University of Liège
Aerospace & Mechanical Engineering

AI and data-driven methods for composite multi-scale analyses

Ludovic Noels
Collaborators: L. Wu, C. Anglade, V.-D. Nguyen, …
Computational & Multiscale Mechanics of Materials – CM3
http://www.ltas-cm3.ulg.ac.be/
Allée de la découverte 9, B4000 Liège
L.Noels@ulg.ac.be

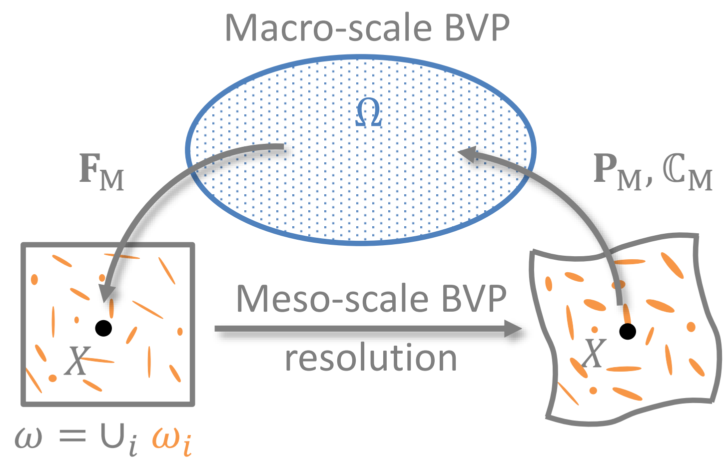
Multi-scale formalism

- 2-Scale problem
  - Idea:
    - Substitute the explicit constitutive model

      $\boldsymbol{\sigma}_{\mathrm{M}}(\boldsymbol{X}, t) = \boldsymbol{\sigma}_{\mathrm{M}}\big(\boldsymbol{\varepsilon}_{\mathrm{M}}(\boldsymbol{X}, t); \boldsymbol{z}_{\mathrm{M}}(\boldsymbol{X}, \tau), \tau \in [0, t]\big)$
    - By an implicit relation resulting from the lower-scale BVP resolution
    - Macro-scale state variables correspond to the set of RVE state variables

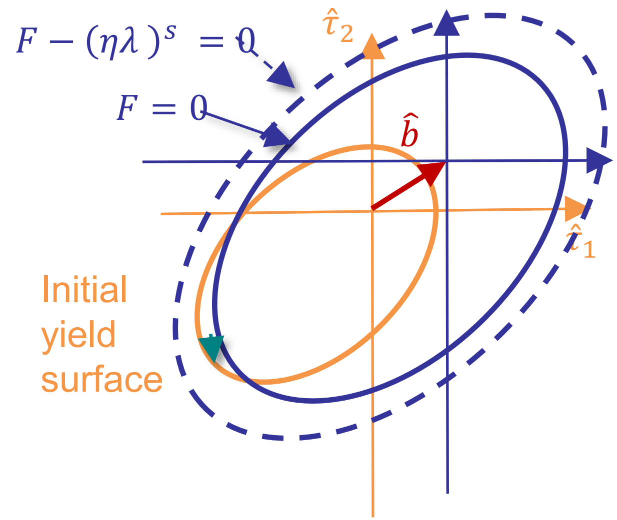
      $\boldsymbol{z}_{\mathrm{M}}(\boldsymbol{X}, t) = \{\boldsymbol{z}_{\mathrm{m}}(\mathbf{x}, t)\ \forall \boldsymbol{x} \in \boldsymbol{\omega}(\boldsymbol{X})\}$
  - Basic assumption: separation of scale

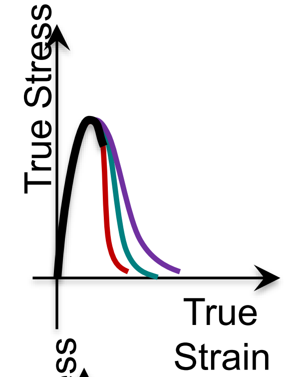
    $l_{\mathrm{M}} >> l_{\mathrm{RVE}} >> l_{\mathrm{m}}$
  - Opportunities
    - Complex simulations (accounting for complex failure modes
    - Stochastic analyses
    - …

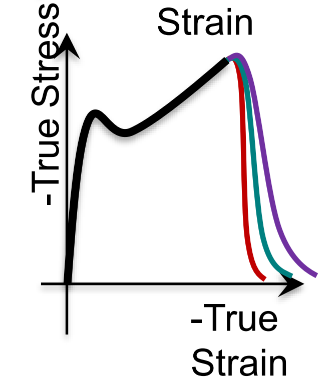


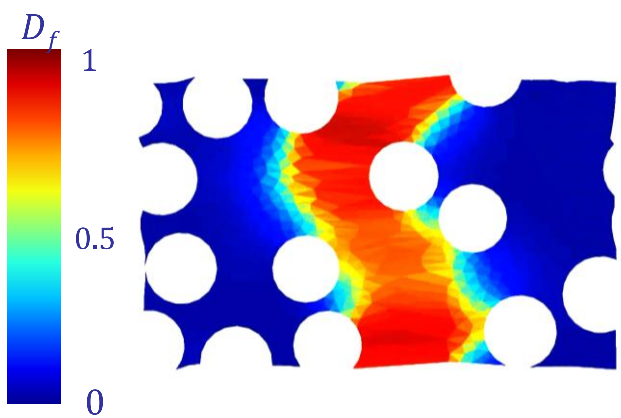
Multi-scale formalism

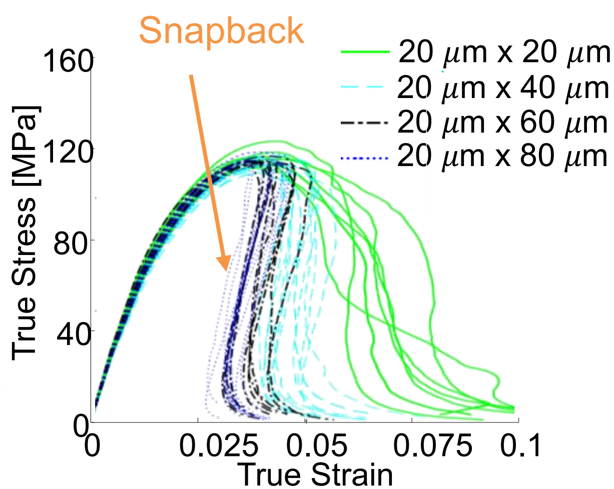
- RVE simulations:
  - Complex material model for resin
    - (Visco-Elasticity-)Visco-plasticity with pressure dependency
    - Triaxiality-dependent damage models
    - Non-local damage models
  - Transverse anisotropy for fibre
    - Failure with phase-field
  - Experimentally validated

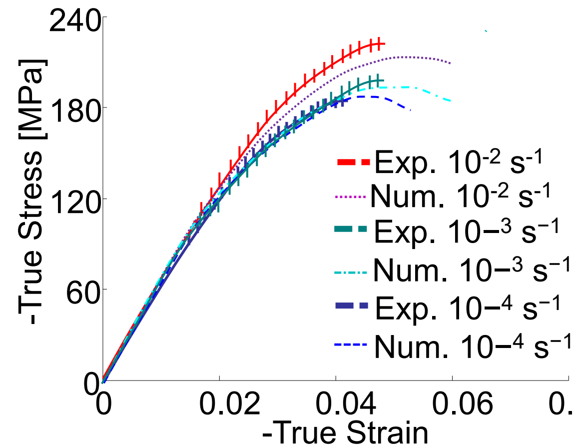








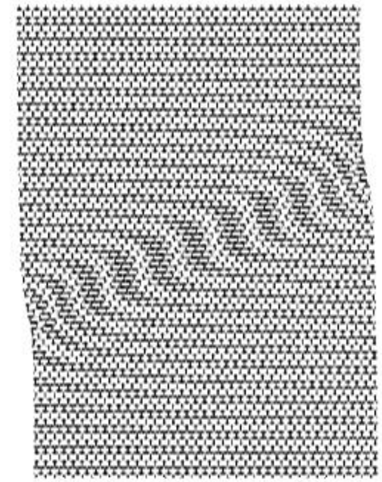


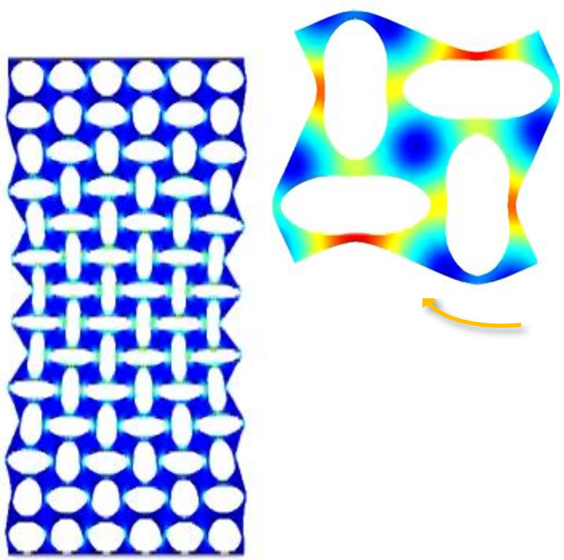


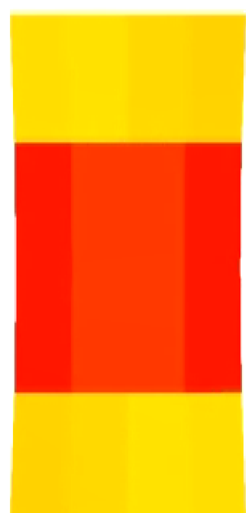
Multi-scale formalism

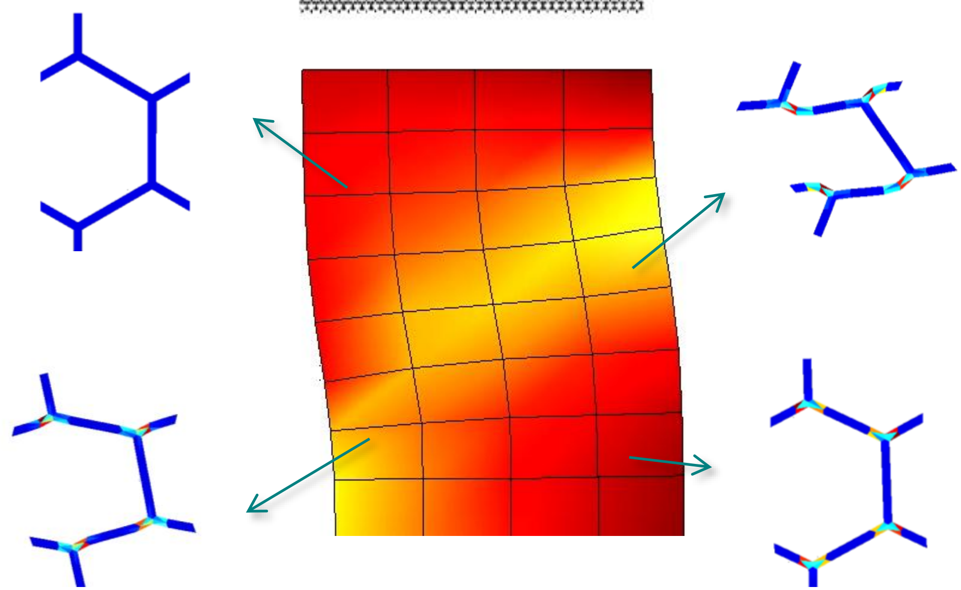
- Multi-Scale FE^2 resolution
  - Method has been applied to a wide variety of problems
  - Several drawbacks
    - Overwhelming computation time
    - Fixed micro-structure

