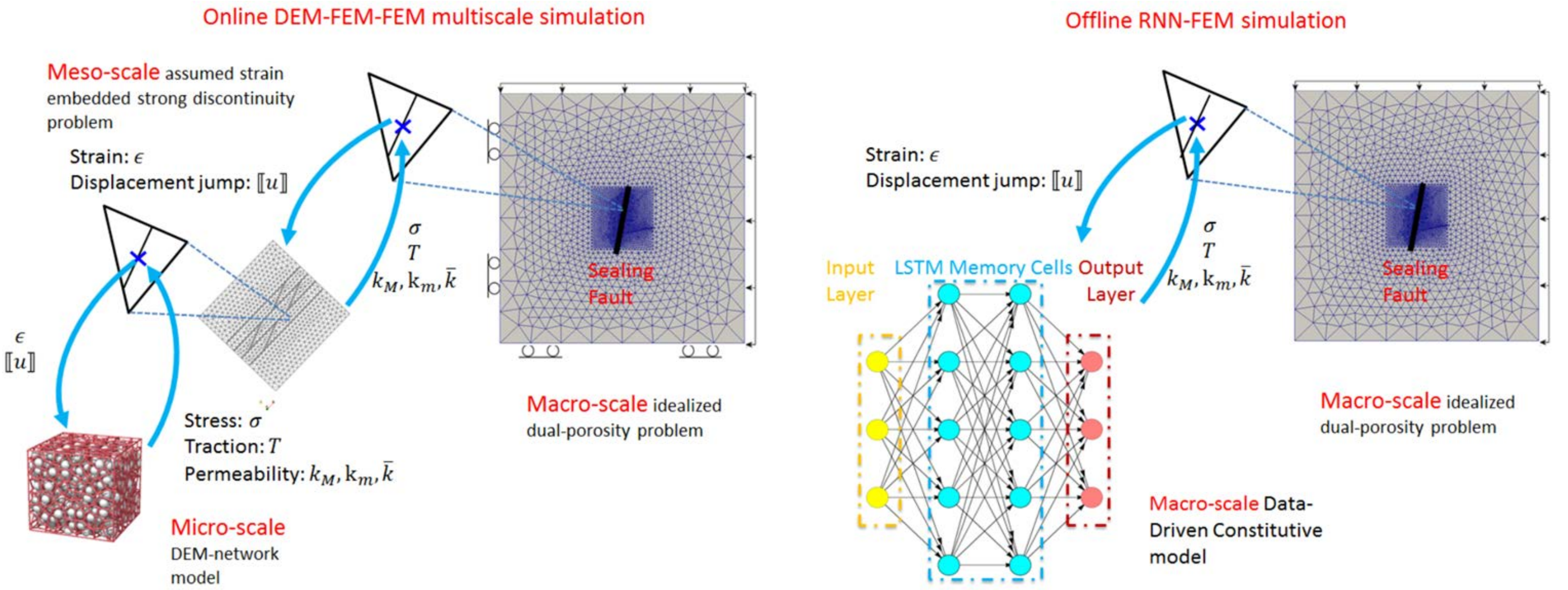
Multi-scale Modeling of geological systems



[Wang & Sun, 2018]

Motivation and Background

Surrogate models of constitutive behaviors across multiple length scales

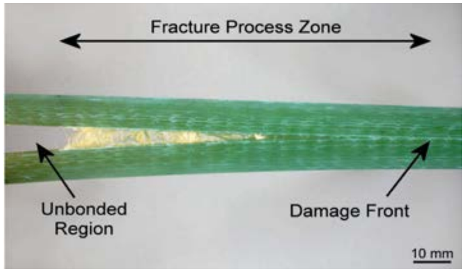


Surrogate models for upscaling can be theory-based models with hand-crafted mathematical expressions, or data-driven models with neural networks (as universal function approximators) learning from data.

[Wang & Sun, 2018]

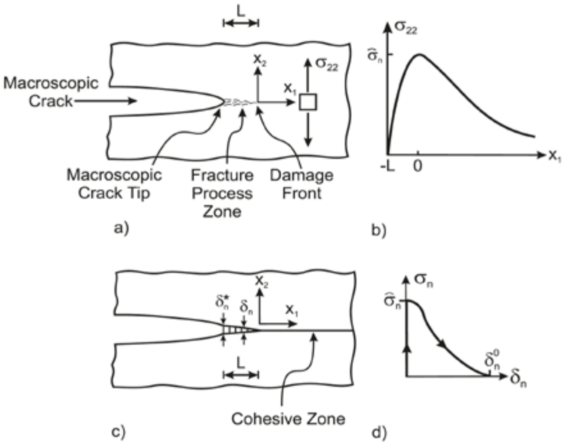
Examples of Material Laws

Traction-separation laws

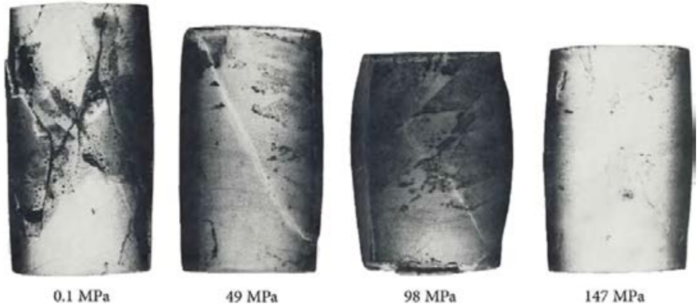


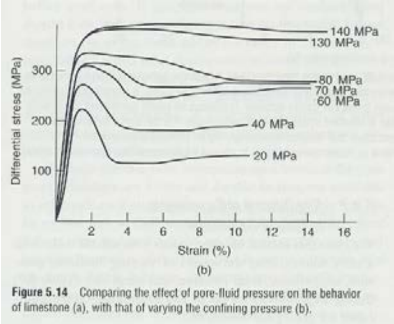
Crack growth in composite

San Andreas fault

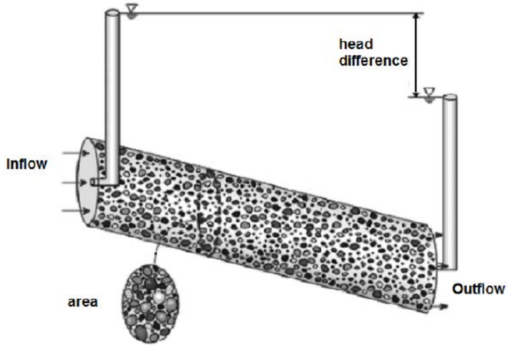


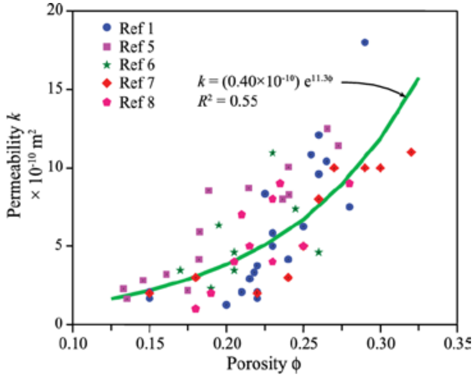
Stress-strain relationship





Porosity-permeability relationship





Accelerate scientific discovery using machine learning

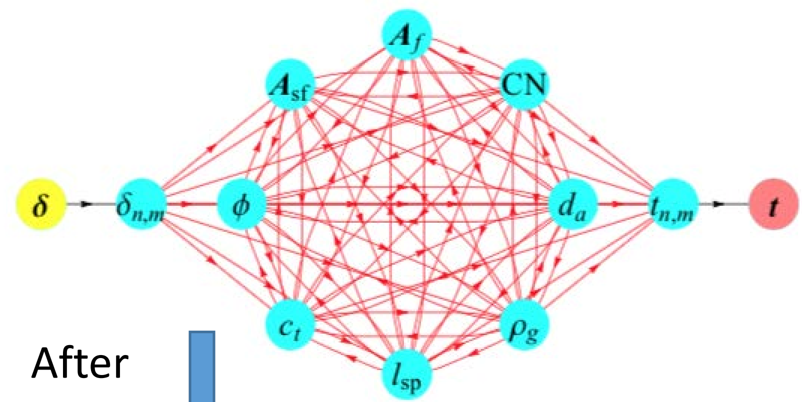
- Mohr-Coulomb 1770s, 1880s
- Von Mises J2 plasticity 1910s
- Drucker-Prager 1950s
- Critical state soil mechanics 1960s
- Sand model with fabric tensor 2004
- …

Dependence on mean effective pressure

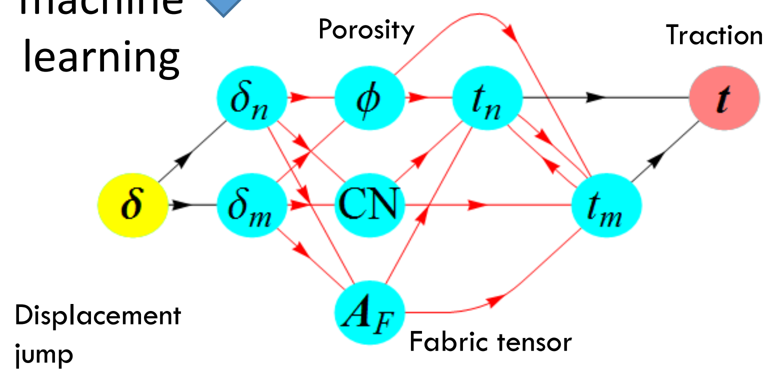
Dependence on void ratio

Dependence on fabric tensor

Knowledge graph from domain experts



After machine learning



Discover new mechanisms

Machine Learning on knowledge graphs:
Nickel, Maximilian et al. (2015),
Battaglia, Peter W., et al. (2018), …

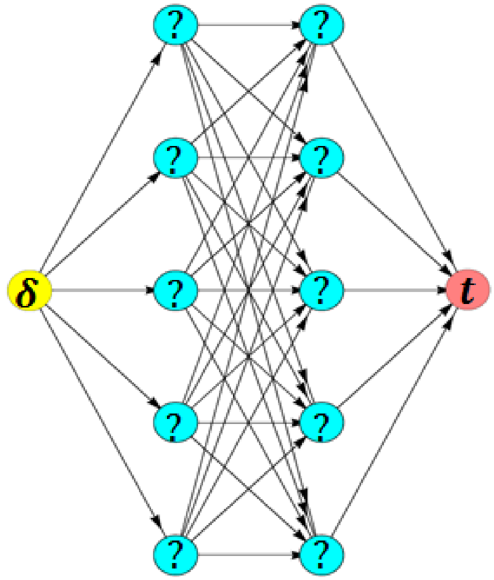
Why machine learning for constitutive modeling?

Key Idea: Use DRL to generate knowledge graph

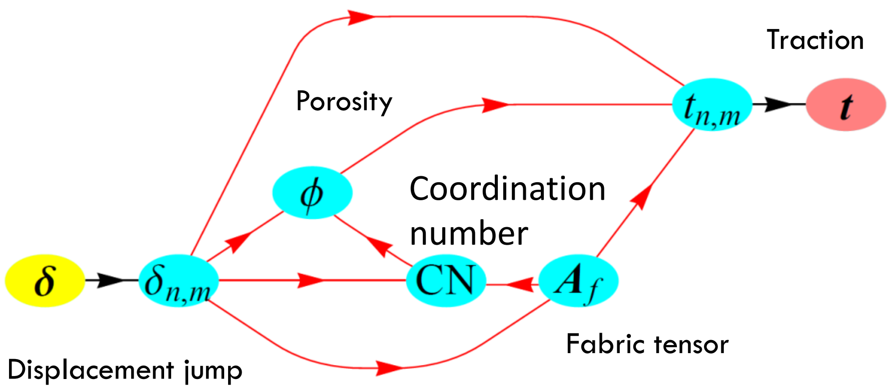
Machine Learning focusing on internal properties

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)*



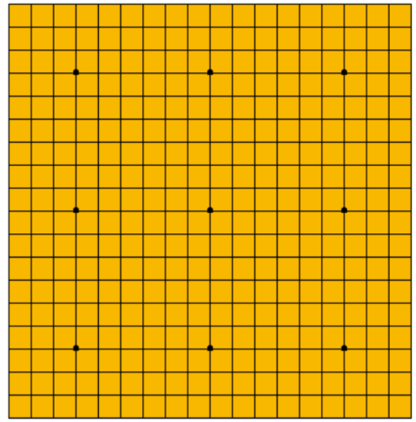
Graph-based predictions – designed to generate knowledge represented by directed graph with the same ***internal properties*** of human thinkers.

Analogy of Constitutive Modeling to Games

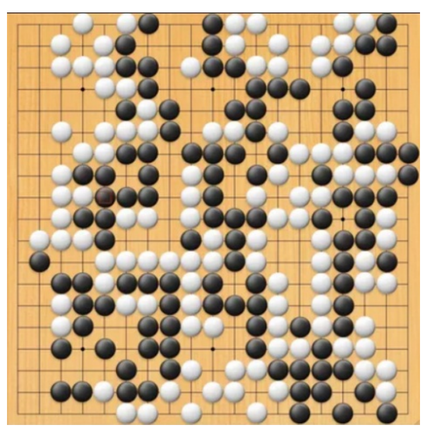
Chess Game

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

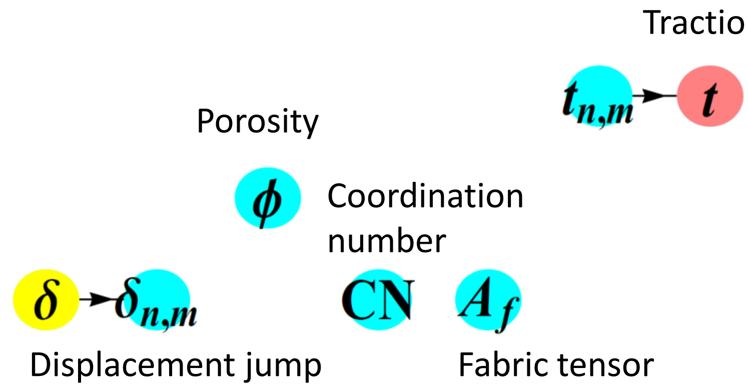
Go Game



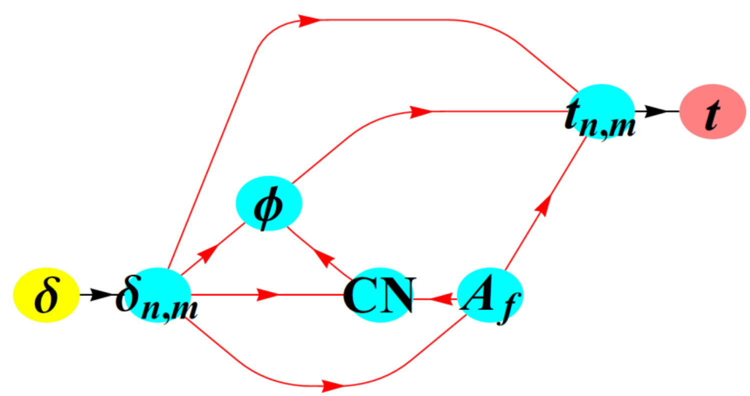
Place pieces to control more territory than your opponent



Meta-modeling Game



Connect edges to generate optimal internal information flow of constitutive models



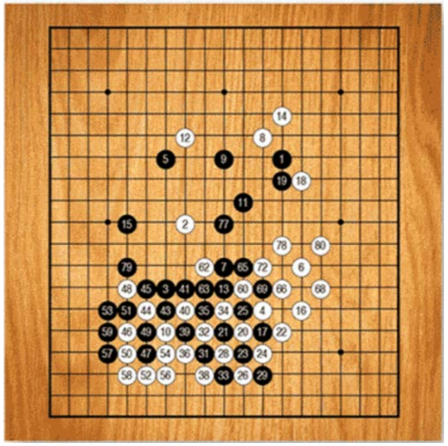
Superhuman Performance of AI in learning the strategies of games