Not manageable for real applications



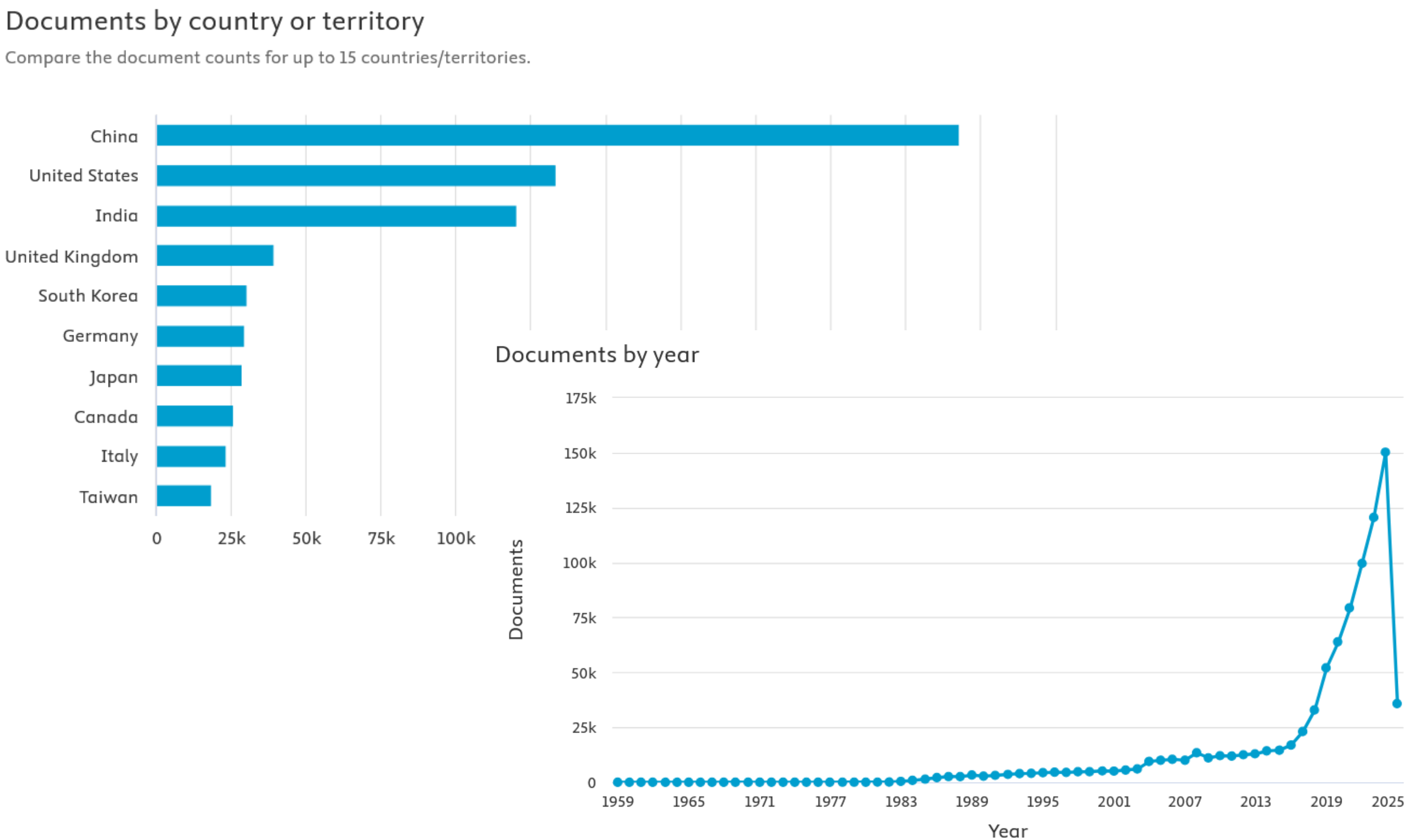






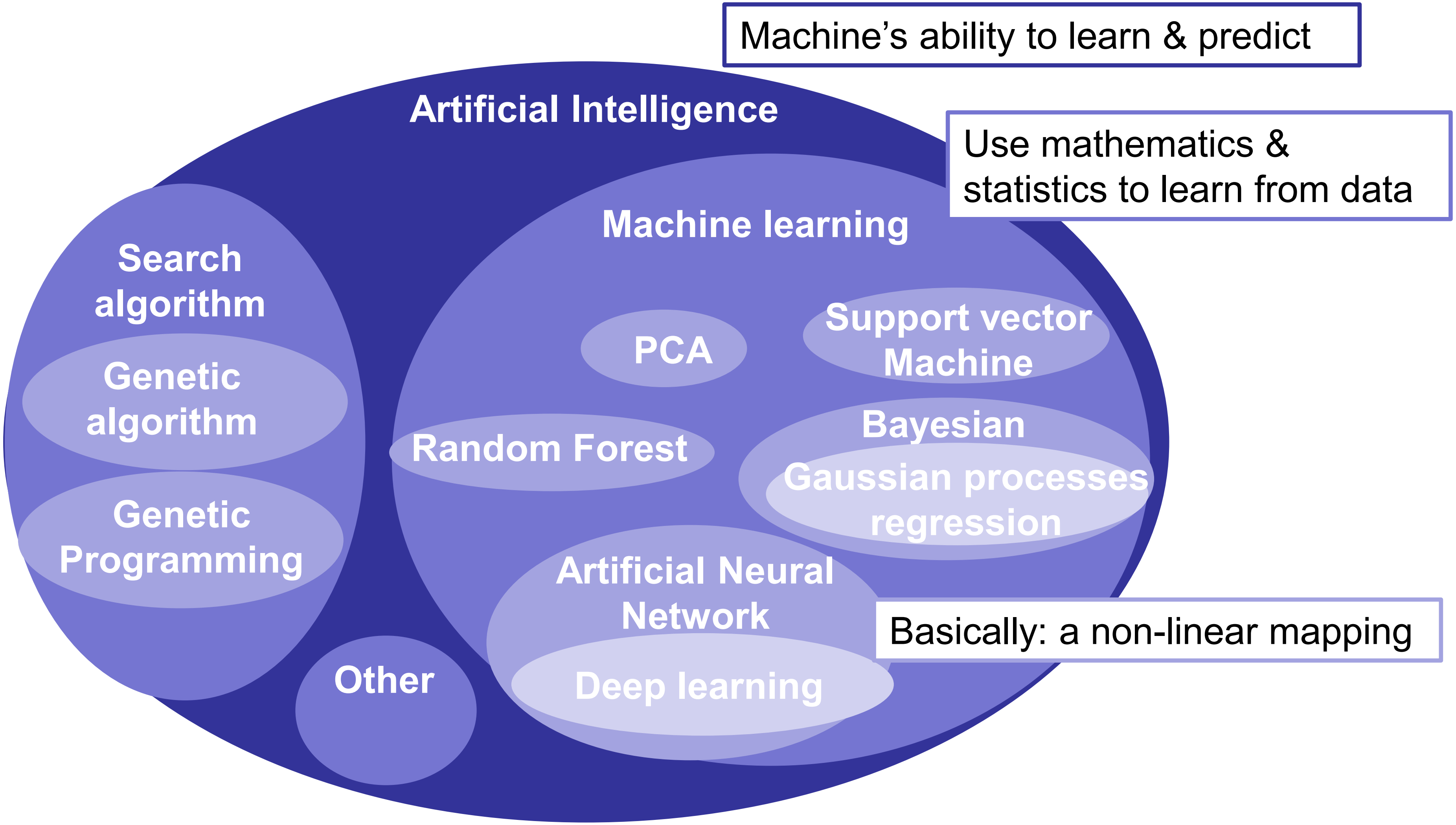
Can AI help?

- Artificial intelligence / machine learning / neural network in engineering



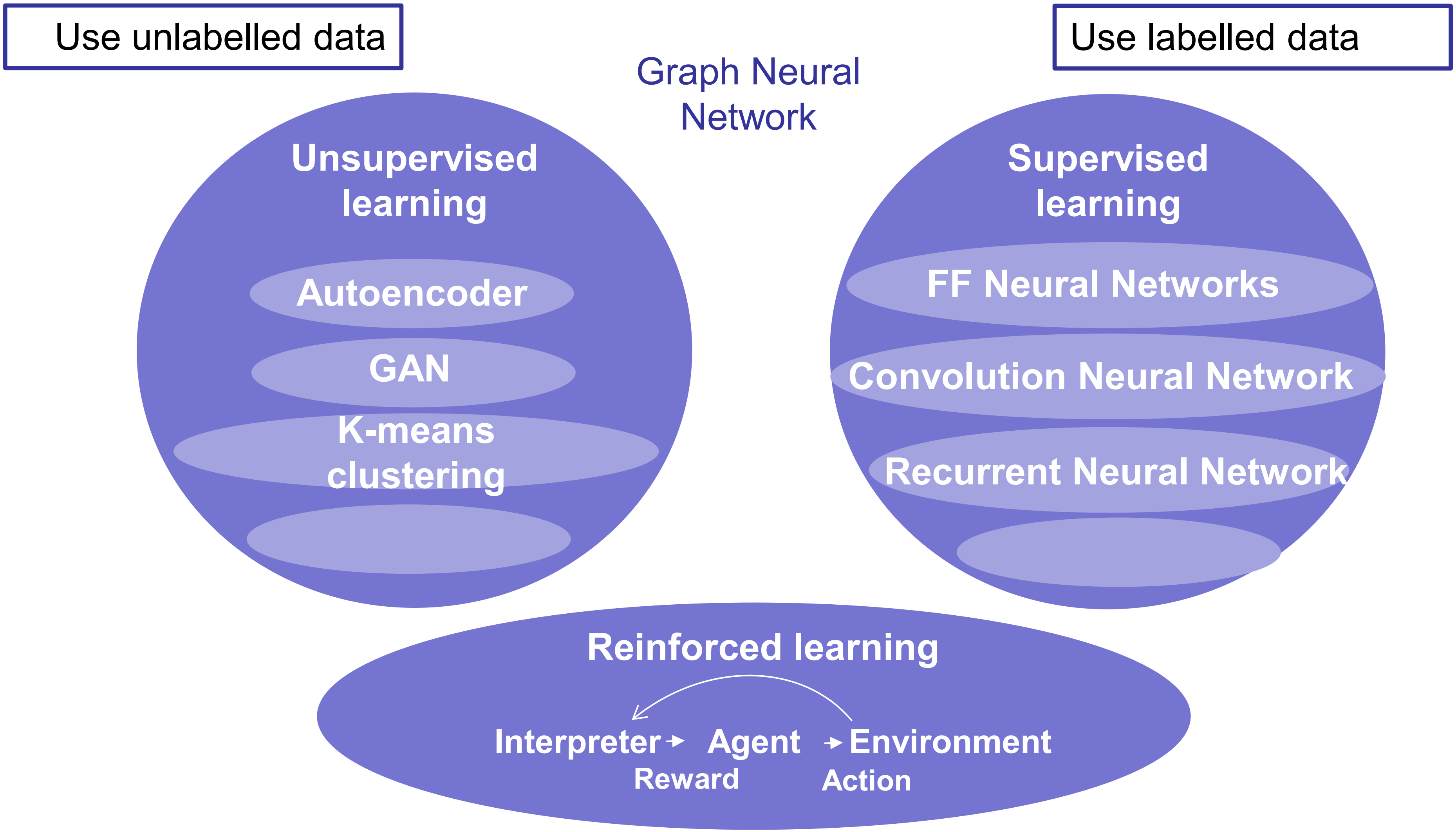
Can AI help?

- What is Artificial Intelligence?

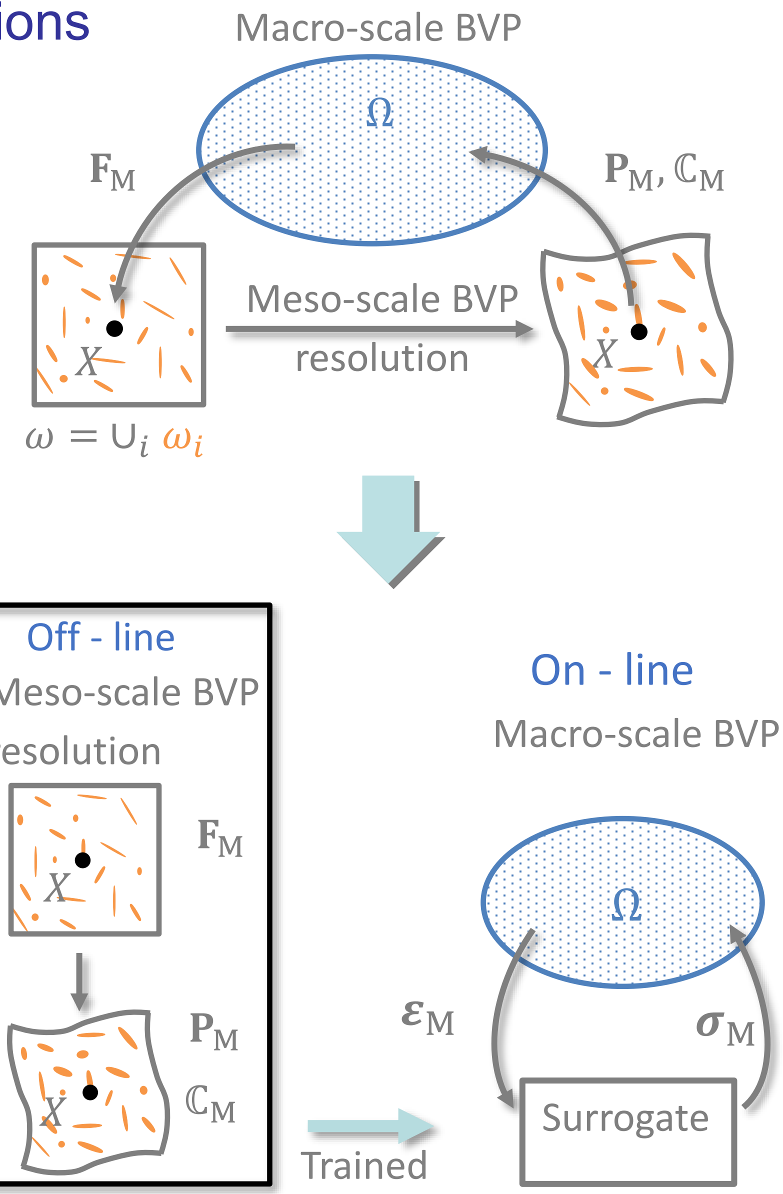


Can AI help?

- What is Artificial Intelligence?

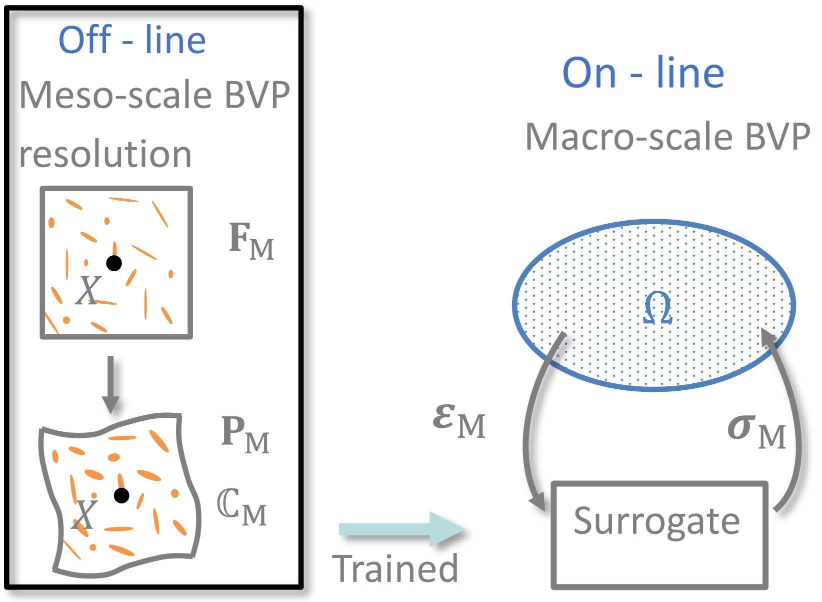


- Surrogated non-linear multi-scale simulations
  - FE multi-scale simulations
    - Problems to be solved at two scales
    - Require Newton-Raphson iterations at both scales
  - Use of surrogate models
    - Train a meso-scale surrogate model
      - Off-line
      - Requires data from RVE simulations
      - Supervised learning
    - Use the trained surrogate
      - On-line analyses
      - Surrogate acts as a homogenised constitutive law
      - Speed-up of 4-5 orders



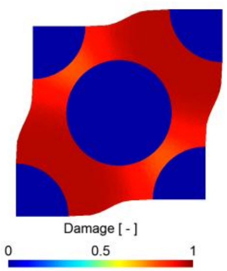
AI-accelerated multi-scale formalism

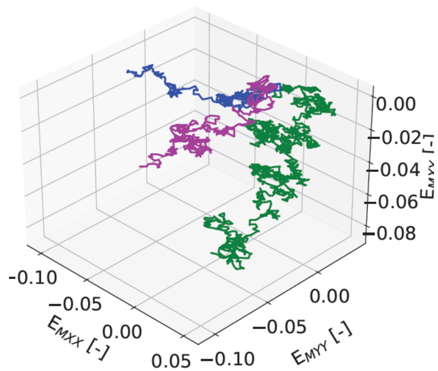
- Opportunities
  - Account for micro-structural effects on macro-scale responses
    - Failure origin
    - Micro-structural variation
  - Accelerate simulations
    - Makes virtual testing possible
- Challenges
  - Which surrogate/ML tool?
  - How to get a relevant database?
  - Quid extrapolating capabilities?

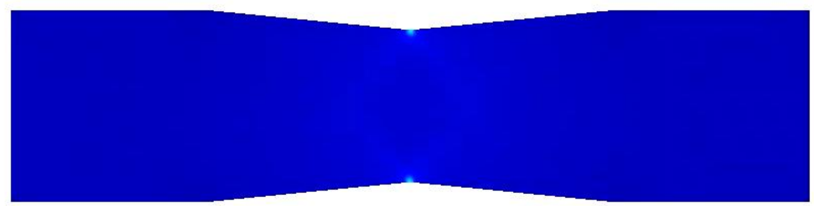


AI-accelerated multi-scale formalism

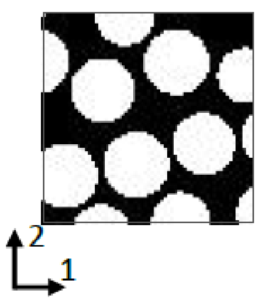
- 2 developments
  - Recurrent neural networks for composite failure analyses

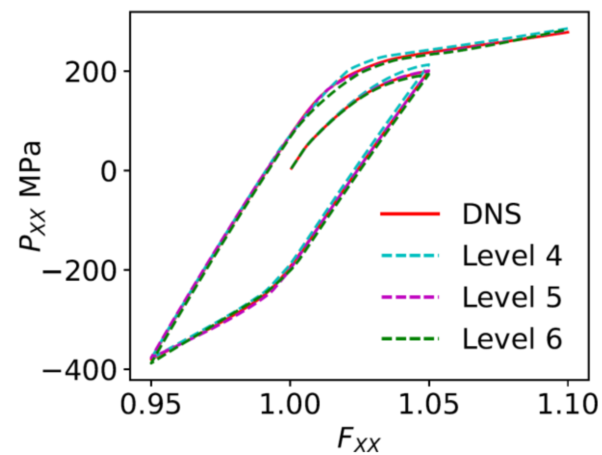




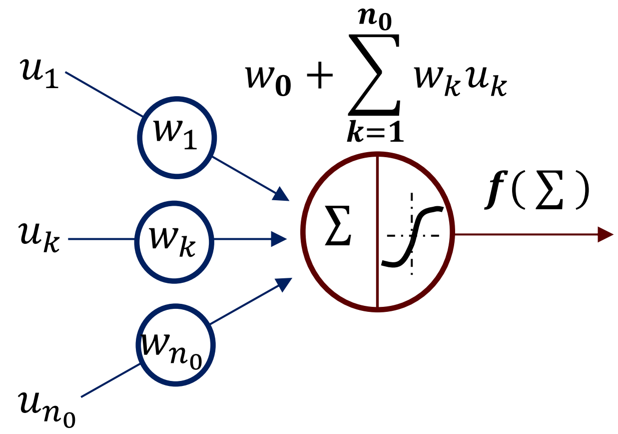


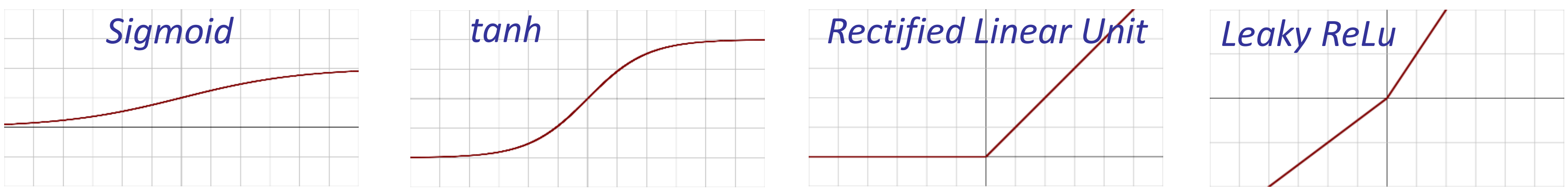
  - Stochastic deep material network



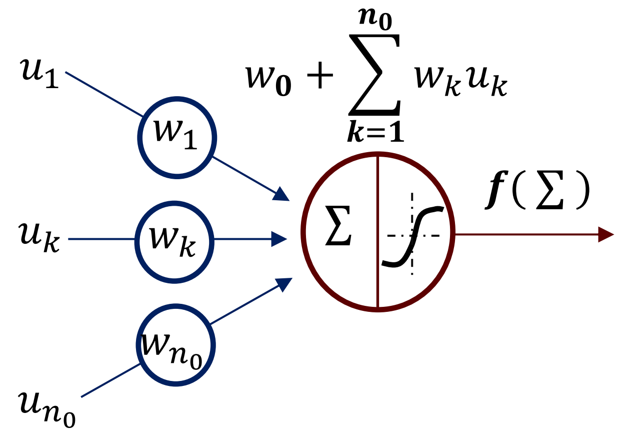


- Definition of the surrogate model
  - Artificial neuron
    - Non-linear function on n_0 inputs u_k
    - Requires evaluation of weights w_k
    - Requires definition of activation function f
  - Activation functions f

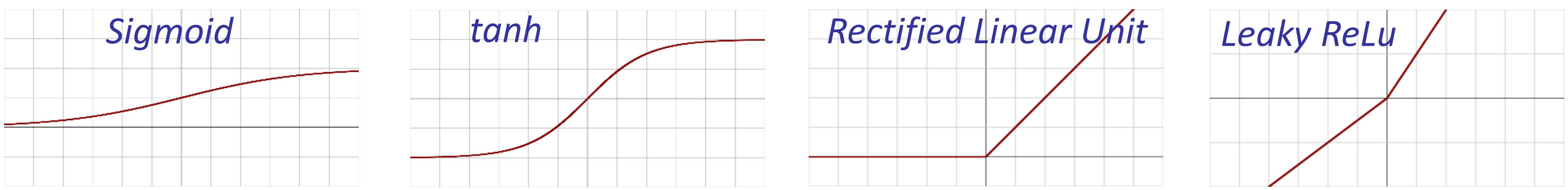




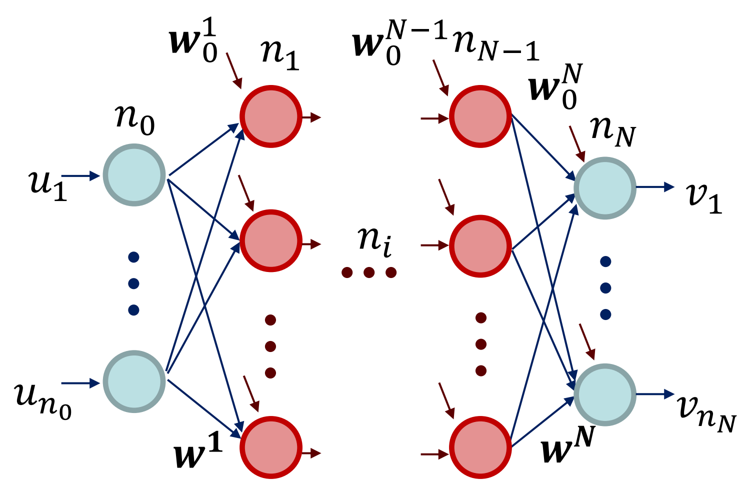
- Definition of the surrogate model
  - Artificial neuron
    - Non-linear function on n_0 inputs u_k
    - Requires evaluation of weights w_k
    - Requires definition of activation function f



  - Activation functions f



  - Feed-Forward Neuron Network
    - Simplest architecture
    - Layers of neurons
      - Input layer
      - $N-1$ hidden layers
      - Output layers
    - Mapping $\mathfrak{R}^{n_0} \rightarrow \mathfrak{R}^{n_N}$: $\boldsymbol{v} = \boldsymbol{g}(\boldsymbol{u})$



AI-accelerated multi-scale formalism

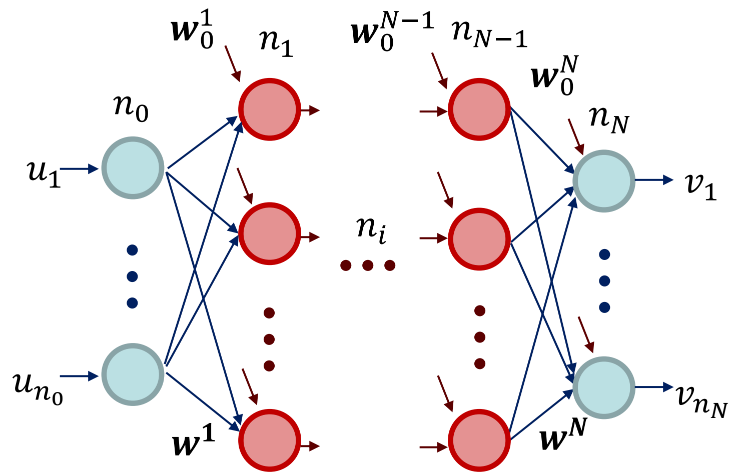
- Training
  - Evaluate
    - The weights w_{kj}^{i}, $k = 1..n_{i-1}, j = 1..n_i$
    - The bias w_0^i
    - Minimise error on prediction $\boldsymbol{v}$ vs. n observations (real) $\boldsymbol{v}^{(p)}$

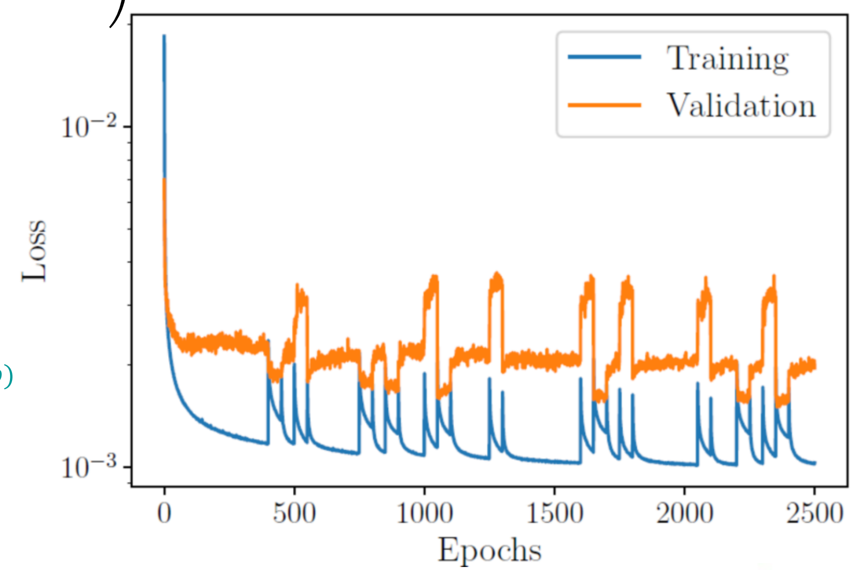
$$L_{\text{MSE}}\left(\mathbf{W}\right) = \frac{1}{n}\sum_{i}^{n}\left\|\boldsymbol{v}_i(\mathbf{W}) - \boldsymbol{v}_i^{(p)}\right\|^2$$

    - Requires an optimiser:

$$\mathbf{\Delta W} = -\mathcal{F}\left(\frac{\partial L_i(\mathbf{W})}{\partial \mathbf{W}}, \quad \left(\frac{\partial L_i(\mathbf{W})}{\partial \mathbf{W}}\right)^2, \text{batch size}, \ldots\right)$$

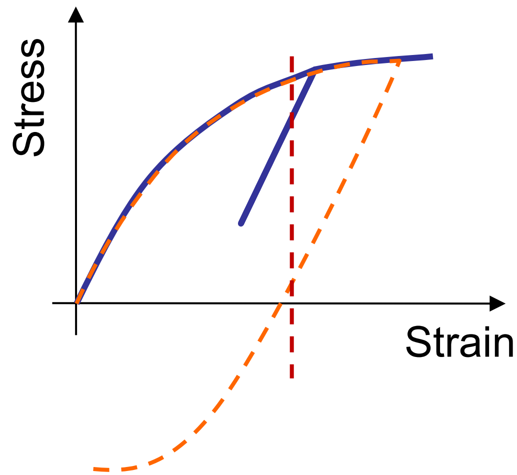
  - Training data
    - Input $\boldsymbol{u}^{(p)}$ & Output $\boldsymbol{v}^{(p)}$
- Testing
  - Use new data
    - Input $\boldsymbol{u}^{(p)}$ & Output $\boldsymbol{v}^{(p)}$
    - Verify prediction $\boldsymbol{v}$ vs. observations (real) $\boldsymbol{v}^{(p)}$





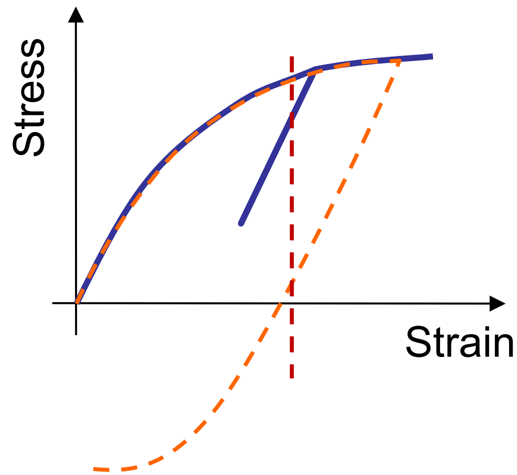
Recurrent Neural Network for multi-scale simulations

- Input / output definition
  - Input: Strain (history): $\mathbf{F}_{\mathrm{M}}$
  - Output: Stress (history): $\mathbf{P}_{\mathrm{M}}$
- Elasto-plastic material behaviour
  - No bijective strain-stress relation
    - Feed-forward NNW cannot be used
    - History should be accounted for

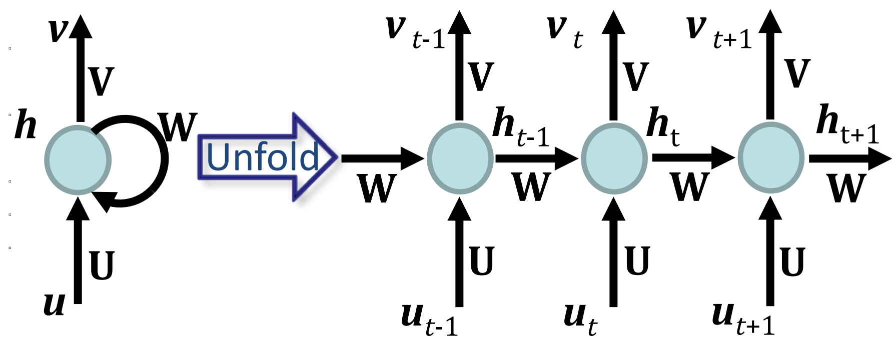


Recurrent Neural Network for multi-scale simulations

- Input / output definition
  - Input: Strain (history): $\mathbf{F}_{\mathrm{M}}$
  - Output: Stress (history): $\mathbf{P}_{\mathrm{M}}$