Alpha Go Zero

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

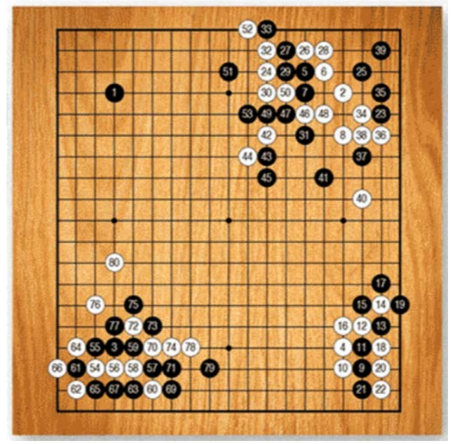
https://deepmind.com/blog/alphago-zero-learning-scratch/

3 hours



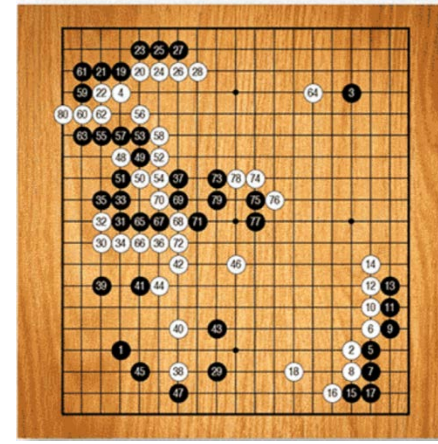
Beginner level with greedy plays

19 hours



Learnt the fundamentals of Go strategies

70 hours



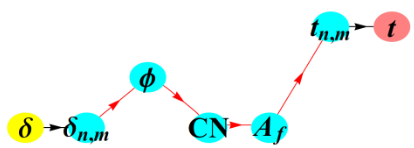
Super-human level with disciplined plays

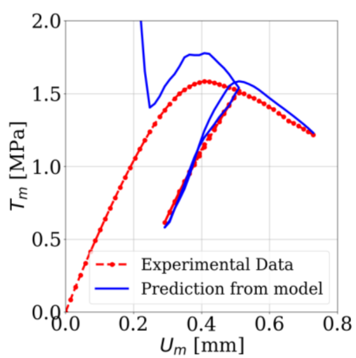
Meta modeling DRL

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

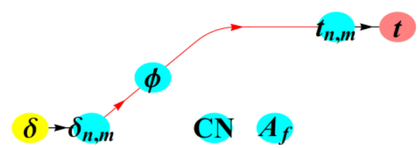
In our example: over 2e4

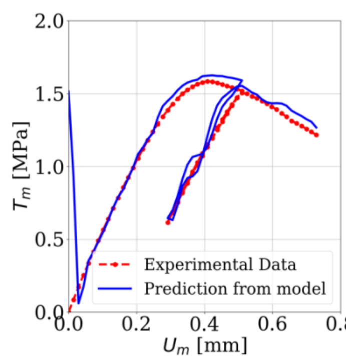
10 games



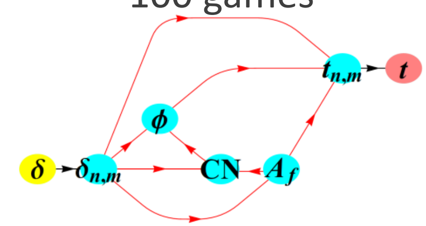


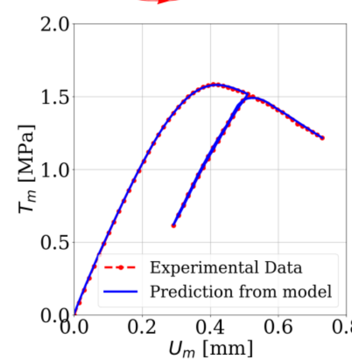
30 games





100 games



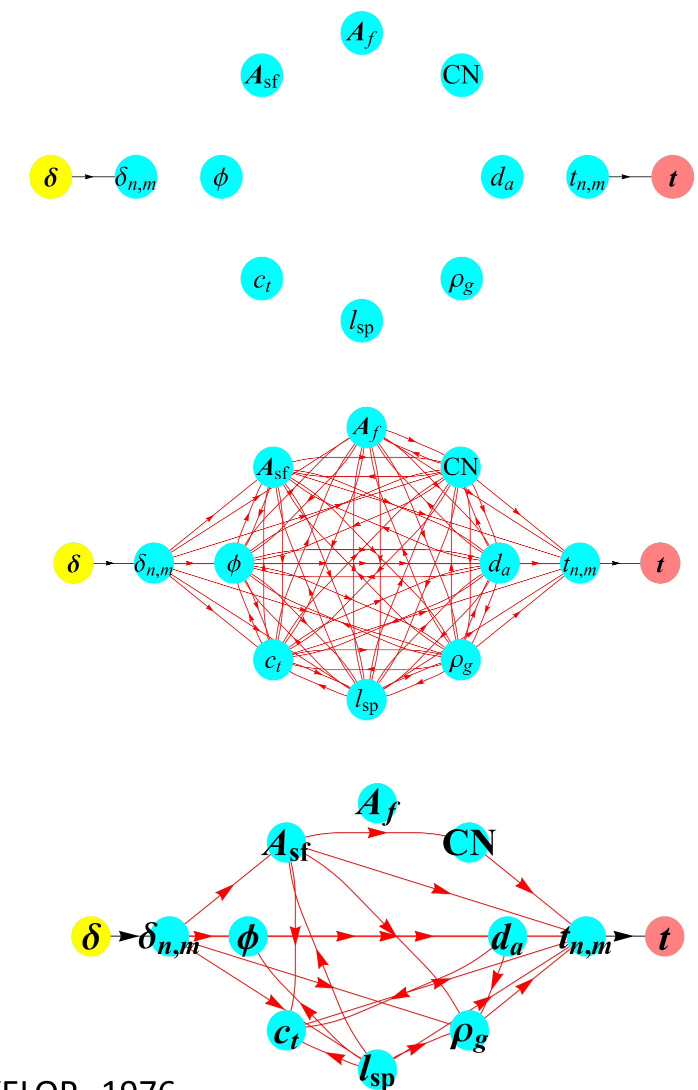


How does the meta-modeling approach work?

Graph representation of knowledge

Representation of mechanics knowledge in graphs, directed Graph and directed multi-graph

- Vertices - a measurable physical properties (permeability, thermal conductivity, force, displacement, strain..etc)
- Directed Edges – a existing hiearcheiral relationship between two vertices
- Edge Labels – the specific model used to connect two physical vertices. The model can be mathematical, neural network, support vector machine …etc
- Directed graph – the ***combinatorial*** optimized configuration of the vertices connected by edges, each with one unique labels.
- Label Directed Multi-graph – all the possible way the vertices are connected by different combination of edges with different labels



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

Leverage hand-crafted models to expand multi-graphs

Use directed multi-graph to represent possible theories and models (Graph representation of knowledge) – Worst case scenario – we recover the best hand-crafted model but we won't generate any new model that performs worse than existing state-of-the-art.

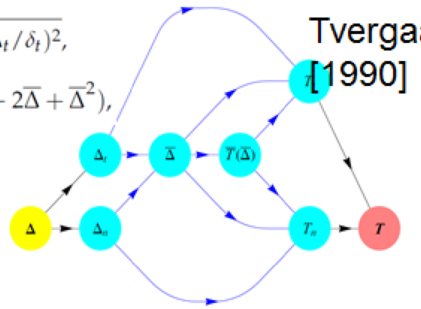
Example: traction-separation models

$$\overline{\Delta} = \sqrt{(\Delta_n/\delta_n)^2 + (\Delta_t/\delta_t)^2},$$

$$\overline{T}(\overline{\Delta}) = \frac{27}{4}\sigma_{\max}\overline{\Delta}(1 - 2\overline{\Delta} + \overline{\Delta}^2),$$

$$T_n = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\frac{\Delta_n}{\delta_n},$$

$$T_t = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\alpha\frac{\Delta_n}{\delta_t}$$



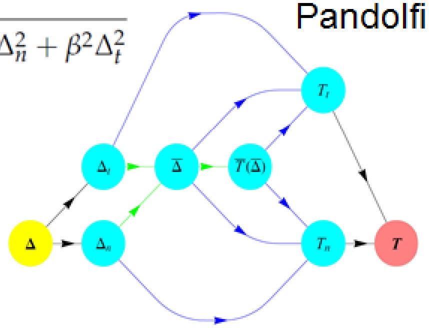
Tvergaard [1990]

$$\overline{\Delta} = \tilde{\Delta}/\delta_n\,,\ \tilde{\Delta} = \sqrt{\Delta_n^2 + \beta^2\Delta_t^2}$$

$$\overline{T}(\overline{\Delta}) = k\overline{\Delta} + c$$

$$T_n = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\frac{\Delta_n}{\delta_n},$$

$$T_t = \frac{\overline{T}(\overline{\Delta})}{\overline{\Delta}}\alpha\frac{\Delta_n}{\delta_t}$$

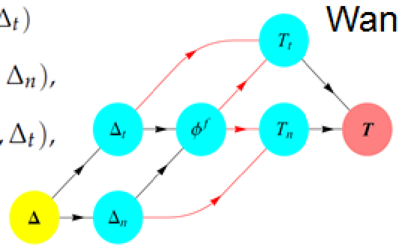


Pandolfi et al. [1999]

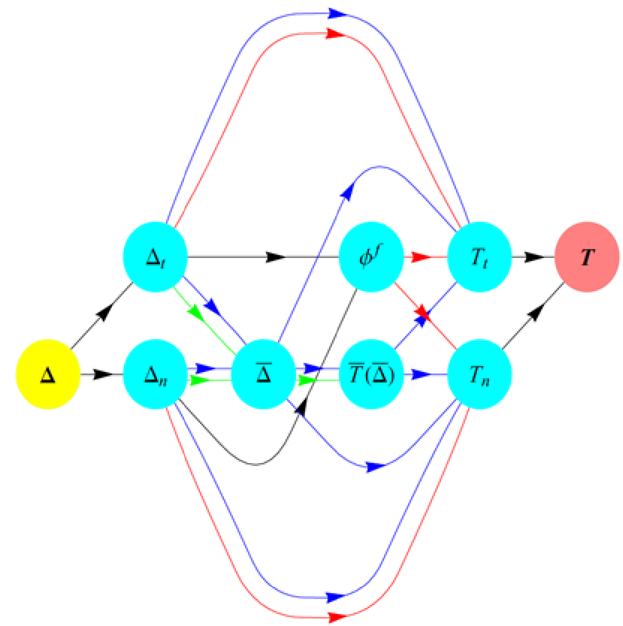
$$\phi^f = \phi_0^f(1 + \Delta_n\Delta_t)$$

$$T_n = f^{\mathrm{LSTM}}(\phi^f, \Delta_n),$$

$$T_t = g^{\mathrm{LSTM}}(\phi^f, \Delta_t),$$



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]

Find optimal directed graph for a constitutive model from directed multi-graph

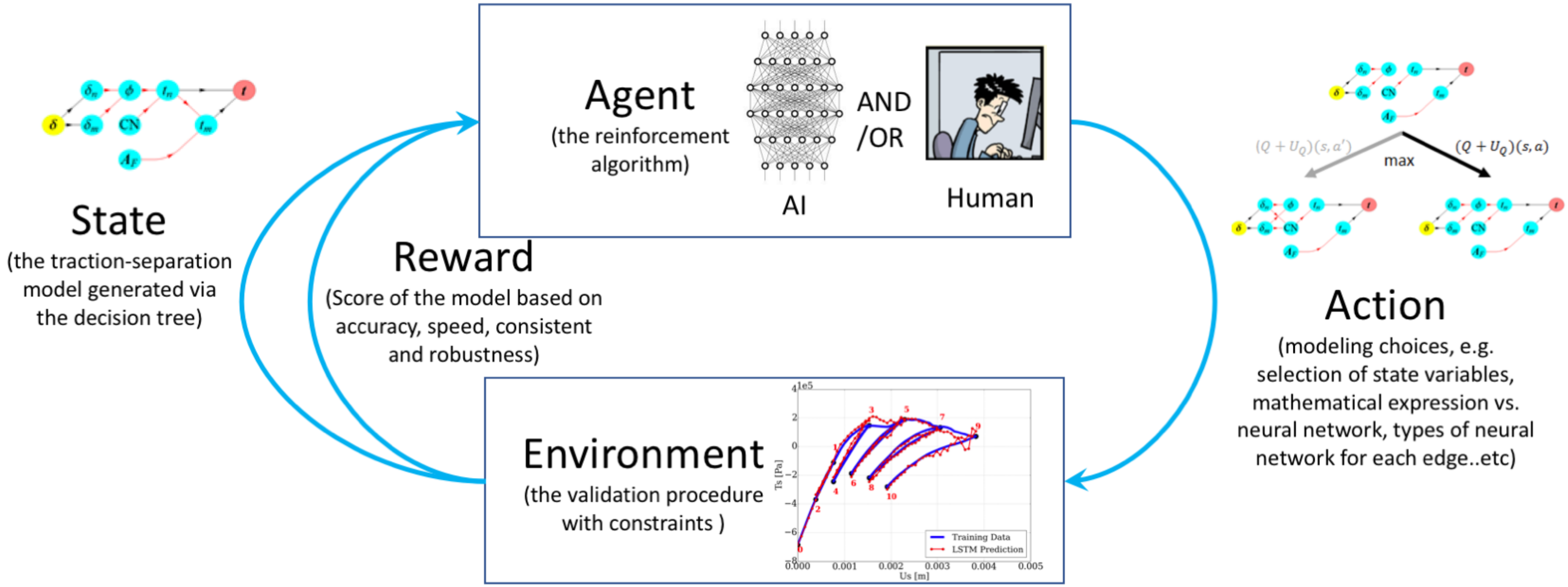
Formalization of the meta-modeling game in graph theory

Possible configurations of constitutive laws as a labeled directed multi-graph. Given a data set which measures a set of physical quantities defined as $\mathbb{V}$ with a corresponding set of labels $\mathbb{L}_{\mathbb{V}}$ where $n_{\mathbb{V}} : \mathbb{V} \rightarrow \mathbb{L}_{\mathbb{E}}$ is a bijective mapping that maps the vertices to the labels. Let $\mathbb{V}_R \subset \mathbb{V}$ and $\mathbb{V}_L \subset \mathbb{V}$ be the root and leave of the directed multi-graph. All possible ways to write constitutive laws that map the input V_R (e.g. strain history) to output V_L (e.g. stress) as information flow can be defined by the sets of edges where each edge that links two physical quantities $\mathbb{E}$, the mappings $s : \mathbb{E} \rightarrow \mathbb{V}$ and $t : \mathbb{E} \rightarrow \mathbb{V}$ that provide the direction of the information flow, and the surjective mapping $n_{\mathbb{E}} : \mathbb{E} \rightarrow \mathbb{L}_{\mathbb{E}}$ that assigns the edge labels (names) to the edges.