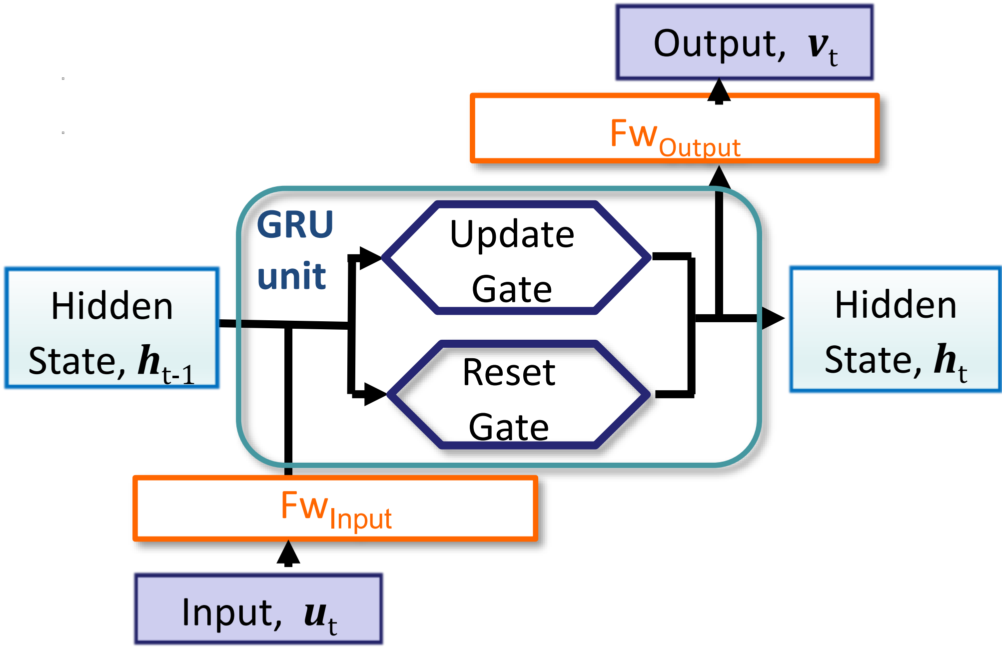
- Elasto-plastic material behaviour
  - No bijective strain-stress relation
    - Feed-forward NNW cannot be used
    - History should be accounted for



- Recurrent neural network
  - Allows a history dependent relation
    - Input: sequence $\boldsymbol{u}_t$
    - Output: sequence $\boldsymbol{v}_t = \boldsymbol{g}(\boldsymbol{u}_t, \boldsymbol{h}_{t-1})$
    - Internal variable $\boldsymbol{h}_t = \boldsymbol{g}(\boldsymbol{u}_t, \boldsymbol{h}_{t-1})$
  - Existing recurrent units
    - GRU, LSTM
    - Both developed for Nature Language Processing
    - …

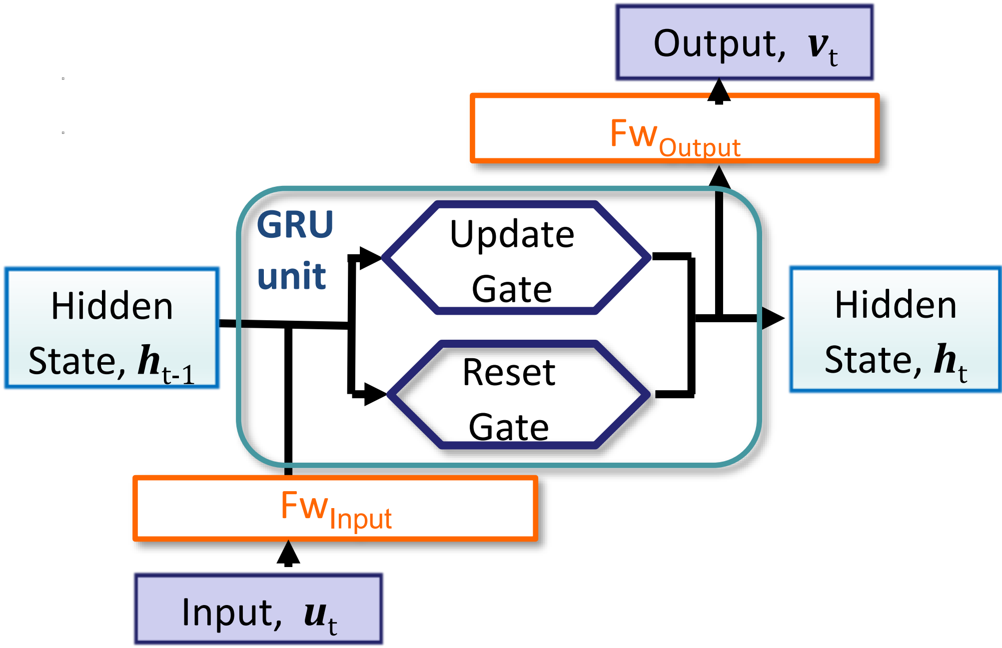
Recurrent Neural Network-accelerated multi-scale simulations

- 1 Gated Recurrent Unit (GRU)
  - Reset gate: select past information to be forgotten
  - Update gate: select past information to be passed along
- 2 feed-forward NNWs
  - FW_{Input} to treat inputs $\boldsymbol{u}_t$
  - FW_{Output} to produce outputs $\boldsymbol{v}_t$

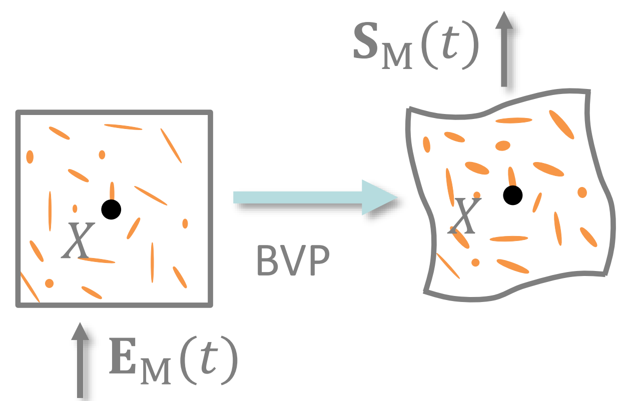


Recurrent Neural Network-accelerated multi-scale simulations

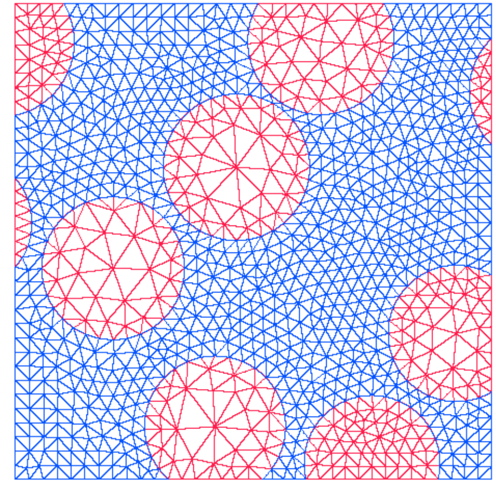
- 1 Gated Recurrent Unit (GRU)
  - Reset gate: select past information to be forgotten
  - Update gate: select past information to be passed along
- 2 feed-forward NNWs
  - FW_{Input} to treat inputs $\boldsymbol{u}_t$
  - $\text{FW}_{\text{Output}}$ to produce outputs $\boldsymbol{v}_t$

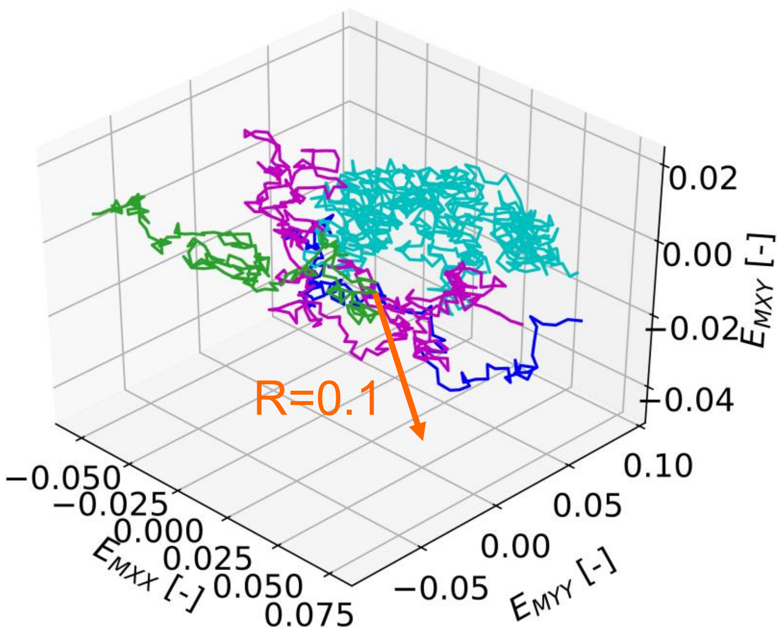


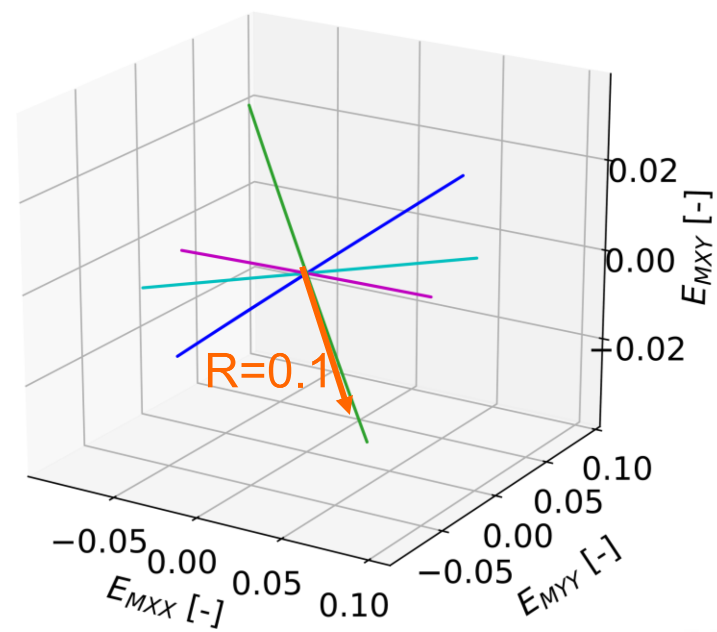
- Details
  - $\boldsymbol{u}_t$: homogenised GL strain $\mathbf{E}_\text{M}$ (symmetric)
  - $\boldsymbol{v}_t$: homogenised 2nd PK stress $\mathbf{S}_\text{M}$ (symmetric)
  - 100 hidden variables $\boldsymbol{h}_t$



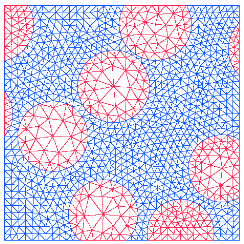
- Data generation
  - Elasto-plastic composite RVE
  - Training stage
    - Should cover full range of possible loading histories
    - Use random walking strategy (thousands)
    - Completed with random cyclic loading (tens)
    - Bounded by a sphere of 10% deformation

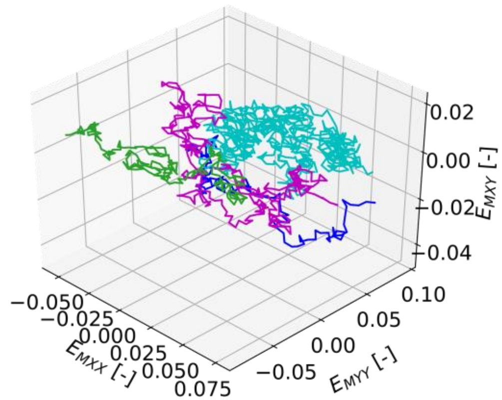


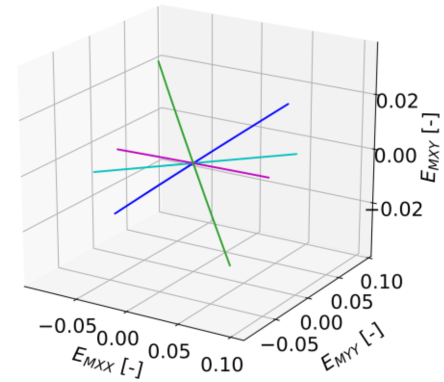


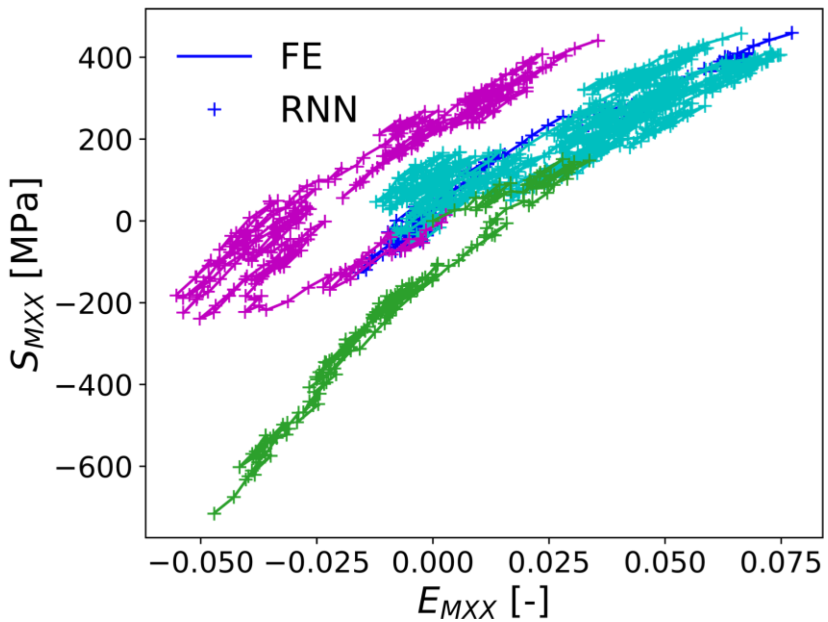


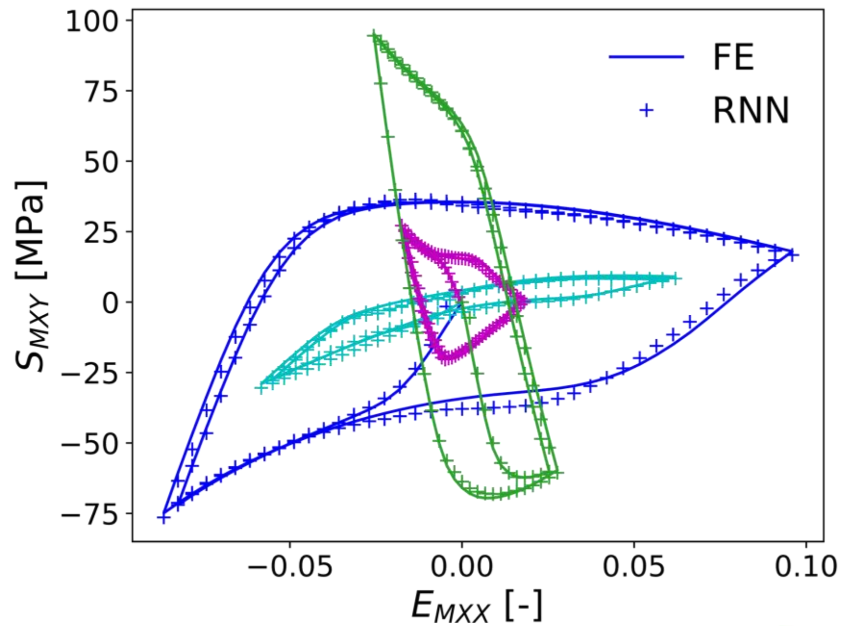
- Testing process (new data)
  - On random walk & cyclic loading