Instants of constitutive laws as direct-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 $\mathbb{G} = (\mathbb{L}_{\mathbb{V}}, \mathbb{L}_{\mathbb{E}}, \mathbb{V}, \mathbb{E}, s, t, n_V, n_E)$, and objective function SCORE and constraints to enforce universal principles. Find an subgraph $\mathbb{G}'$ of $\mathbb{G}$ consists of vertices $V \in \mathbb{V}^s \subseteq \mathbb{V}$ and edges $E \in \mathbb{E}^s \subseteq \mathbb{E}$ 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.,

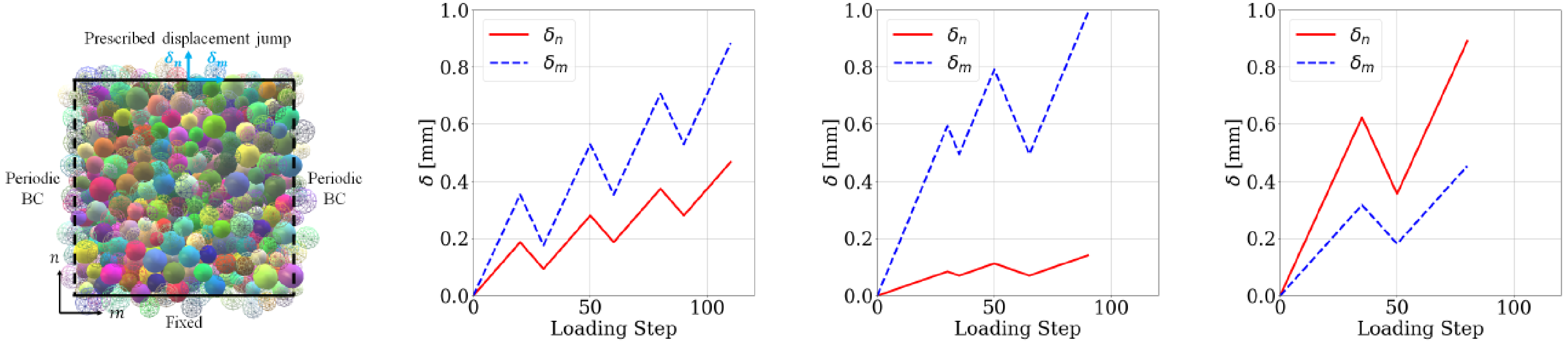
$$\begin{aligned} \underset{l_i}{\text{maximize}} \quad & \text{SCORE}(l_1, l_2, \ldots, l_n) \\ \text{subject to} \quad & f_i(i_i) = 0, \ i = 1, \ldots, m. \end{aligned} \tag{17}$$

Key ingredients of a game

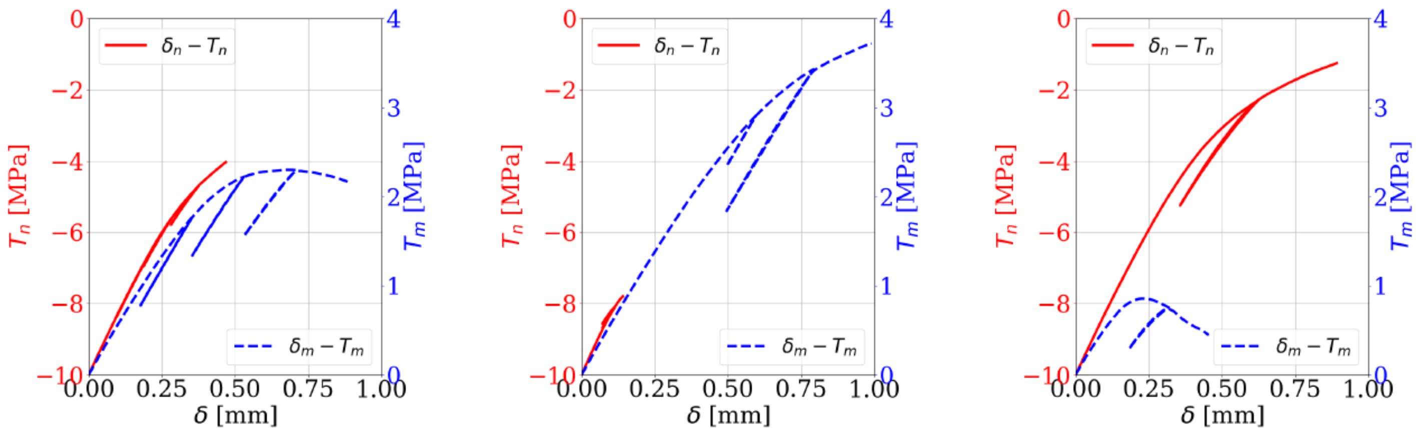


Environment	Idealized multigraph for constitutive models validated against unseen data
Agent	Human or AI
State s	The generated constitutive laws
Action a	The decisions that lead to the generation of constitutive laws
Reward r	Score (objective function) of the constitutive model
$v(s)$	Expected model 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

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

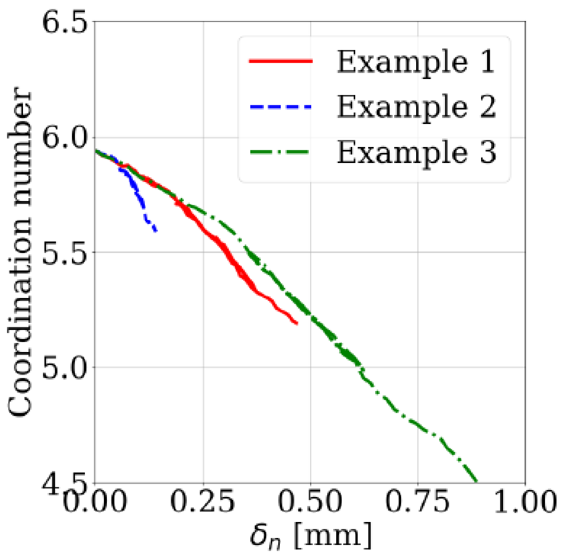


(a) RVE of frictional surface **(b)** Example loading path 1 **(c)** Example loading path 2 **(d)** Example loading path 3

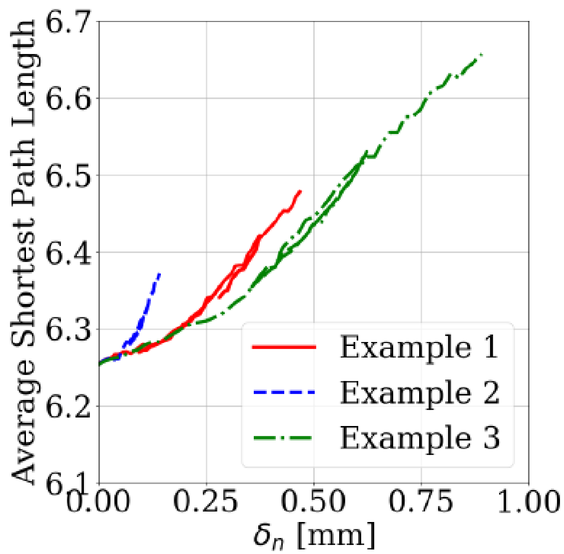


(a) Example loading path 1 **(b)** Example loading path 2 **(c)** Example loading path 3

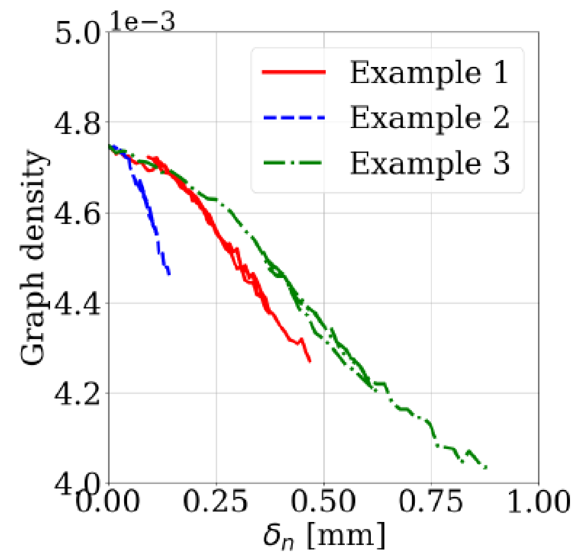
Internal properties (nodes of the directed graphs) may include porosity, coordination number, fabric tensor, and quantitative measures of the graph of grain contact connectivity (e.g., average shortest path, graph density)



(a) Coordination number



(b) Average Shortest Path Length

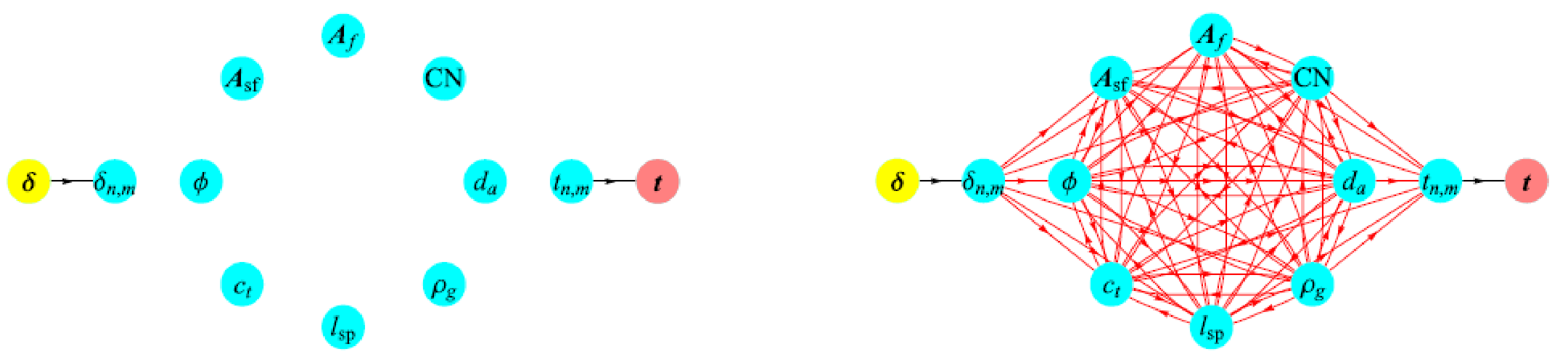


(c) Graph density

Game Board

Possible configurations of constitutive laws as a labeled directed multi-graph. Given a data set which measures a set of physical quantities defined as $\mathbb{V}$ with a corresponding set of labels $\mathbb{L}_{\mathbb{V}}$ where $n_{\mathbb{V}} : \mathbb{V} \rightarrow \mathbb{L}_{\mathbb{E}}$ is a bijective mapping that maps the vertices to the labels. Let $\mathbb{V}_R \subset \mathbb{V}$ and $\mathbb{V}_L \subset \mathbb{V}$ be the root and leave of the directed multi-graph. All possible ways to write constitutive laws that map the input V_R (e.g. strain history) to output V_L (e.g. stress) as information flow can be defined by the sets of edges where each edge that links two physical quantities $\mathbb{E}$, the mappings $s : \mathbb{E} \rightarrow \mathbb{V}$ and $t : \mathbb{E} \rightarrow \mathbb{V}$ that provide the direction of the information flow, and the surjective mapping $n_{\mathbb{E}} : \mathbb{E} \rightarrow \mathbb{L}_{\mathbb{E}}$ that assigns the edge labels (names) to the edges.

Example game

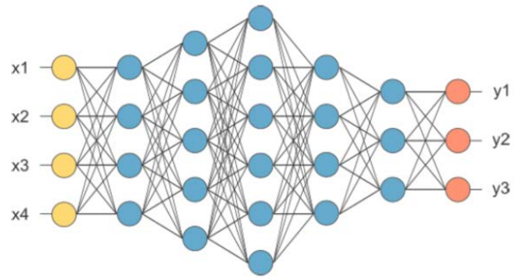


(a) Initial configuration of the "game board"

(b) All possible actions on the "game board"

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

Multilayer perceptron

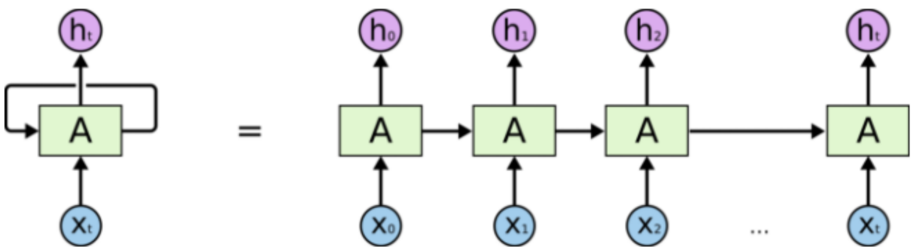


[J Ghaboussi et al. 1991]

[M Lefik and BA Schrefler. 2003]

Treating path-dependent behavior is non-trivial

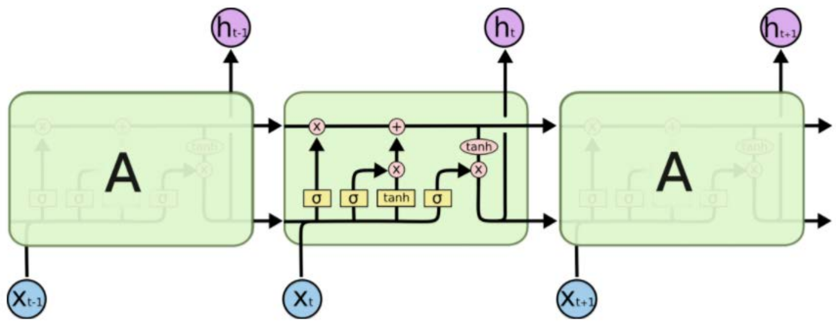
Recurrent neural networks



[Zhu JH et al. 1998]

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

Long-short term memory



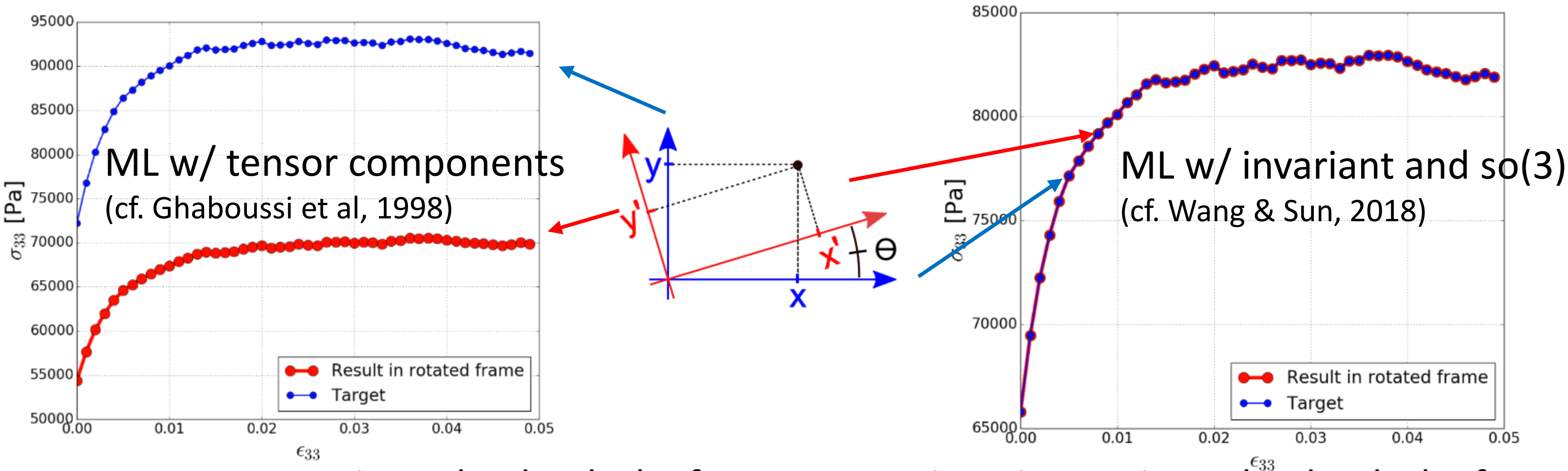
The repeating module in an LSTM contains four interacting layers.

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

This work

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

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

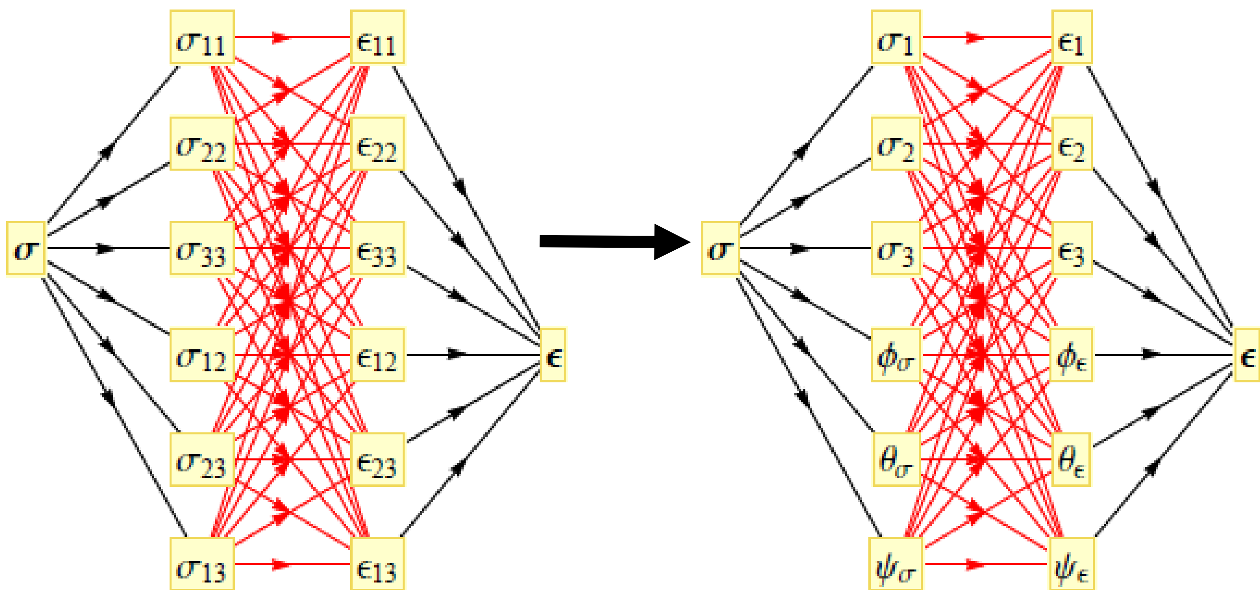


Tensor component as input lead to lack of objectivity (prediction depends on observer)

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

$$\boldsymbol{\sigma}' = \sum_{A=1}^{3} \sigma'_A \boldsymbol{n}_\sigma^{(A)} \otimes \boldsymbol{n}_\sigma^{(A)},$$

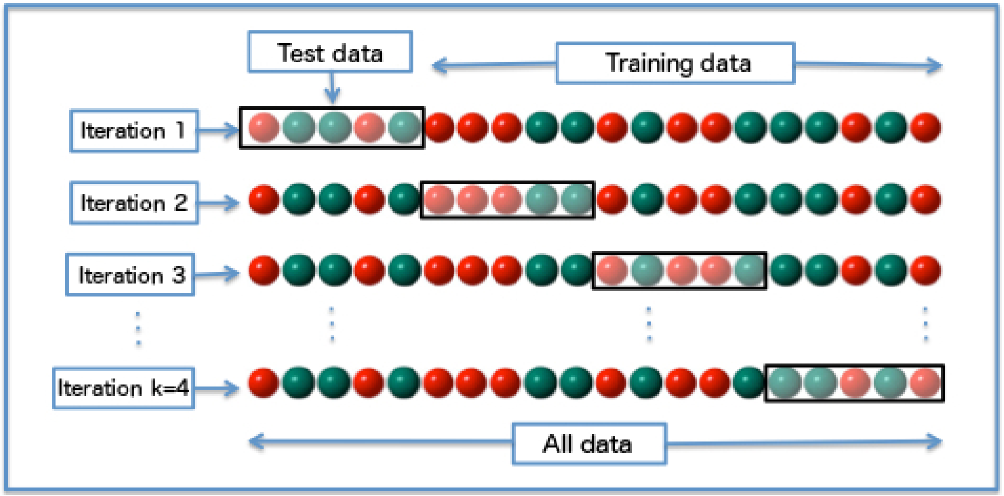
$$\boldsymbol{\epsilon} = \sum_{A=1}^{3} \epsilon_A \boldsymbol{n}_\epsilon^{(A)} \otimes \boldsymbol{n}_\epsilon^{(A)},$$

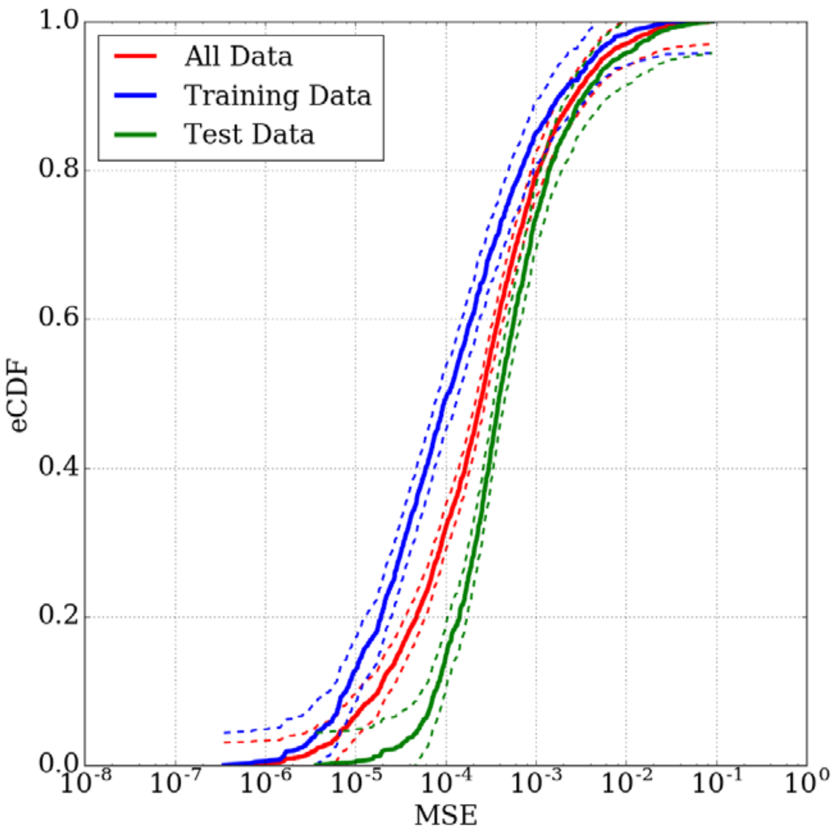


Game Reward: Objective function with calibration/validation

$$\text{SCORE} = \left(\prod_{j=1}^{n_{\text{crit}}} A_j^{\text{crit}}\right) \cdot \left(\sum_{i=1}^{n_{\text{pfm}}} w_i A_i^{\text{pfm}}\right), \tag{4}$$

where $w_i \in [0,1]$ is the weight associated with the measure A_i^{pfm}, and $\sum_{i=1}^{n_{\text{pfm}}} w_i = 1$. In this section, two examples of measures of accuracy A_{accuracy} and prediction consistency $A_{\text{consistency}}$ are presented.

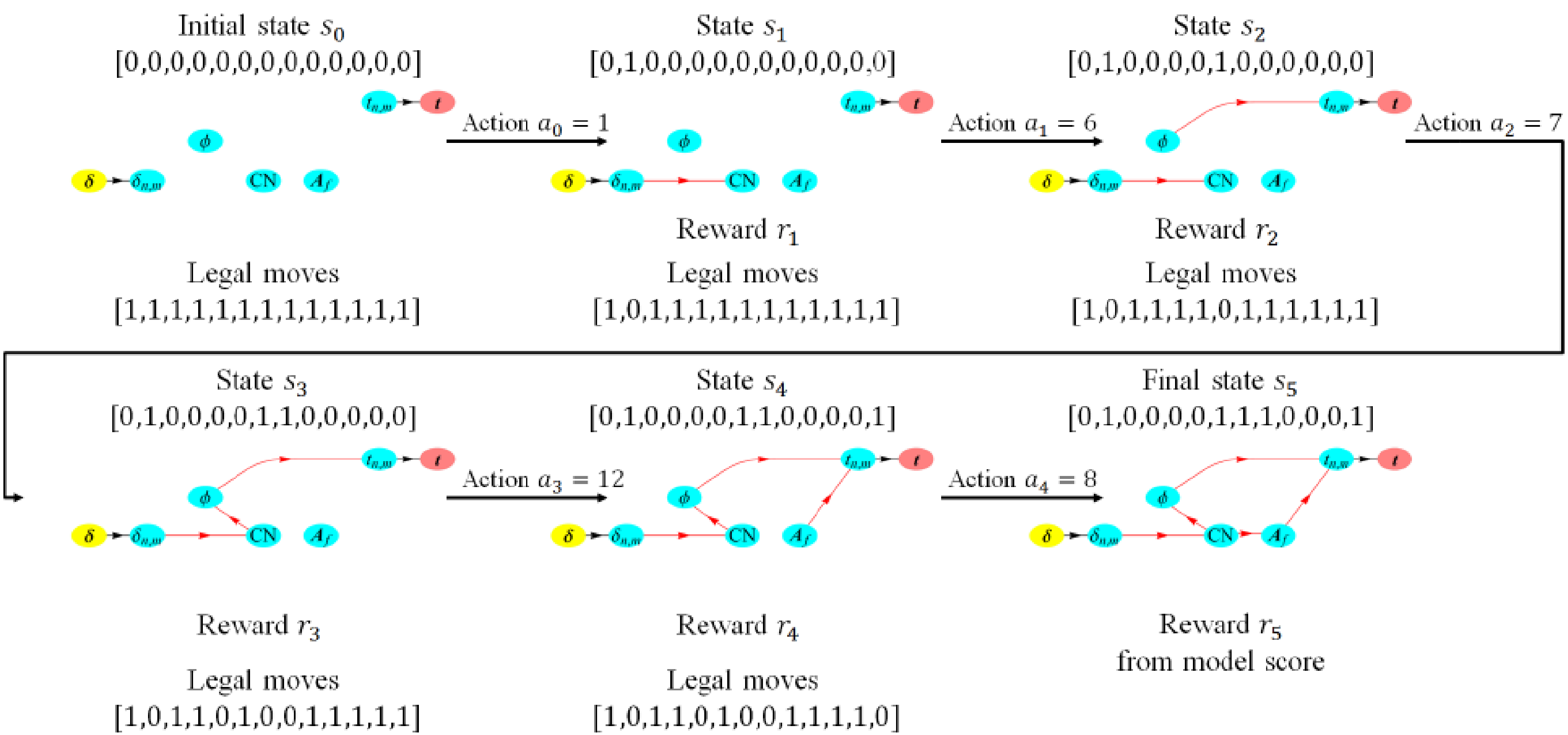




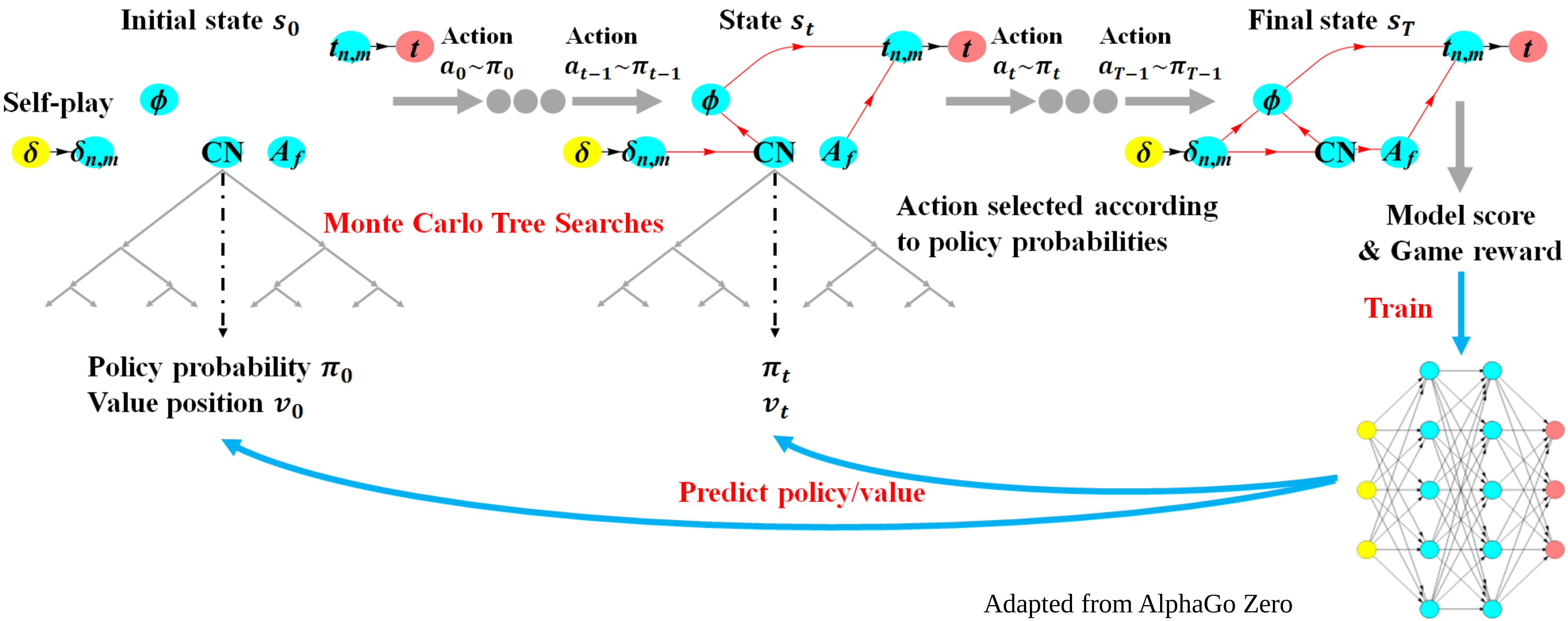
$$A_{\text{accuracy}} = \max\left(\frac{\log[\max(\varepsilon_{p\%}, \varepsilon_{\text{crit}})]}{\log \varepsilon_{\text{crit}}}, 0\right),$$

$$A_{\text{consistency}} = H^{\alpha_{\text{gof}}} = \begin{cases} 0, & p\text{-value} < \alpha_{\text{gof}}, \\ 1, & p\text{-value} \geq \alpha_{\text{gof}}. \end{cases}$$

Modeling procedure of meta-modeling: Markov decision process



A gameplay example formalized as a Markov decision process



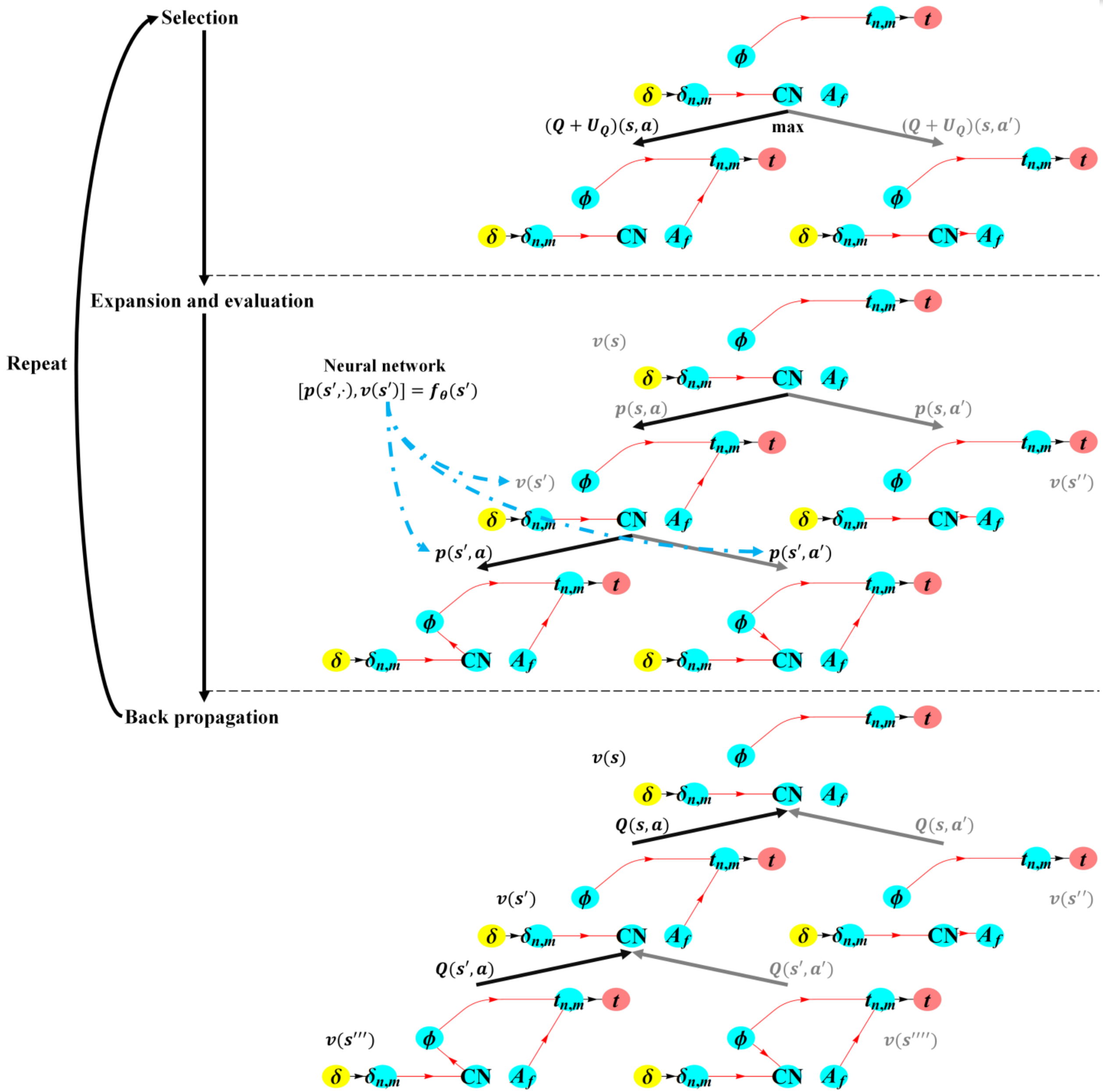
Adapted from AlphaGo Zero
[D Silver et al., 2017]

A (deep) neural network f_θ with parameters θ (weights, bias, ... of the artificial neurons) takes in the current configuration of the directed graph of the constitutive law s and outputs a policy vector p with each component $p_a = p(s,a)$ representing the probability of taking the action a from state s, as well as a scaler v estimating the expected score of the constitutive law game from state s, i.e.,

$$(p, v) = f_\theta(s).$$

These outputs from the policy / value network guide the game play from the AI agent.