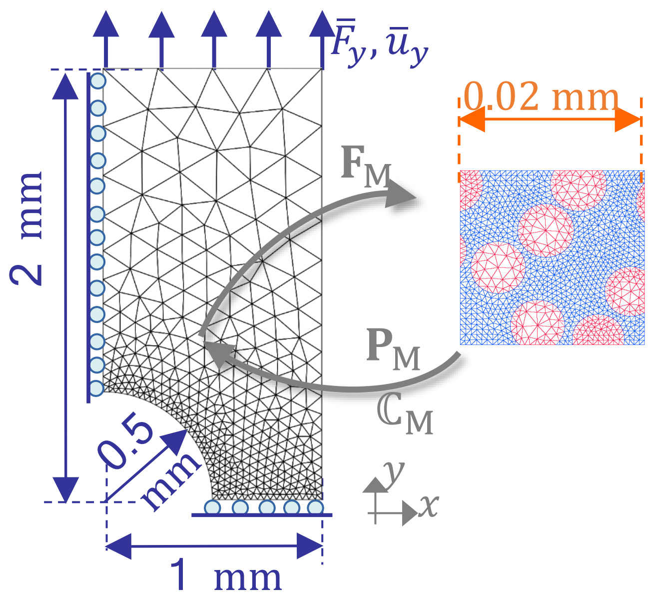


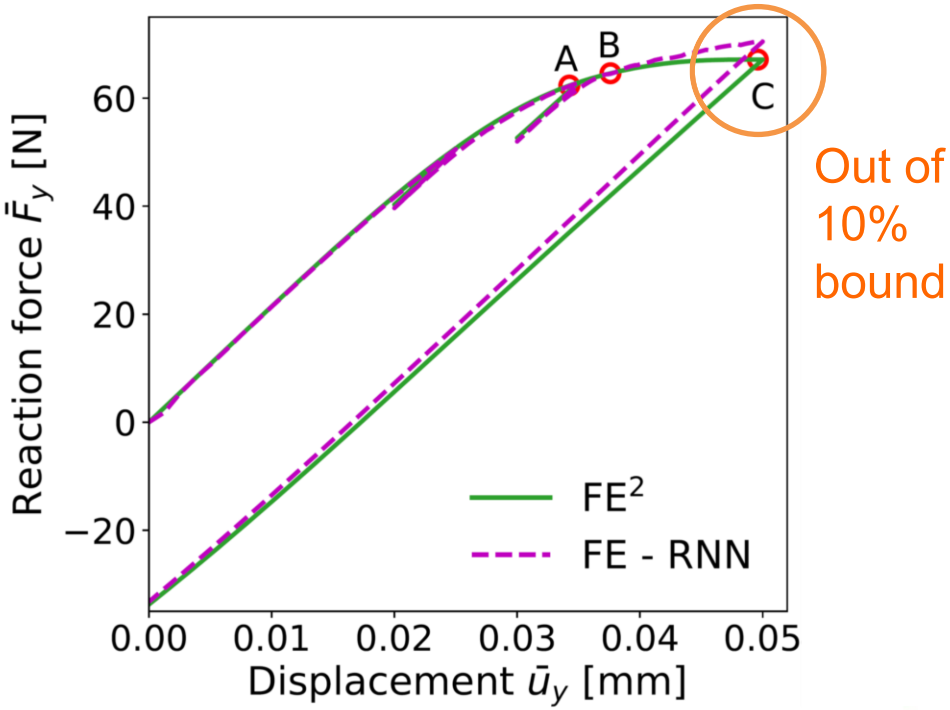


Recurrent Neural Network-accelerated multi-scale simulations

- Multiscale simulation
  - Elasto-plastic composite RVE
  - Comparison FE^2 vs. RNN-surrogate
  - Training data
    - Bounded at 10% deformation

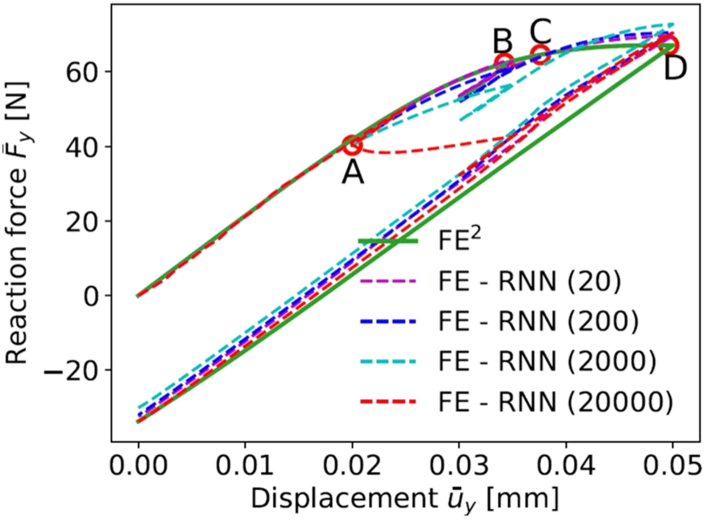
Off-line	FE^2	FE-RNN
Data generation	-	9000 x 2 h-cpu
Training	-	3 day-cpu
On-line	**FE^2**	**FE-RNN**
Simulation	18000 h-cpu	0.5 h-cpu





- What if online simulations use smaller increments?
  - Oscillations / loss of accuracy can appear with GRU, LSTM*
    - Both developed for Nature Language Processing
  - One needs to enforce self-consistency*
  - Need to replace the GRU/LSTM unit

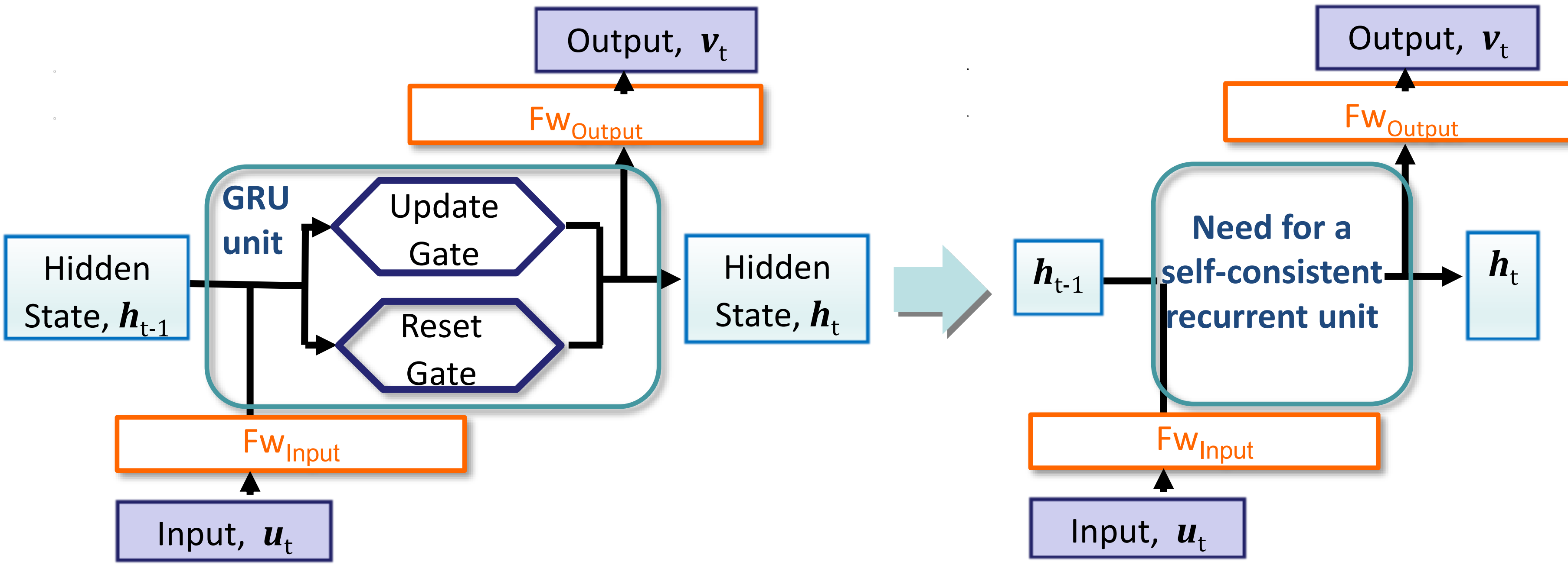
Simplified minimal recurrent unit



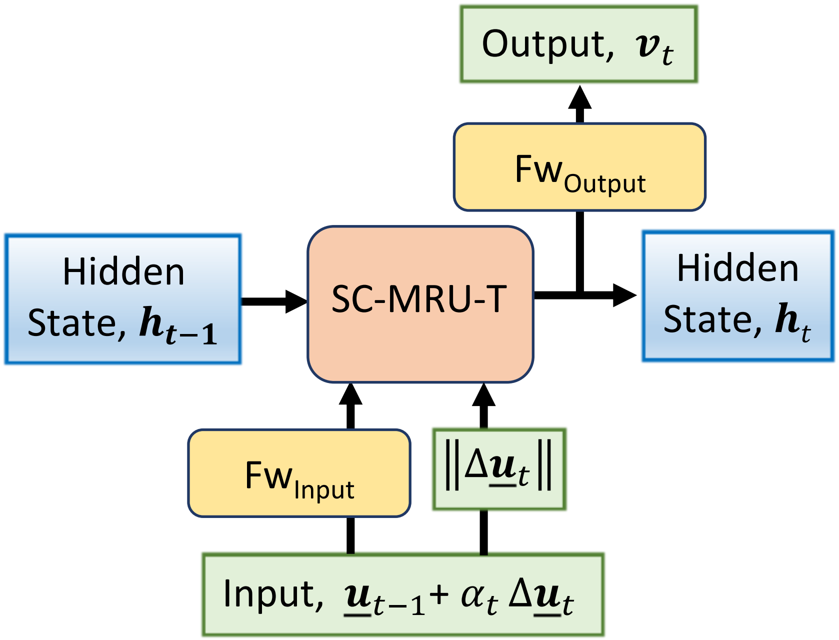
*Colin Bonatti, Dirk Mohr, On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids, Journal of the Mechanics and Physics of Solids, 158, 2022, 104697, https://doi.org/10.1016/j.jmps.2021.104697.

Self-Consistent Recurrent Neural Network for multi-scale simulations

- Self-Consistency reinforcement through ad hoc recurrent unit/cell
  - SC-cell originally to surrogate a constitutive model
  - Can we develop easy and fast to train surrogate for RVE responses?

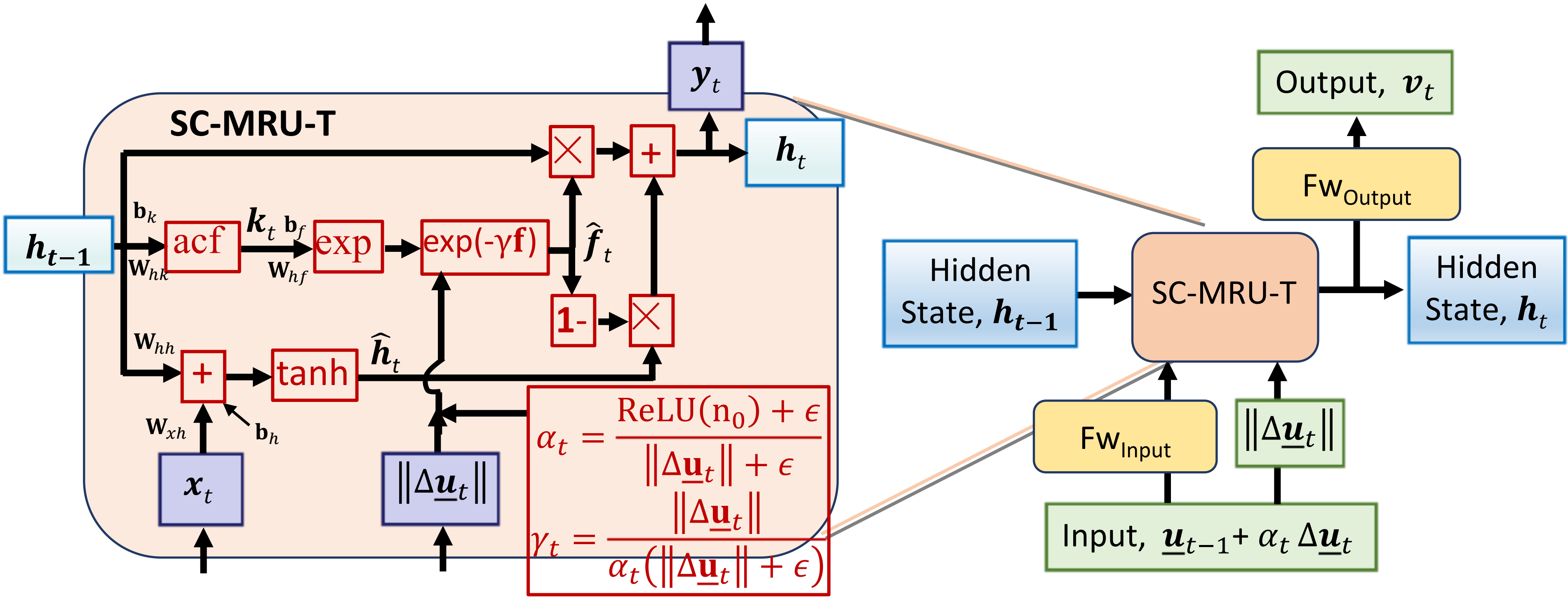


- The Self-Consistent Recurrent Neural Network based on a SC unit



- The **total form of the input variable**
  - Use as input $\underline{\boldsymbol{u}}_{t-1} + \alpha_t \Delta \underline{\boldsymbol{u}}_t$
  - Lift the size of the input to the one of the hidden state with $\mathrm{Fw}_{\mathrm{input}}$
- & **increment norm** $\|\Delta \boldsymbol{u}_t\|$
  - For $\|\Delta \boldsymbol{u}_t\| \rightarrow 0$ the hidden state should not be updated
  - For larger $\|\Delta \boldsymbol{u}_t\|$, the hidden state should be updated