Game Playing: Monte Carlo Tree Search



$v(s)$: Value of state *s* of each edge

$p(s, a)$: Probability of taking action *a* at state *s*

$Q(s, a)$: Value of taking action *a* at state *s*

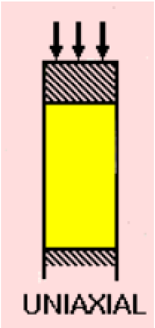
$U_Q(s, a)$: Estmated upper bound of $Q(s, a)$ of taking action *a* at state *s*

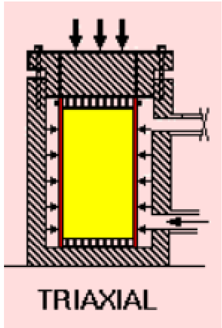
Two agents to play the meta-modeling game collaboratively

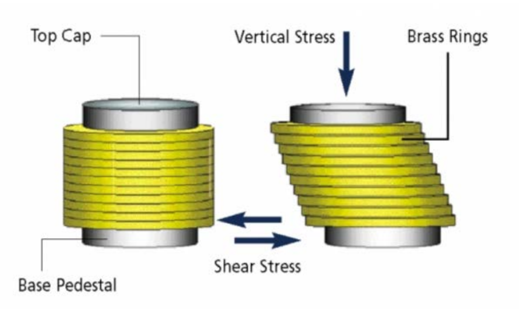
Experimentalist Agent

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)



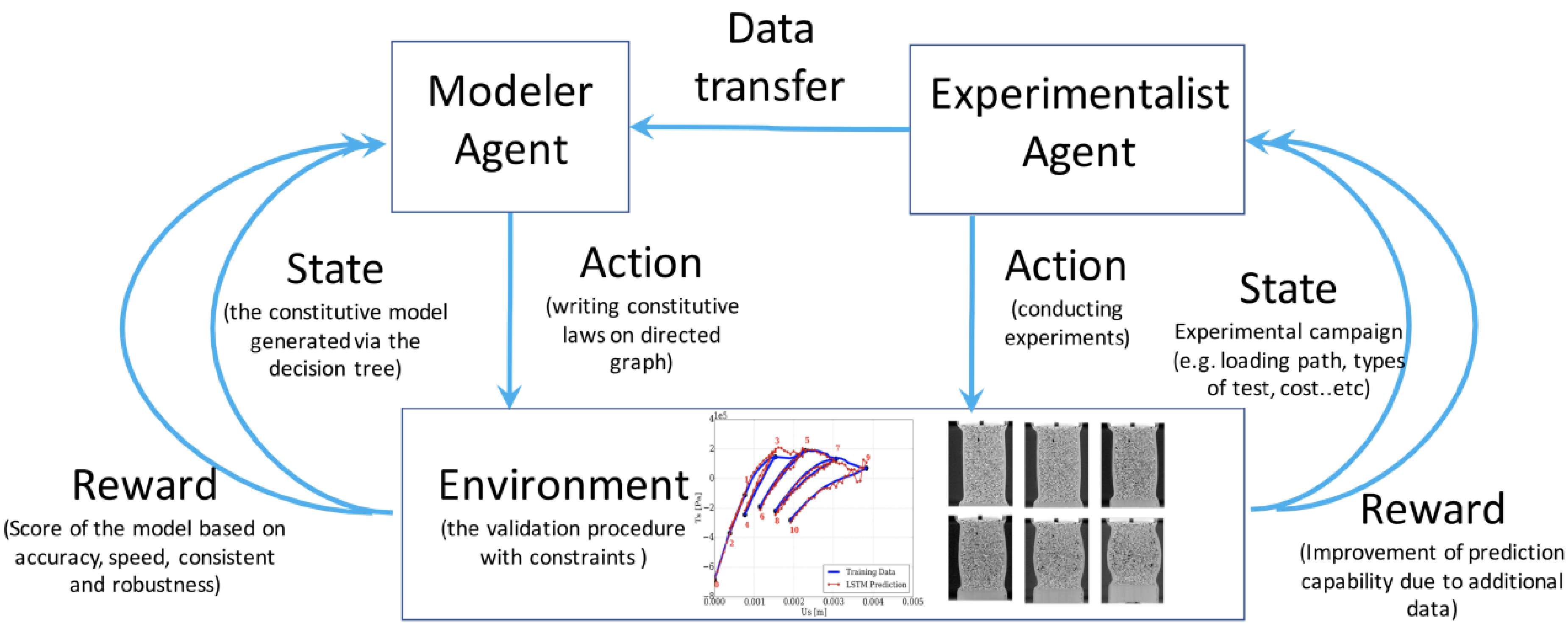




Modeler Agent (identical to the previous single agent)

- 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

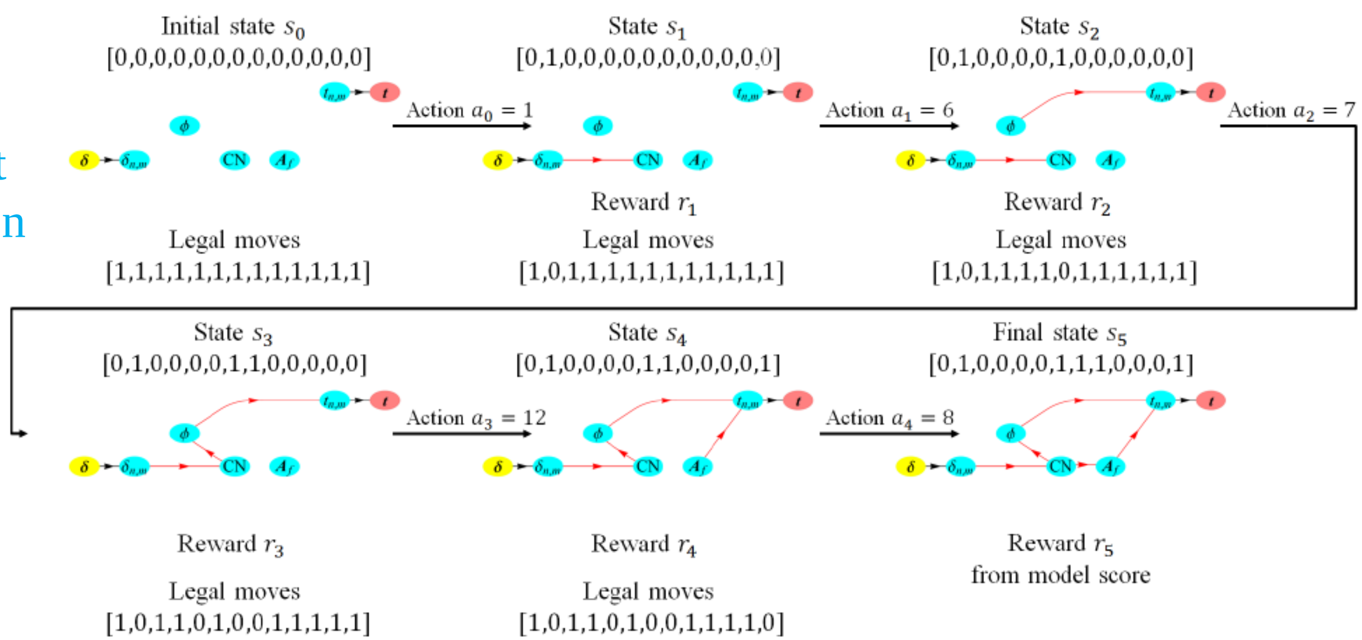
Markov decision process for data collection and meta-modeling

Experimentalist Agent

Initial state s_0	Action $a_0 = T3$	State s_1	Action $a_1 = T5$	State s_2	Action $a_2 = T8$	Final state s_3
Set of experiments [T1]	→	**Set of experiments [T1, T3]**	→	**Set of experiments [T1, T3, T5]**	→	**Set of experiments [T1, T3, T5, T8]**
		Reward r_1		Reward r_2		Reward r_3

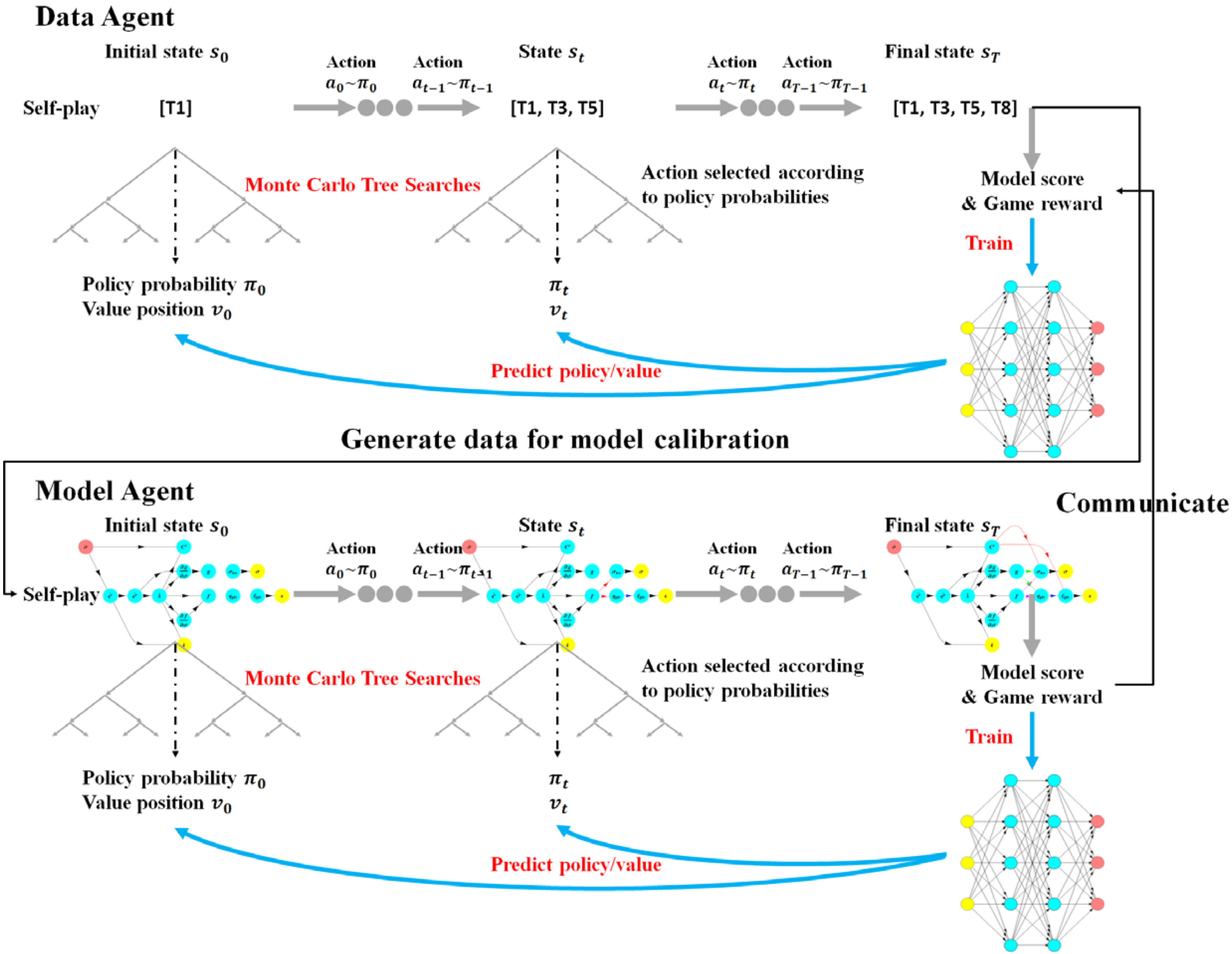
Modeler Agent (identical to the previous single agent)

Construct calibration data



Global Reward

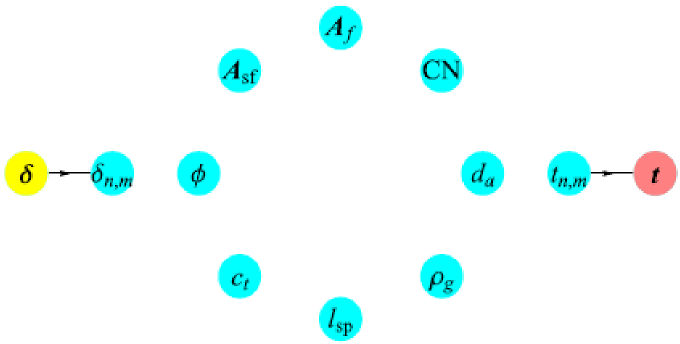
Self-play reinforcement learning of Data Agent and Model Agent



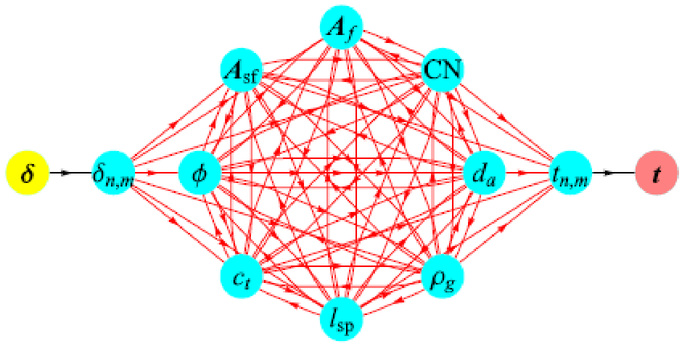
What could be the applications of the meta-modeling approach?

Example 1: Learning traction-separation models from DEM simulations

Game board



(a) Initial configuration of the "game board"



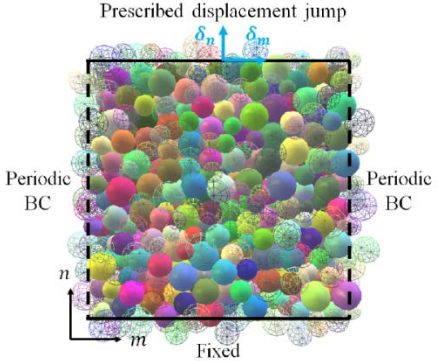
(b) All possible actions on the "game board"

Material: DEM

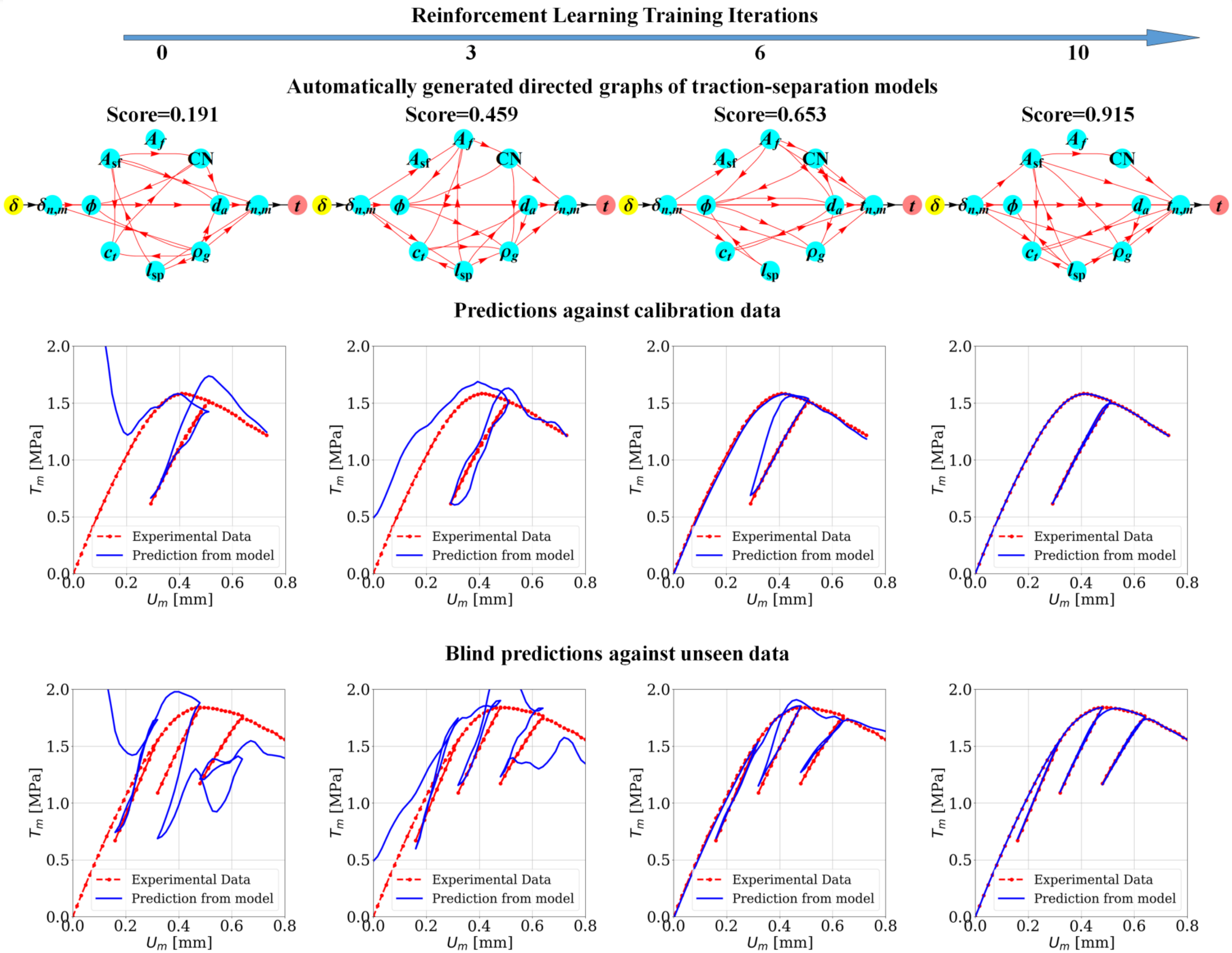
Graph nodes of internal features: porosity, coordination number, fabric tensor, strong fabric tensor, degree assortativity, transitivity coefficient, average shortest path length, density of the graph

Graph edge choices: LSTM neural networks

Parameters of DRL: number of Iterations = 10
number of episodes = 30
number MCTS simulations = 30

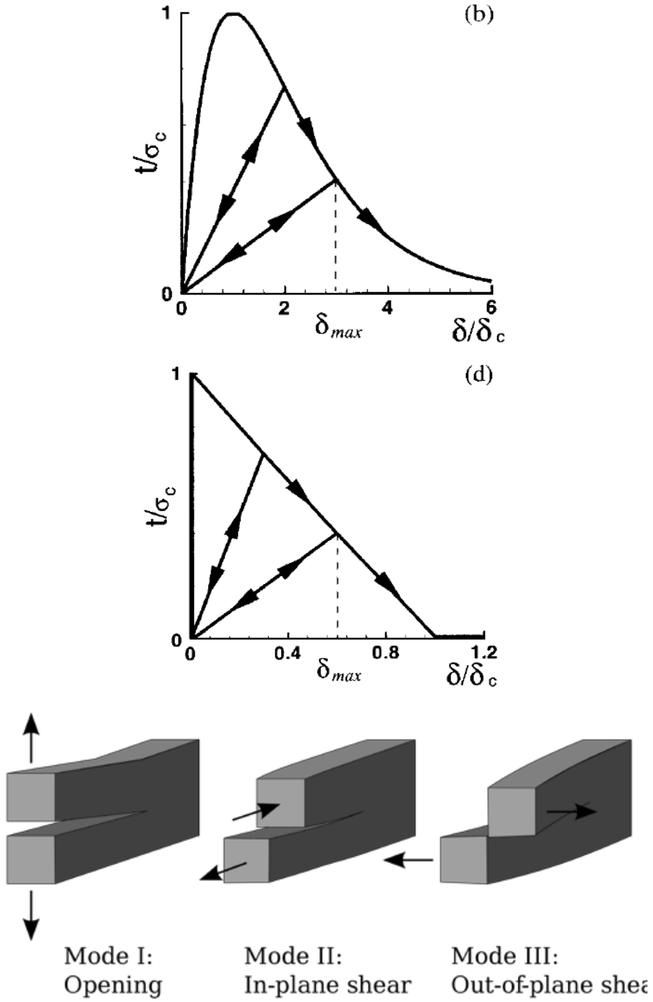


Example 1: Learning traction-separation models from DEM simulations

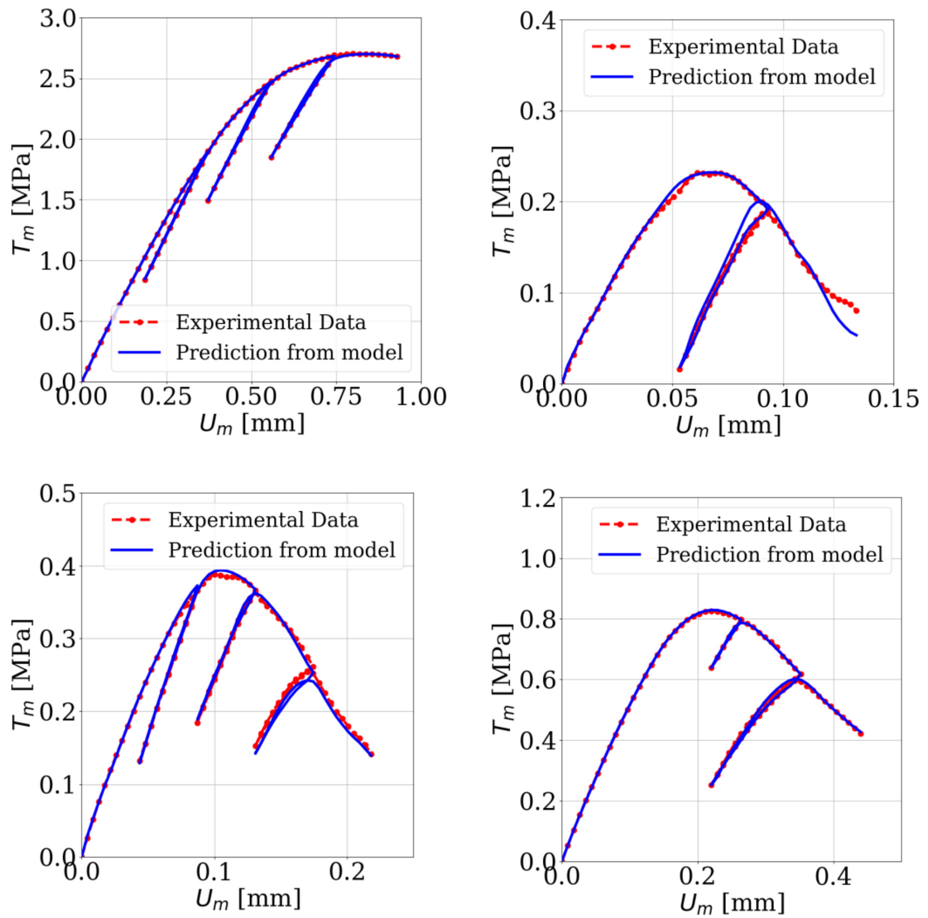


Example 1: Learning traction-separation models from DEM simulations

Hand-crafted traction-separation models reviewed in [M Ortiz, A Pandolfi, 1999]

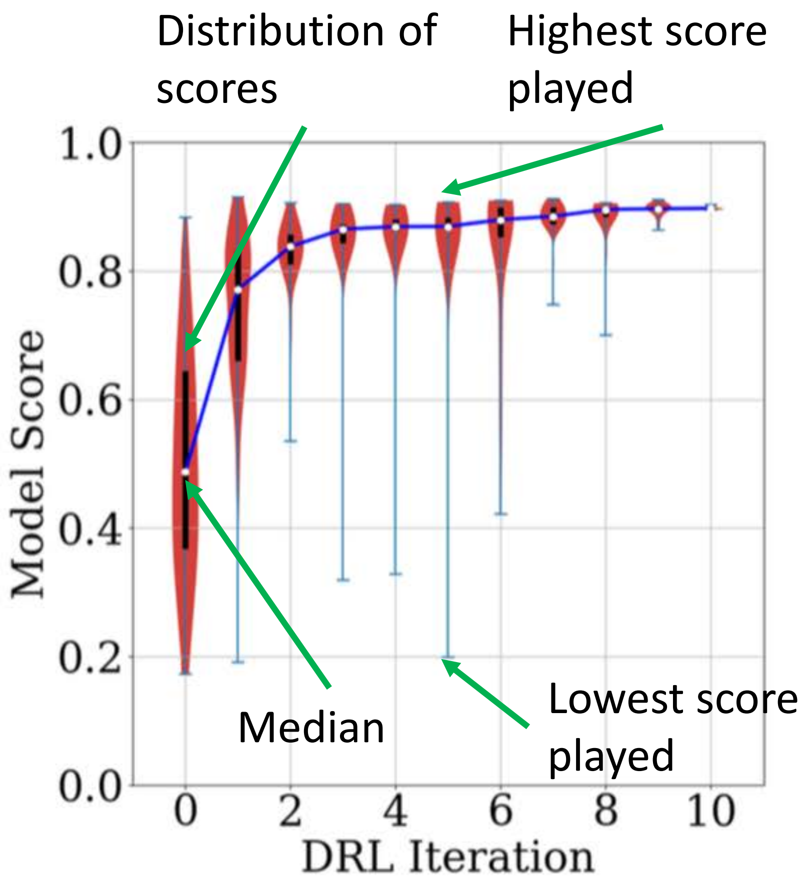


Self-reinforcement-learned traction-separation model validated against cyclic data

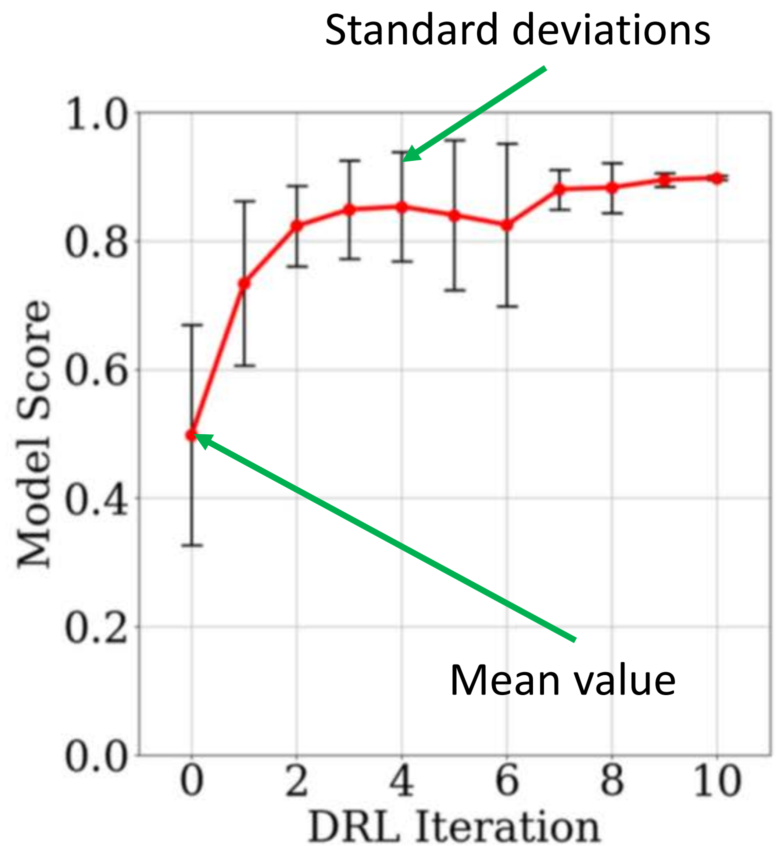


Example 1: Learning traction-separation models from DEM simulations

Statistical performance of meta-modeling over self-learning sessions



Violin plots of the density distribution of model scores

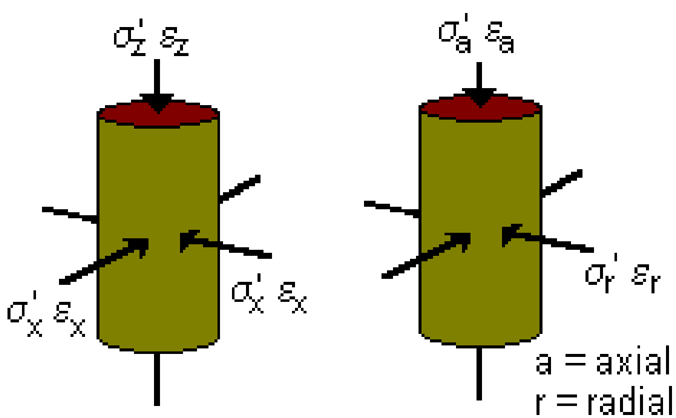


Mean value and ± standard deviation of model score in each DRL iteration in each DRL iteration

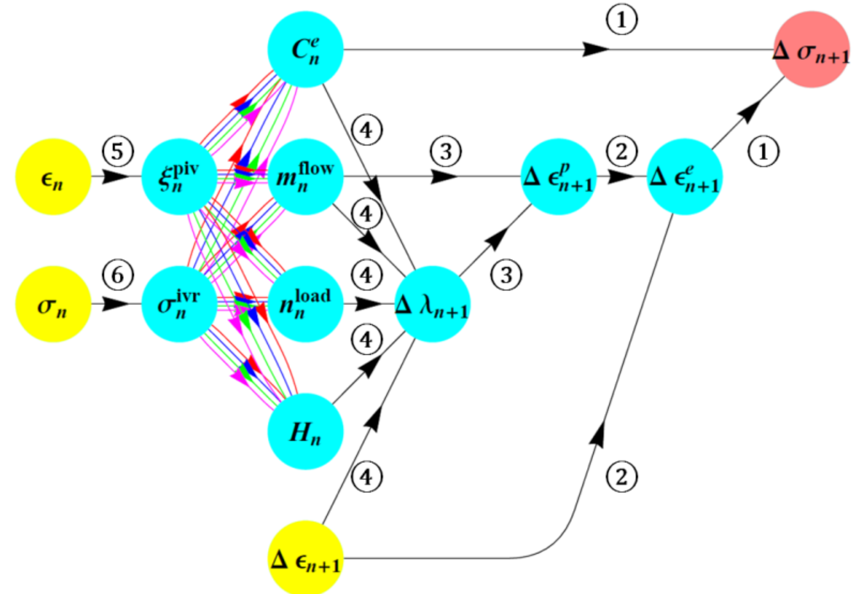
Example 2: Learning elasto-plasticity constitutive models