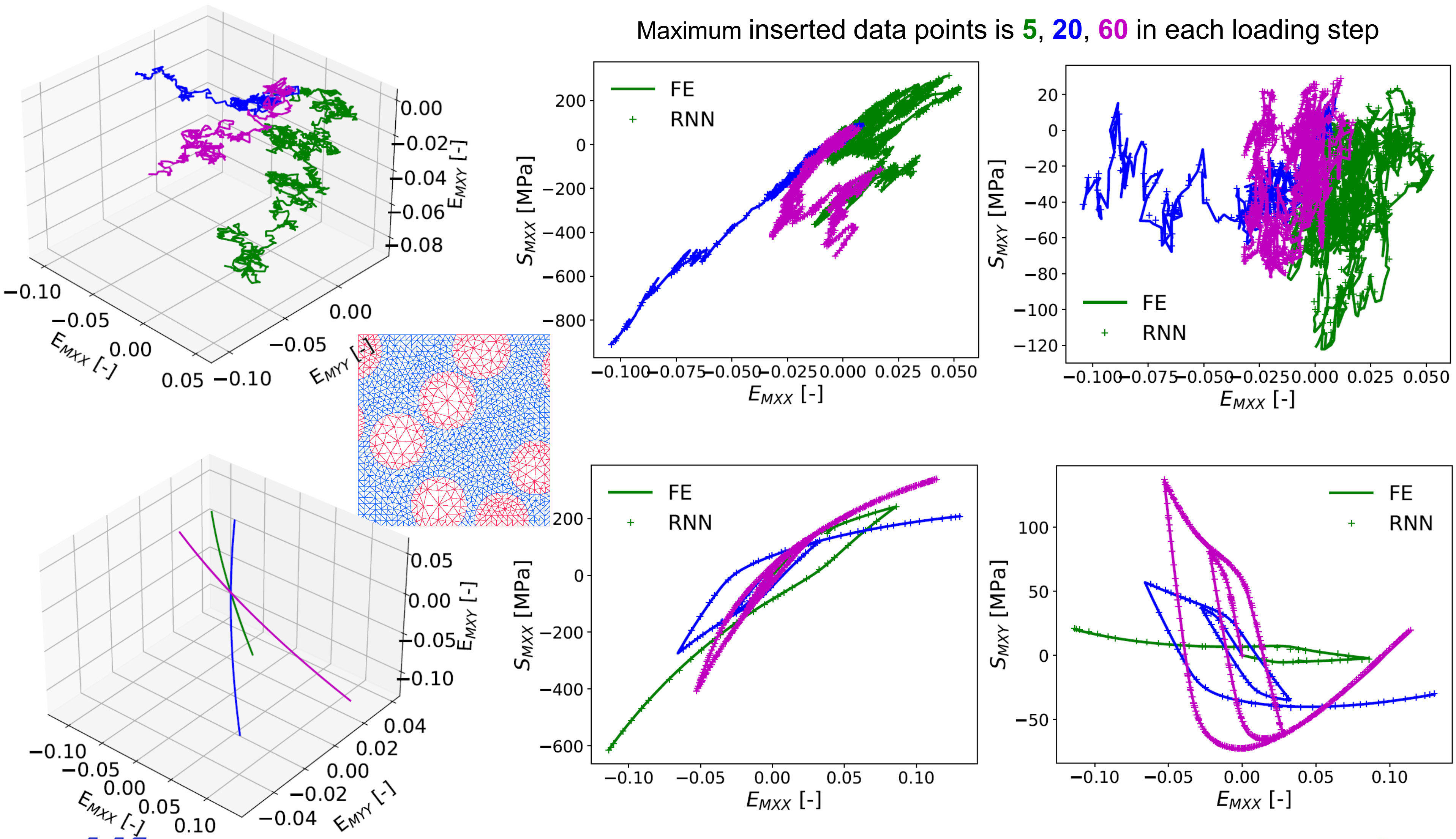
- The Self-Consistent Minimal Recurrent Unit



- **Self-consistency enforced**
  - Double exponential function $f_t = \exp[W_f k_t + b_f] > 0$ & $\hat{f}_t = \exp[-\gamma(\|\Delta u_t\|)\, f_t] \in [0, 1]$
  - h_t is an element-wise interpolation (ratio $\hat{f}_t$ dependent on $\|\Delta u_t\|$) between
    - previous value h_{t-1}
    - & update hidden variables $\hat{h}_t$
  - n_0 is a learnable parameter

- SC-MRU-T: Testing data with inserted extra-points

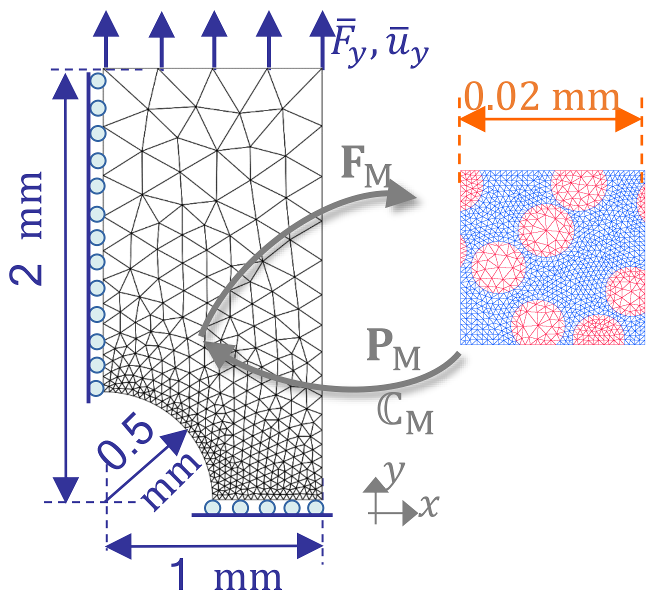
Maximum inserted data points is **5**, **20**, **60** in each loading step

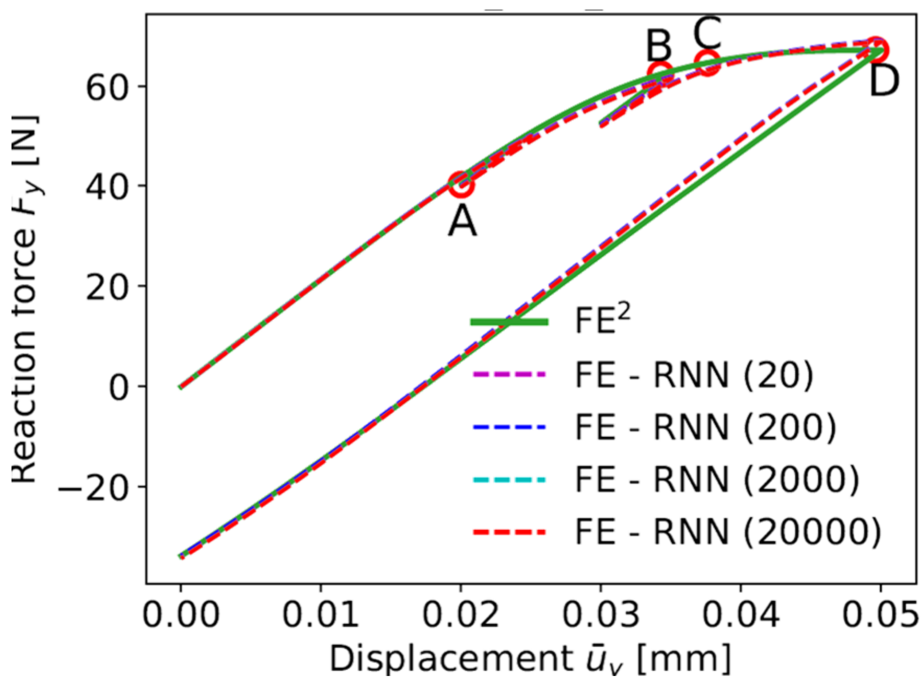


Self-Consistent Recurrent Neural Network for multi-scale simulations

- FE2 vs. FE-RNN:
  - Change in the increment size between points A&B
  - Training is much faster with the SC-MRU cell

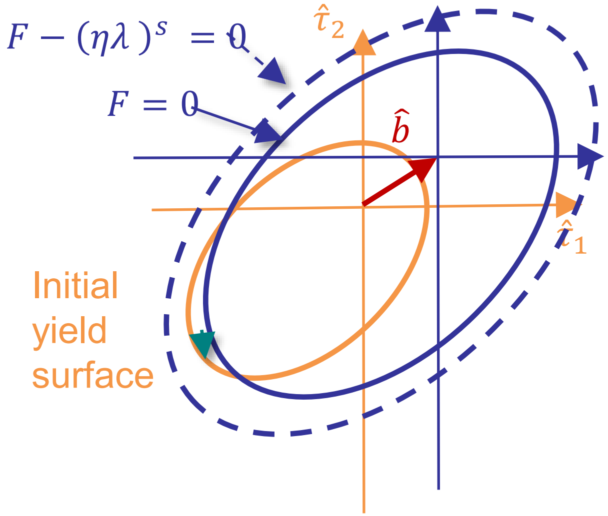
Off-line	FE^2	FE-GRU	FE-SC
Data generation	-	9000 x 2 h-cpu	23 500 h-cpu
Training	-	3 day-cpu	<10 h-cpu
On-line	**FE^2**	**FE-GRU**	**FE-SC**
Simulation	18000 h-cpu	0.5 h-cpu	0.4 h-cpu

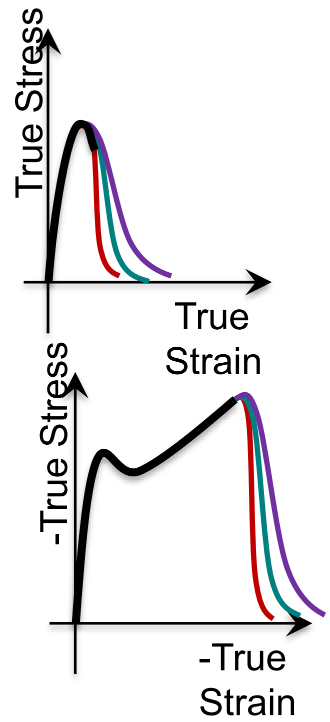


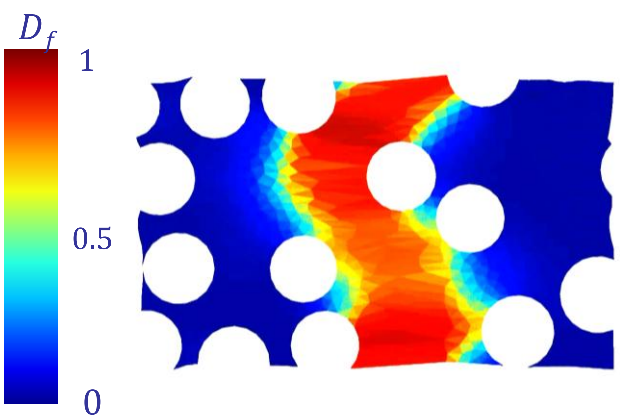


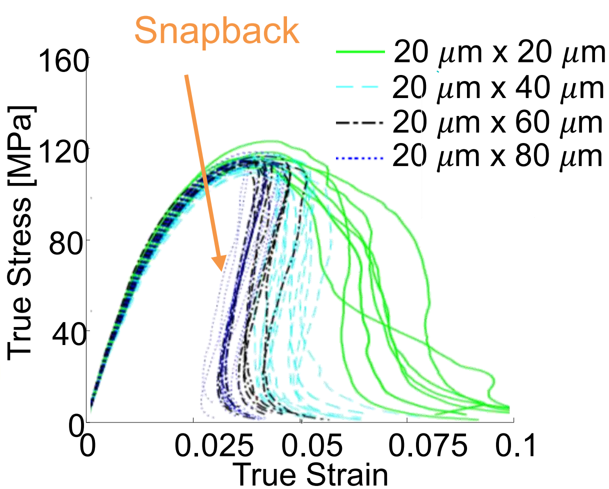
AI-accelerated multi-scale analysis of composite failure

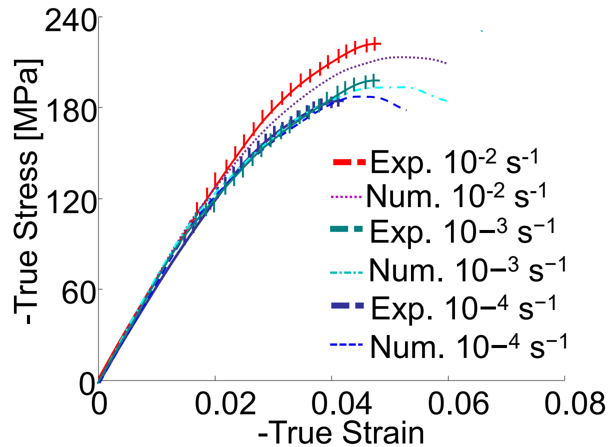
- RVE simulations:
  - Complex material model for resin
    - (Visco-Elasticity-)Visco-plasticity with pressure dependency
    - Triaxiality-dependent damage models
    - Non-local damage models
  - Transverse anisotropy for fibre
    - Failure with phase-field
  - Experimentally validated