Two-agent game: data collections and meta-modeling

Game choices for experimentalist agent



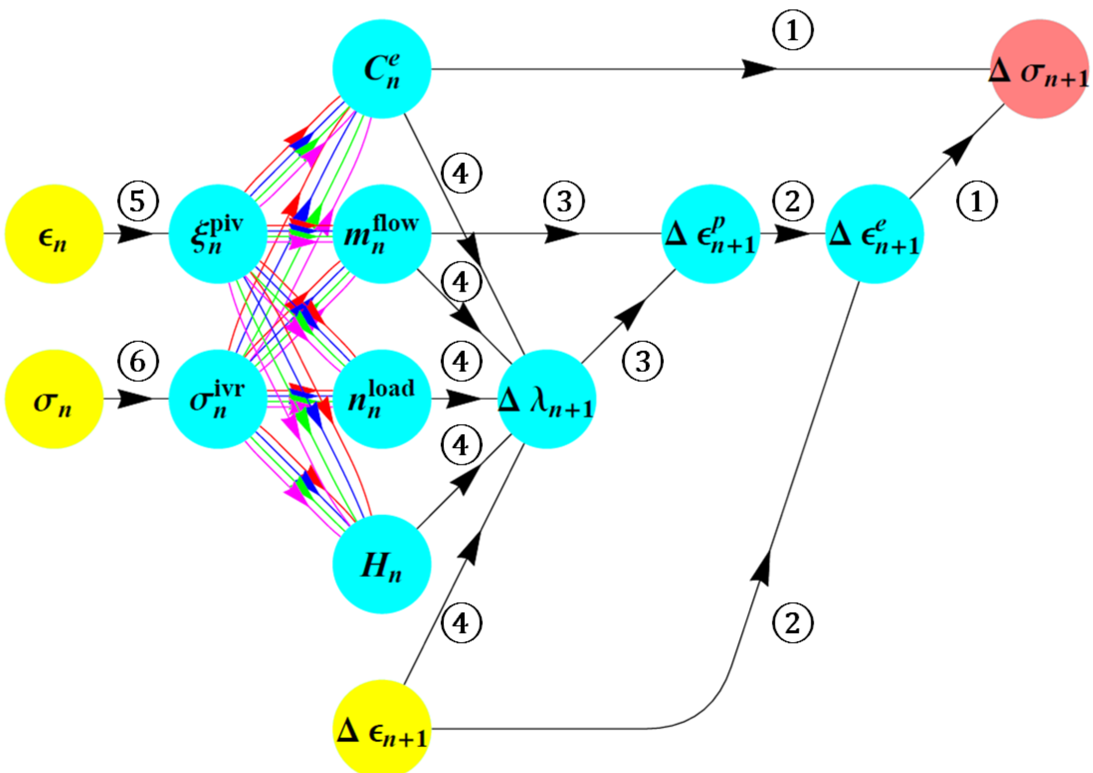
Game choices for modeler agent



Parameters of DRL: number of Iterations = 10, number of episodes = 30, number MCTS simulations = 300

Example 2: Learning elasto-plasticity constitutive models

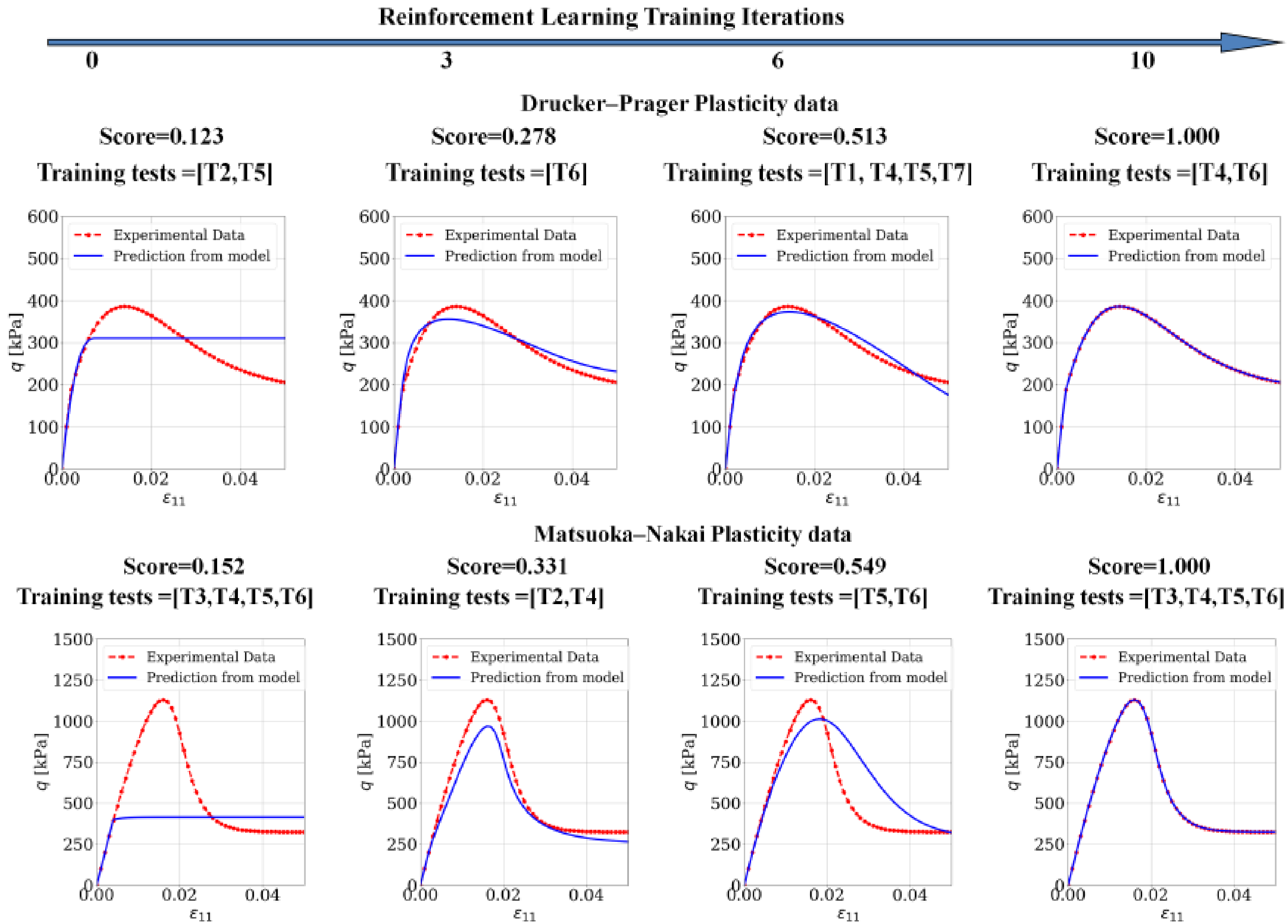
Game board (Directed multi-graph of generalized elasto-plasticity)



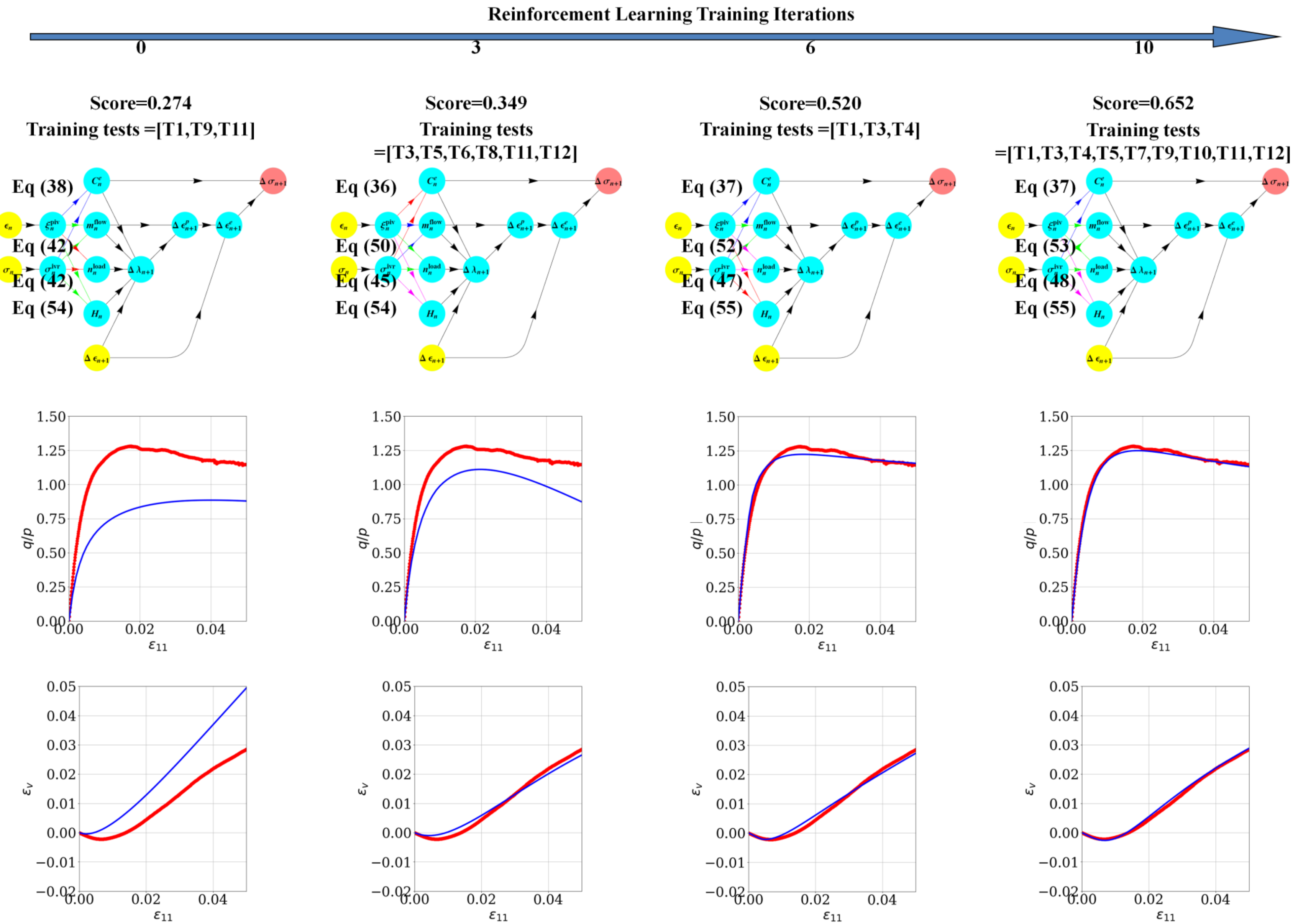
- Graph nodes: elastic stiffness, loading direction, plastic flow direction, plastic modulus
- Graph edges for definitions in generalized plasticity.
- 4 million possible number of configurations for both experimentalist and modeler combined – impossible to hand crafting all possible choices.
- DRL with MCTS is used to solve the combinatorial optimization problem
- With 2% of total evaluation, we obtain the estimated optimal choice.

Example 2: Reverse engineering from data of black-box models

Validation (reverse engineering): given data generated by a specific elasto-plasticity model, check whether the meta-modeling can identify the correct constitutive model.

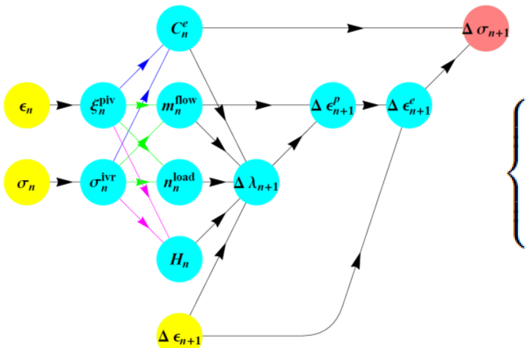


Example 2: Learning elasto-plasticity constitutive models



Example 2: Learning elasto-plasticity constitutive models

Score=0.652



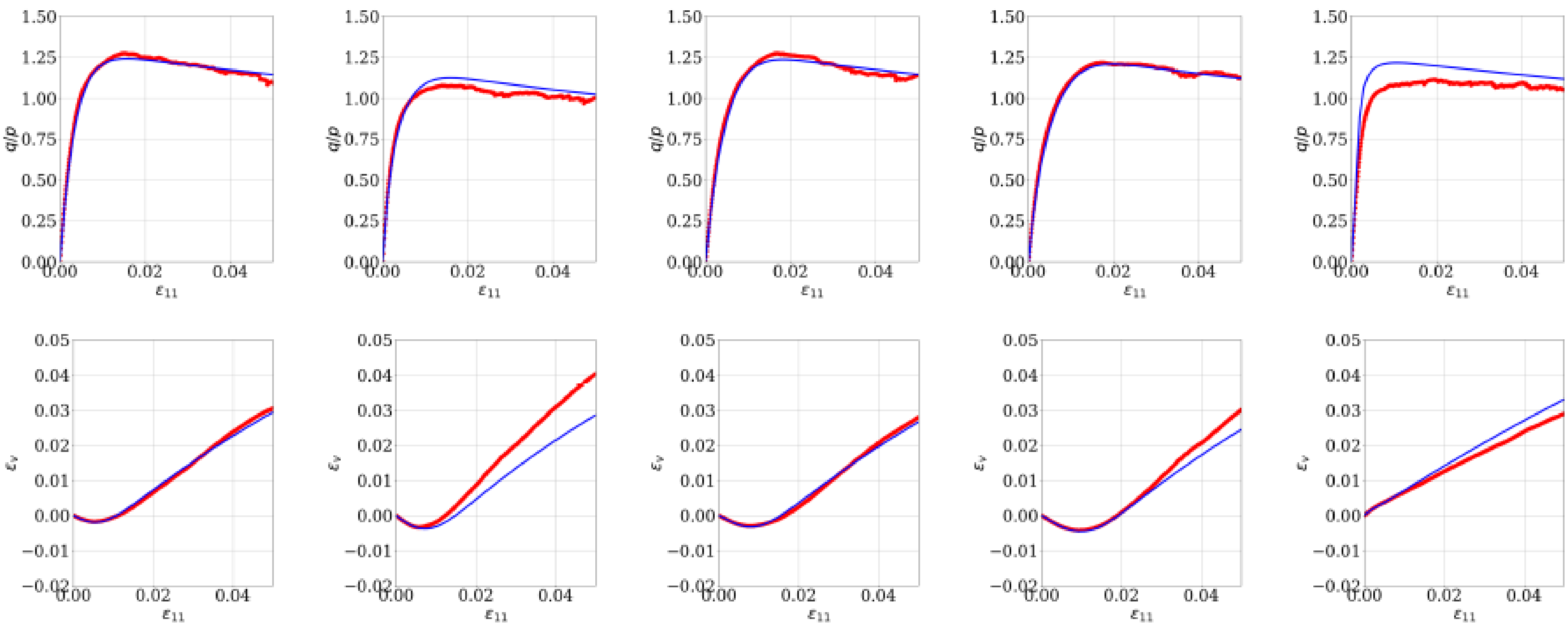
Edges of the optimal digraph

$$\begin{cases} K = K_0(\frac{p}{p_{at}})^a \\ G = G_0(\frac{p}{p_{at}})^a \end{cases}$$

$$\begin{cases} n_v^{load} = \frac{d_f}{\sqrt{1+d_f^2}} \\ n_s^{load} = \frac{1}{\sqrt{1+d_f^2}} \\ d_f = (1+\alpha)(M_f \exp(m_f(1-e)) + q/p) \end{cases}$$

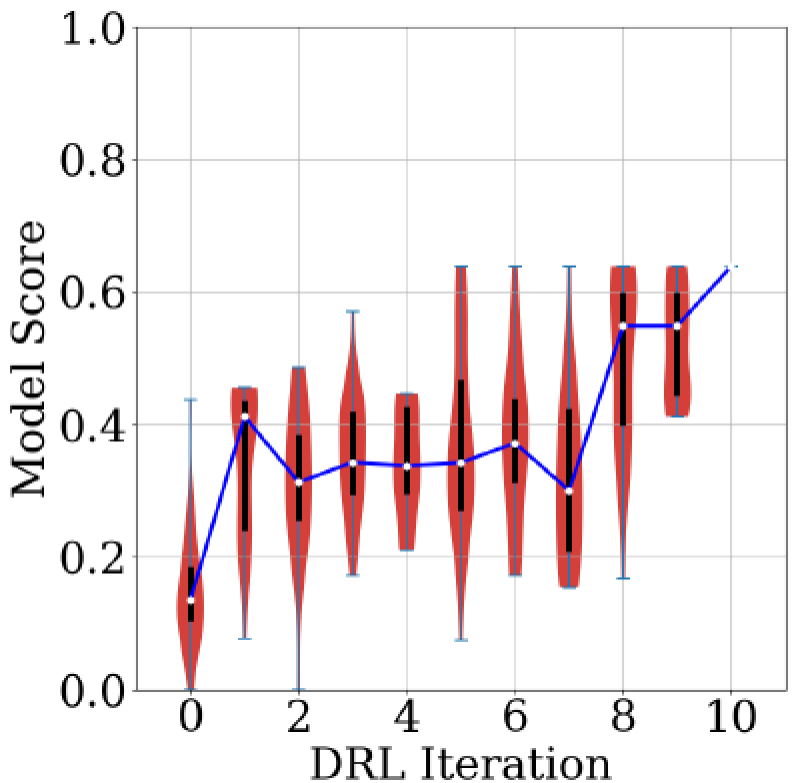
$$\begin{cases} m_v^{flow} = \frac{d_g}{\sqrt{1+d_g^2}} \\ m_s^{flow} = \frac{1}{\sqrt{1+d_g^2}} \\ d_g = (1+\alpha)(M_g \exp m_g \psi + q/p) \\ \psi = e - e_{c0} + \tilde{\lambda}(p/p_{at})^a \end{cases}$$

Examples of blind predictions from the optimal digraph configuration against data from the tests

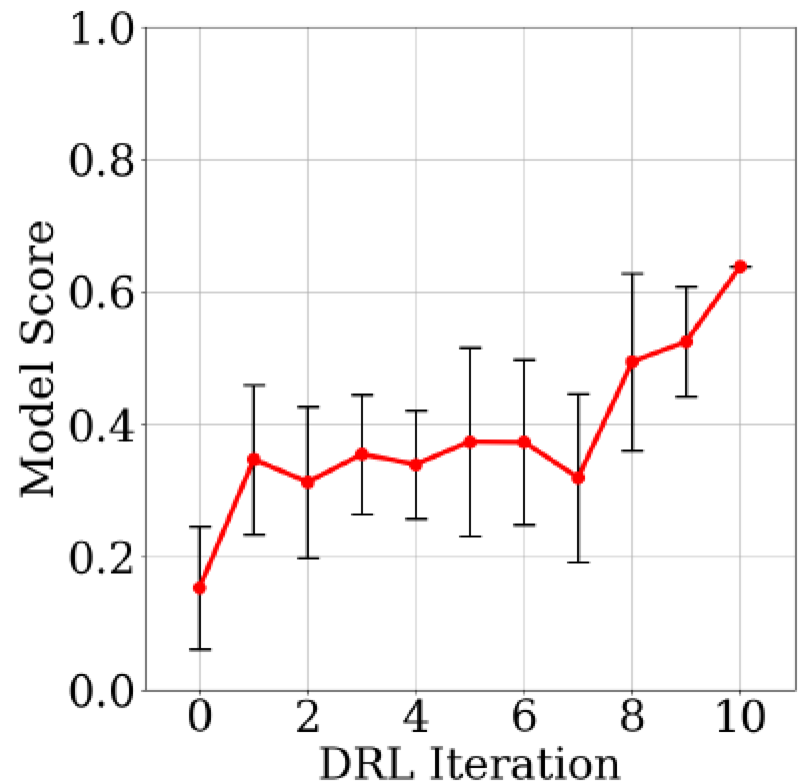


Example 2: Learning elasto-plasticity constitutive models

Statistical performance of meta-modeling over self-learning sessions



Violin plots of the density distribution of model scores



Mean value and ± standard deviation of model score in each DRL iteration in each DRL iteration

Example 2: Post-game analysis: performance in blind predictions

Five classes of the constitutive models generated during the deep reinforcement learning

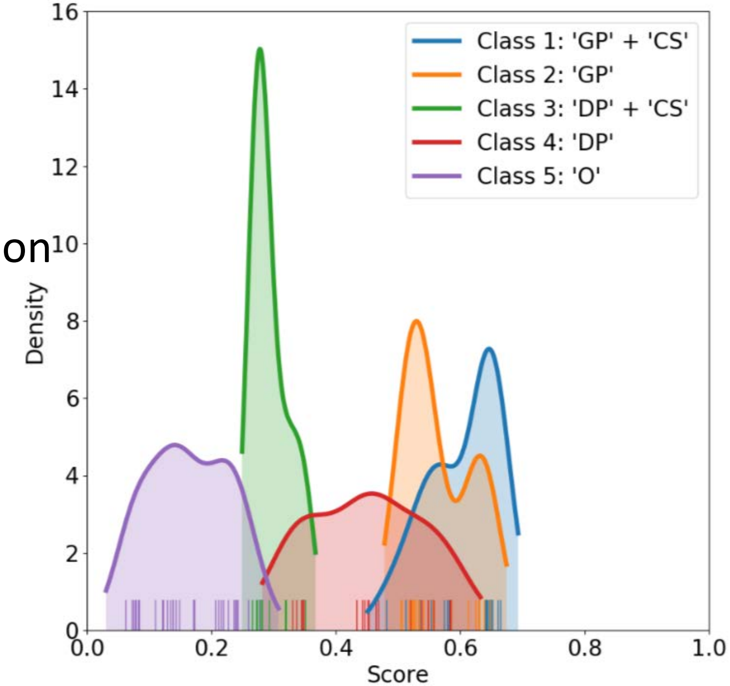
Model Class	Number of Models	Mean Score	Standard deviation	Generalized Plasticity 'GP'	Critical State 'CS'	Classical pressure dependent elasto-plasticity 'DP'	Others 'O'
1	22	0.603	0.054	✓	✓		
2	25	0.565	0.051	✓			
3	13	0.295	0.028		✓	✓	
4	19	0.450	0.086			✓	
5	33	0.163	0.063				✓

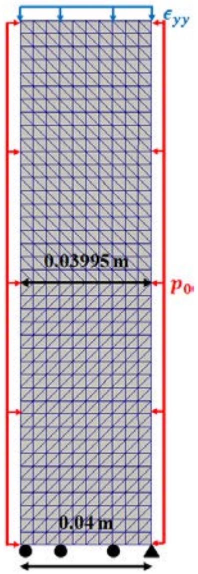
Distribution of the scores of the models generated during the deep reinforcement learning.

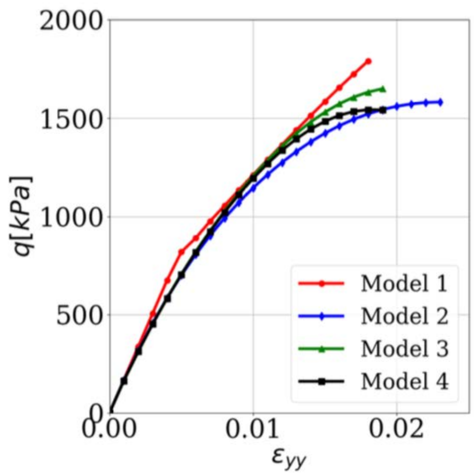
The models are grouped into five families (see Table).

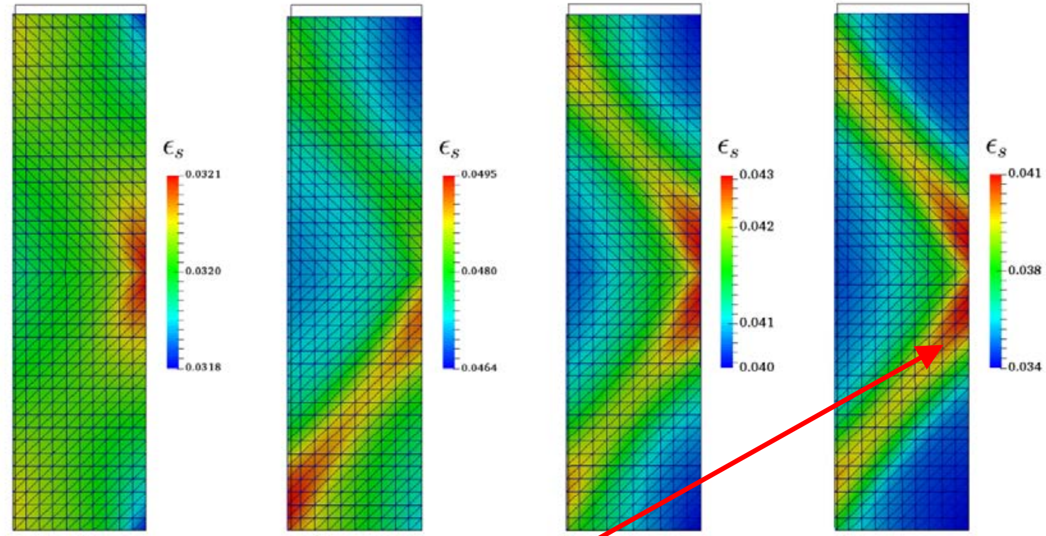
The curves present the Gaussian kernel density estimation of the model score distributions

The analysis confirms that **generalized plasticity** (no yield surface and plastic potential) and **critical state** (dependence on pressure and porosity) are important ingredients in an accurate elasto-plasticity model for granular materials

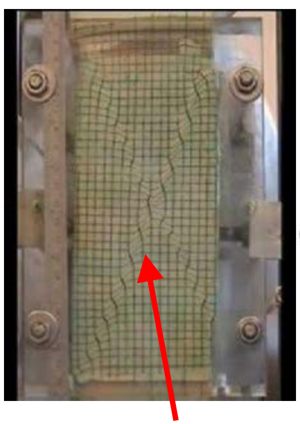




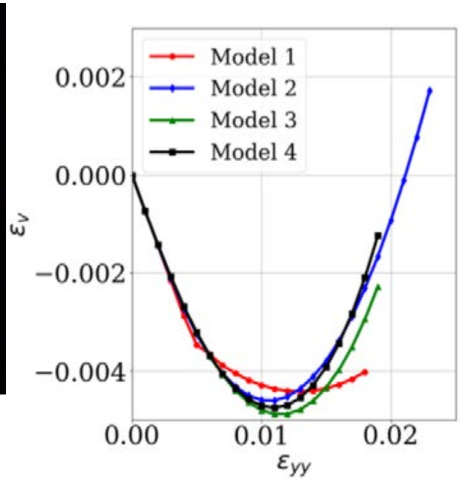


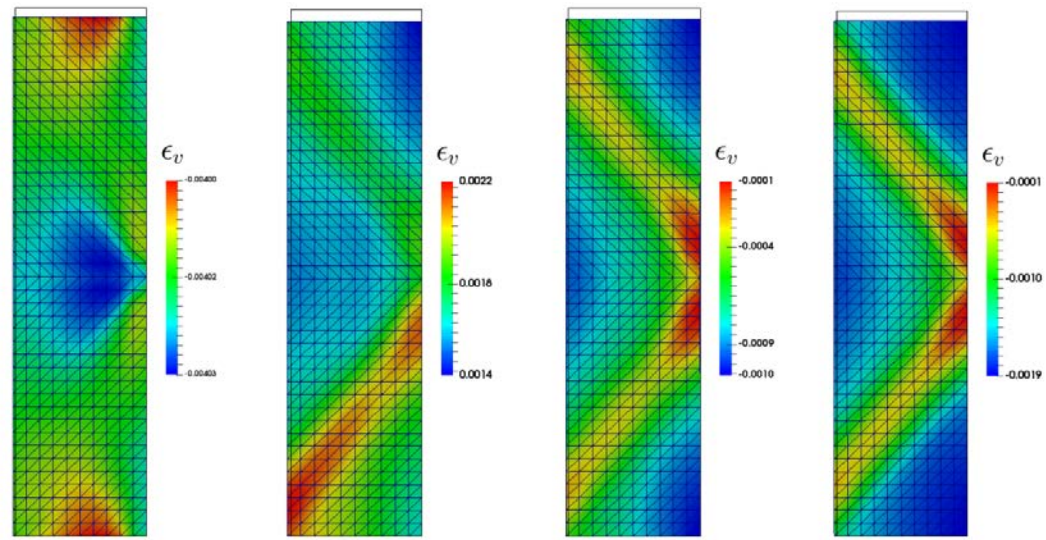


Shear band replicated in ML-FEM simulation



Shear band observed in experiment





1st iteration | 5th iteration | 8th iteration | 10th iteration

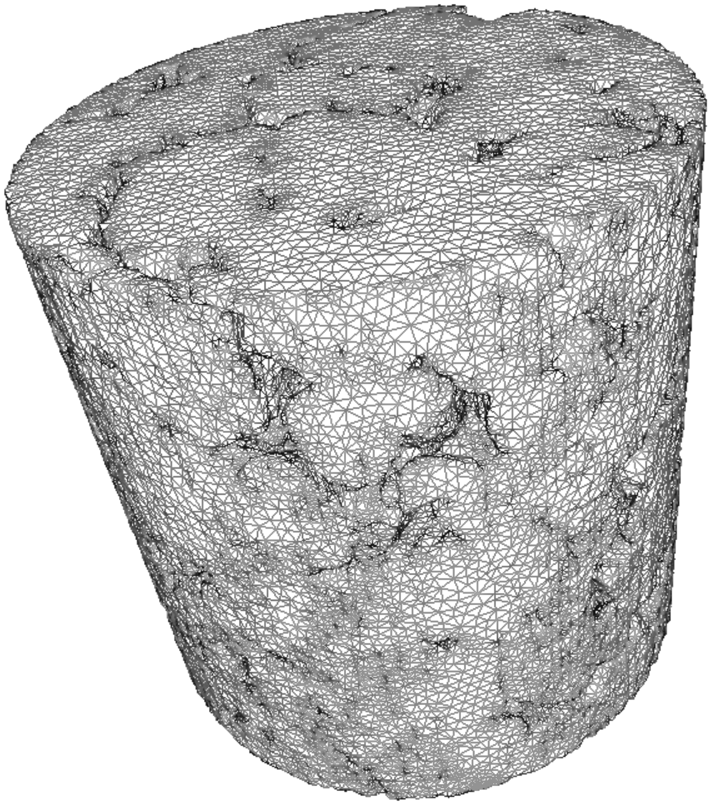
Conclusion

- This metal-modeling approach is the key for us to exploit the computer power to make repeated trial-and-errors and improve from experiments over time to generate the best model outcomes, instead of spending significant human time to explore through curve-fitting physical processes. Human labor can focus on **expanding action spaces** (nodes and edges choices), **designing rules and objective functions**.

- Since the machine learning procedure is automated, models intended for fulfilling different demands (speed, accuracy, robustness) can be **automatically generated and improved** over time through self-play in the model-creation game.

- Since the validation procedure is introduced as the reward mechanism for the agent to find the best models available, the resultant models are **always validated** at the end of the game.

- The metal-modeling approach is **generic, reusable and easily expandable**, which means that it can handle different situations with different data, action spaces, objectives and rules set by human without going through additional derivation, implementation, material parameter identification and validation.



THANK YOU!

More information can be found at
http://www.poromehanics.org

- K. Wang, W.C. Sun, Q. Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation, Comput. Mech., 2019
- K. Wang, W.C. Sun, Meta-modeling game for deriving theory-consistent, micro-structure-based traction-separation laws via deep reinforcement learning, CMAME, 2019.
- K. Wang, W.C. Sun, A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning, CMAME, 2018.