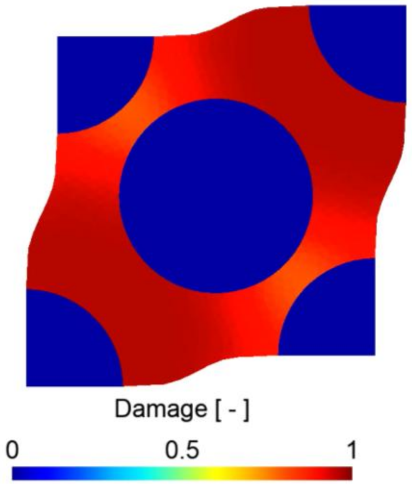


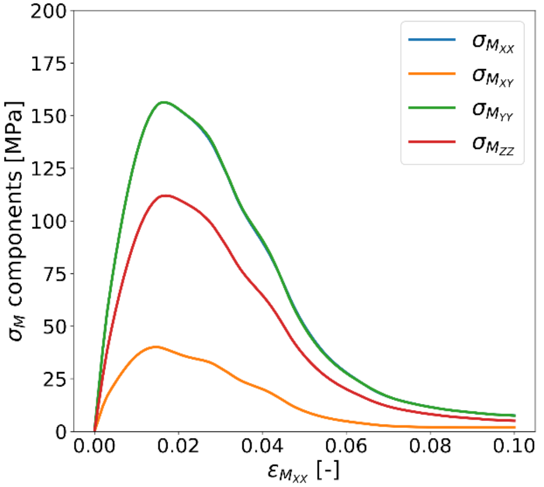


AI-accelerated multi-scale analysis of composite failure

- Challenges to develop the AI surrogate:
  - We need an extensive data-base
    - Simulations should reach total failure
    - Resin is brittle ➡ snapback

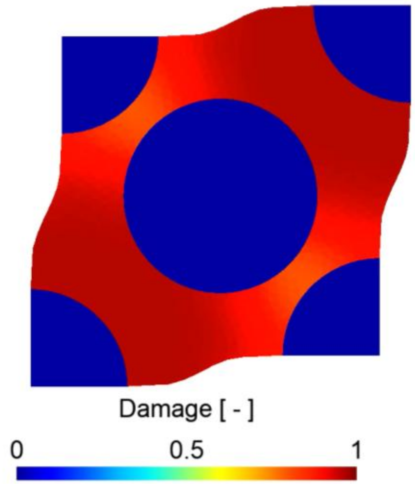
➡ (R)VE size is reduced

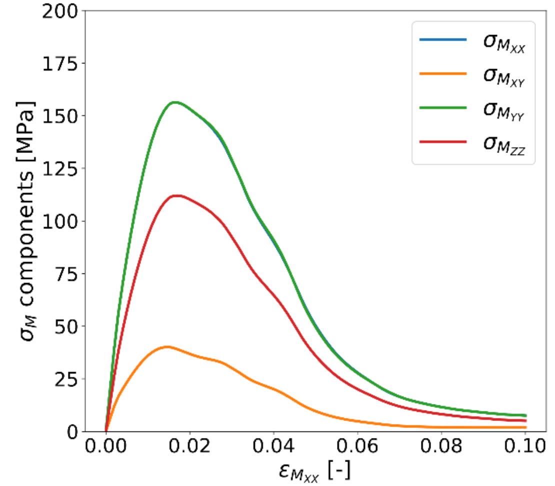




AI-accelerated multi-scale analysis of composite failure

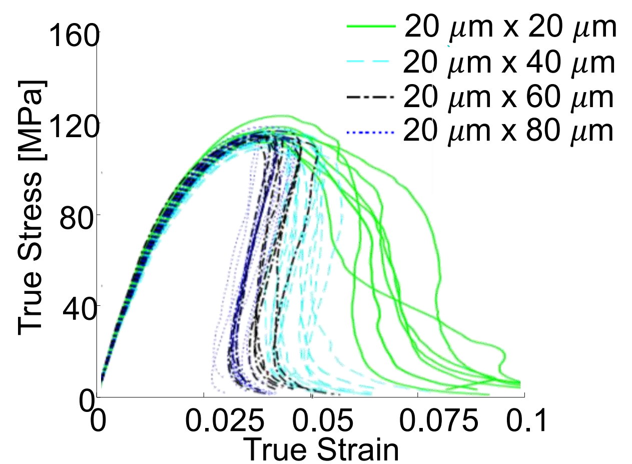
- Challenges to develop the AI surrogate:
  - We need an extensive data-base
    - Simulations should reach total failure
    - Resin is brittle ➔ snapback
  - ➔ (R)VE size is reduced





  - Homogenized curves are meaningless
    - They exhibit a strain softening
    - Softening depends on the RVE size
    - Only physical value is fracture energy G_c
  - ➔ We need to regularize the homogenized response

$$\tilde{\alpha} - l_c^2 \, \nabla^2 \tilde{\alpha} = \alpha$$



↳ Parameter tuned to recover the fracture energy G_c

- Development of an embedding non-local framework:
  - Identification of local/non-local variables
    - Plastic strain tensor (from unloading in apparent space)

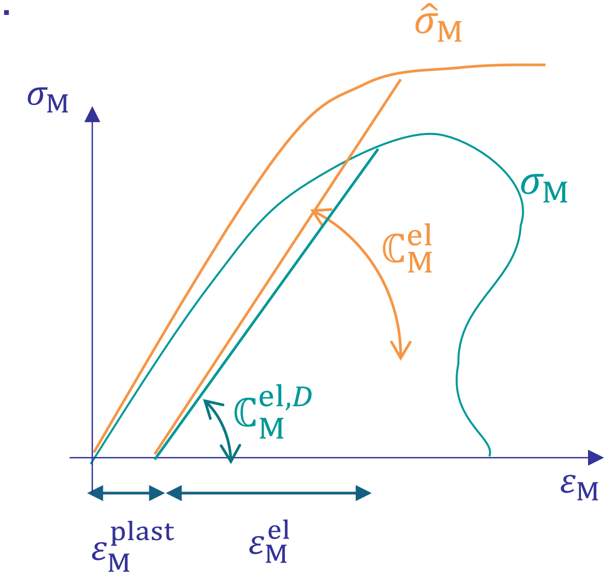
$$\boldsymbol{\varepsilon}_{\mathrm{M}}^{\mathrm{plast}} = \boldsymbol{\varepsilon}_{\mathrm{M}} - \mathbb{C}_{\mathrm{M}}^{\mathrm{el},D^{-1}} : \boldsymbol{\sigma}_{\mathrm{M}}$$

    - Component-wise damage (to be reduced*)

$$D_{\mathrm{M}_{ijkl}} = 1 - \frac{C_{\mathrm{M}_{ijkl}}^{\mathrm{el},D}}{C_{\mathrm{M}_{ijkl}}^{\mathrm{el}}}$$

    - New macro-scale Helmholtz equations

$$D_{\mathrm{M}_a} - l_d^2 \boldsymbol{\nabla}^2 \widetilde{D}_{\mathrm{M}_a} = D_{\mathrm{M}_a}$$
$$\tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} - l_p^2 \boldsymbol{\nabla}^2 \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} = \varepsilon_{\mathrm{M}_b}^{\mathrm{plast}}$$



*J. Yvonnet, Q-C. He, P. Li, Reducing internal variables and improving efficiency in data-driven modelling of anisotropic damage from RVE simulations, Computational Mechanics 72 (2023), p 37-55

AI-accelerated multi-scale analysis of composite failure

- Development of an embedding non-local framework:
  - Identification of local/non-local variables
    - Plastic strain tensor (from unloading in apparent space)

      $$\boldsymbol{\varepsilon}_{\mathrm{M}}^{\mathrm{plast}} = \boldsymbol{\varepsilon}_{\mathrm{M}} - {\mathbb{C}_{\mathrm{M}}^{\mathrm{el},D}}^{-1} : \boldsymbol{\sigma}_{\mathrm{M}}$$

    - Component-wise damage (to be reduced*)

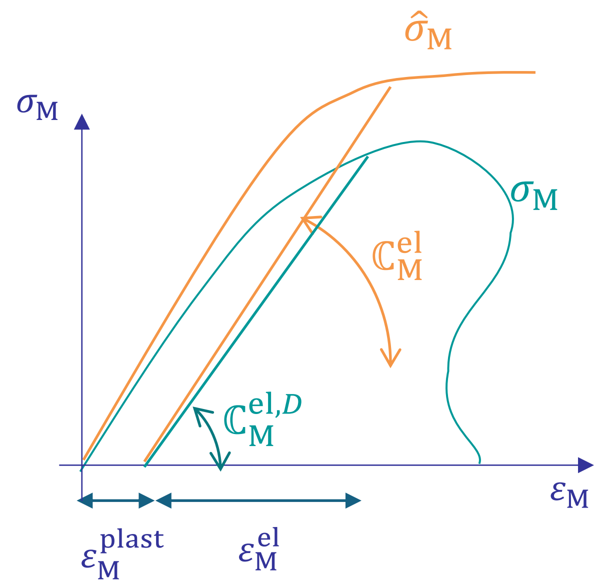
      $$D_{\mathrm{M}_{ijkl}} = 1 - \frac{C_{\mathrm{M}_{ijkl}}^{\mathrm{el},D}}{C_{\mathrm{M}_{ijkl}}^{\mathrm{el}}}$$

    - New macro-scale Helmholtz equations

      $$D_{\mathrm{M}_a} - l_d^2 \nabla^2 \widetilde{D}_{\mathrm{M}_a} = D_{\mathrm{M}_a}$$
      $$\tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} - l_p^2 \boldsymbol{\nabla}^2 \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} = \varepsilon_{\mathrm{M}_b}^{\mathrm{plast}}$$

  - Reconstructed (mesh-insensitive) apparent stress

    $$\boldsymbol{\sigma}_{\mathrm{M}_{\mathrm{rec}}} = \left[(1 - \widetilde{\mathbb{D}}_{\mathrm{M}}) \odot \mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right] : \left(\boldsymbol{\varepsilon}_{\mathrm{M}} - \tilde{\boldsymbol{\varepsilon}}_{\mathrm{M}}^{\mathrm{plast}}\right)$$



*J. Yvonnet, Q-C. He, P. Li, Reducing internal variables and improving efficiency in data-driven modelling of anisotropic damage from RVE simulations, Computational Mechanics 72 (2023), p 37-55

- Development of an embedding non-local framework:
  - Identification of local/non-local variables
    - Plastic strain tensor (from unloading in apparent space)

$$\boldsymbol{\varepsilon}_{\mathrm{M}}^{\mathrm{plast}} = \boldsymbol{\varepsilon}_{\mathrm{M}} - {\mathbb{C}_{\mathrm{M}}^{\mathrm{el},D}}^{-1} : \boldsymbol{\sigma}_{\mathrm{M}}$$

    - Component-wise damage (to be reduced*)

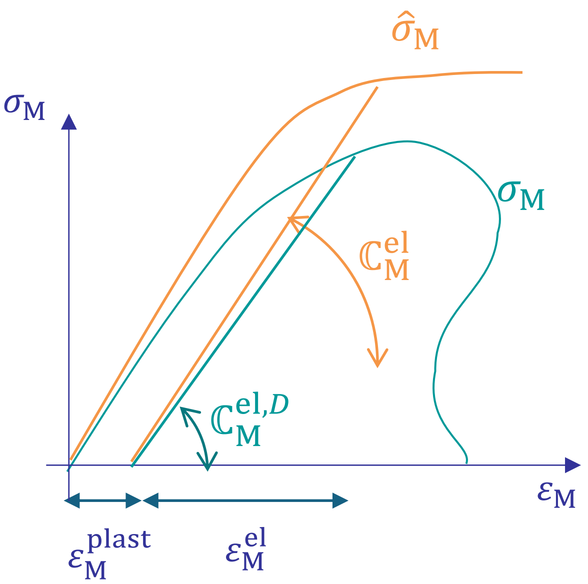
$$D_{\mathrm{M}_{ijkl}} = 1 - \frac{C_{\mathrm{M}_{ijkl}}^{\mathrm{el},D}}{C_{\mathrm{M}_{ijkl}}^{\mathrm{el}}}$$

    - New macro-scale Helmholtz equations

$$D_{\mathrm{M}_a} - l_d^2 \nabla^2 \widetilde{D}_{\mathrm{M}_a} = D_{\mathrm{M}_a}$$
$$\tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} = \varepsilon_{\mathrm{M}_b}^{\mathrm{plast}}$$

  - Reconstructed (mesh-insensitive) apparent stress

$$\boldsymbol{\sigma}_{\mathrm{M}_{\mathrm{rec}}} = \left[(1 - \widetilde{\mathbb{D}}_{\mathrm{M}}) \odot \mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right] : \left(\boldsymbol{\varepsilon}_{\mathrm{M}} - \tilde{\boldsymbol{\varepsilon}}_{\mathrm{M}}^{\mathrm{plast}}\right)$$



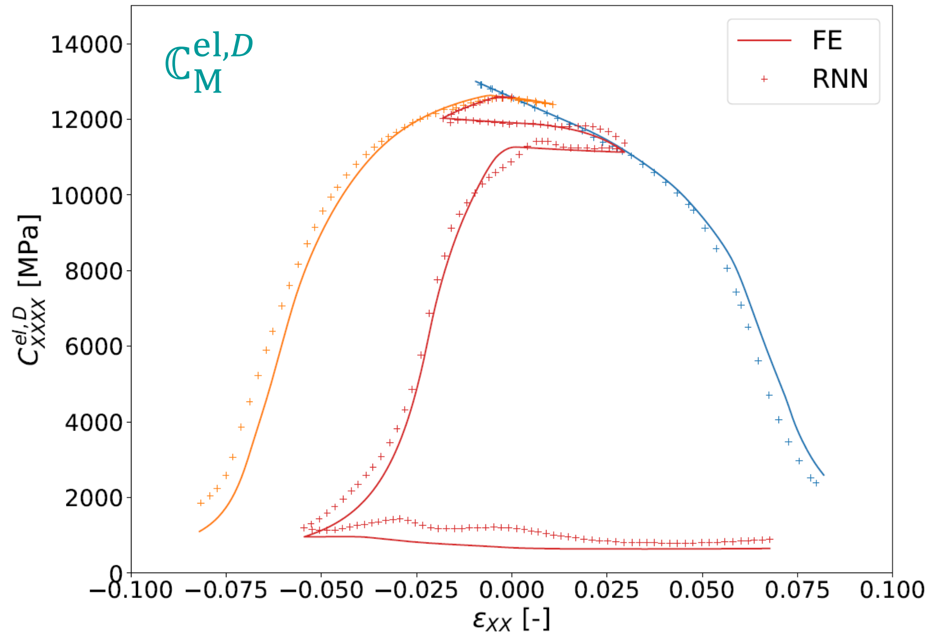
Neural networks to be developed:

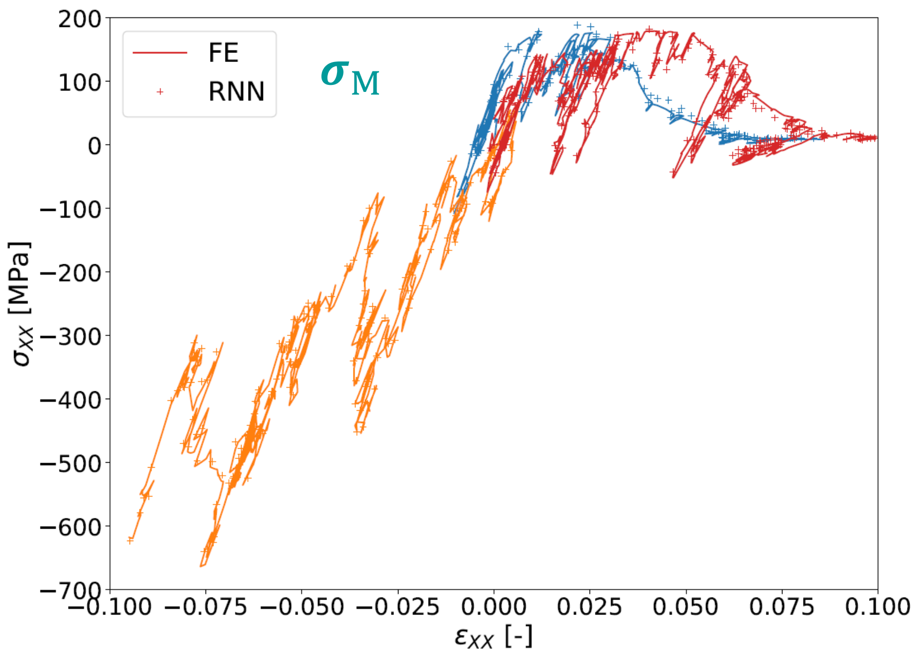
- Inputs: $\underline{\boldsymbol{u}}_t = \{\boldsymbol{\varepsilon}_{\mathrm{M}}\}_t$
- Outputs: $\underline{\boldsymbol{v}}_t = \left\{\boldsymbol{\sigma}_{\mathrm{M}}, \mathbb{C}_{\mathrm{M}}^{\mathrm{el},D}, \mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right\}_t$

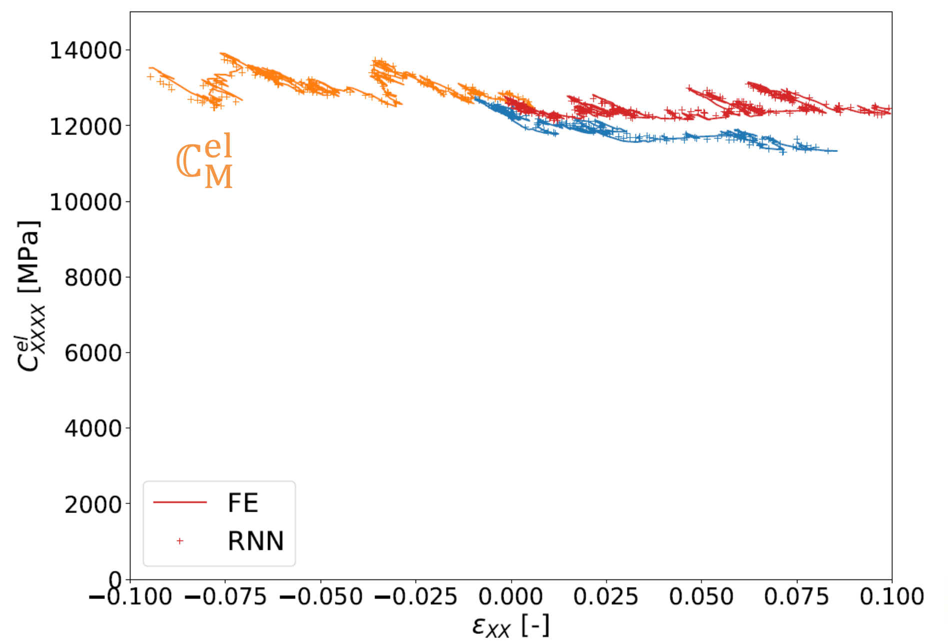
*J. Yvonnet, Q-C. He, P. Li, Reducing internal variables and improving efficiency in data-driven modelling of anisotropic damage from RVE simulations, Computational Mechanics 72 (2023), p 37-55

AI-accelerated multi-scale analysis of composite failure

- Training data-base
  - ~10 000 random paths
  - ~ 5000 cyclic paths
- Testing the neural networks
  - Inputs: $\underline{\boldsymbol{u}}_t = \{\boldsymbol{\varepsilon}_{\mathrm{M}}\}_t$
  - Outputs: $\underline{\boldsymbol{v}}_t = \left\{\boldsymbol{\sigma}_{\mathrm{M}}, \mathbb{C}_{\mathrm{M}}^{\mathrm{el},D}, \mathbb{C}_{\mathrm{M}}^{\mathrm{el}}\right\}_t$

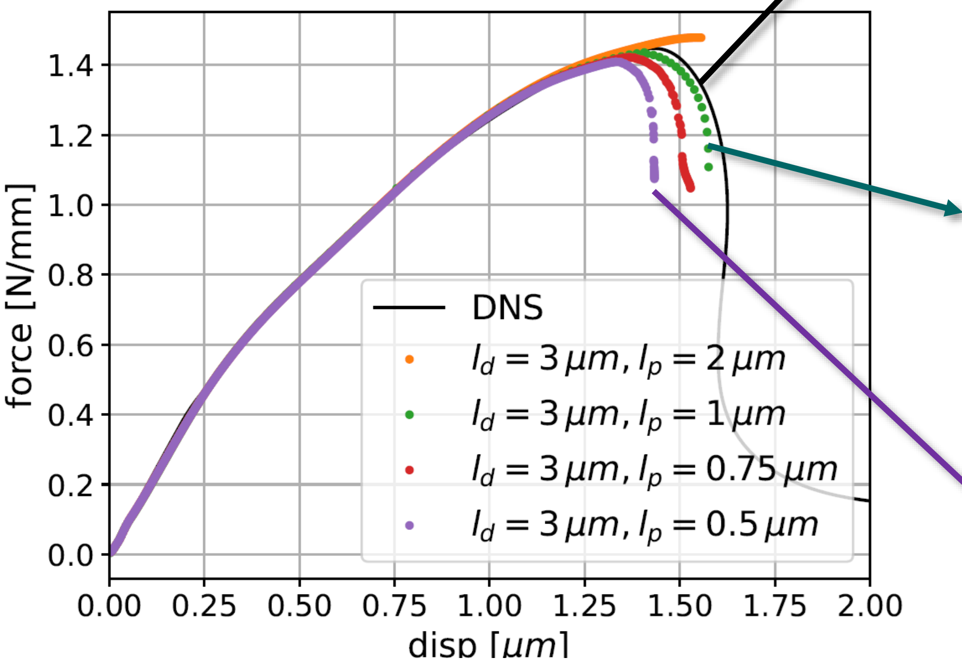


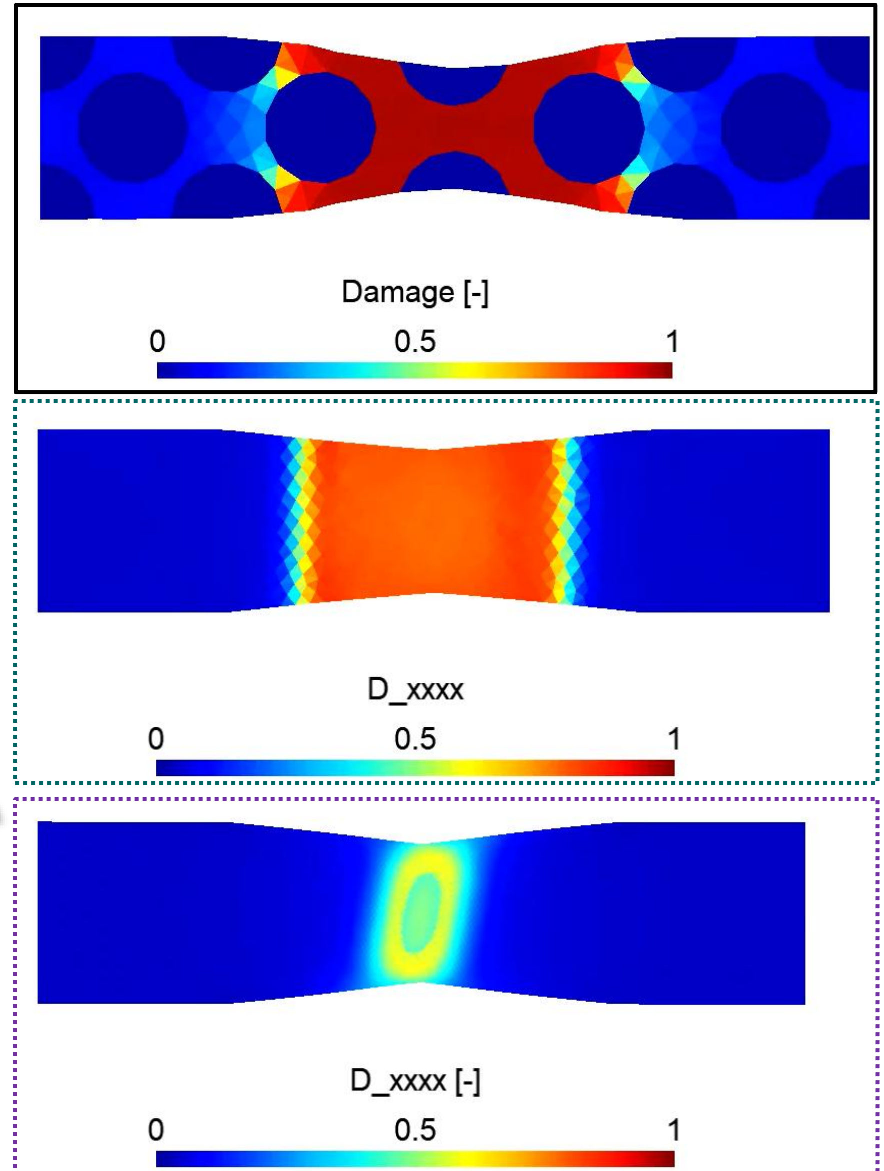




AI-accelerated multi-scale analysis of composite failure

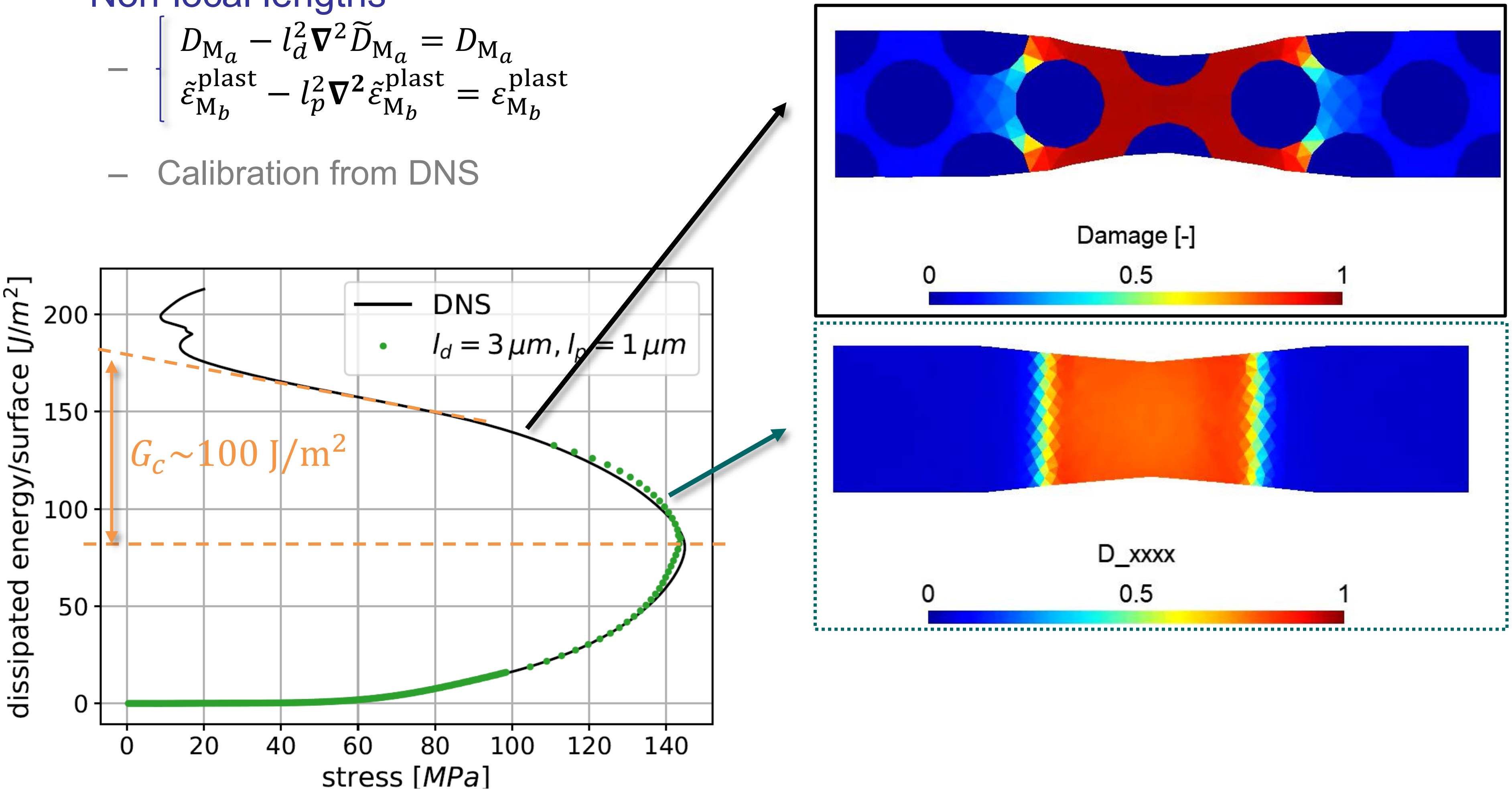
- Non-local lengths
  - $$\begin{cases} D_{\mathrm{M}_a} - l_d^2 \nabla^2 \widetilde{D}_{\mathrm{M}_a} = D_{\mathrm{M}_a} \\ \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} = \varepsilon_{\mathrm{M}_b}^{\mathrm{plast}} \end{cases}$$
  - Calibration from DNS





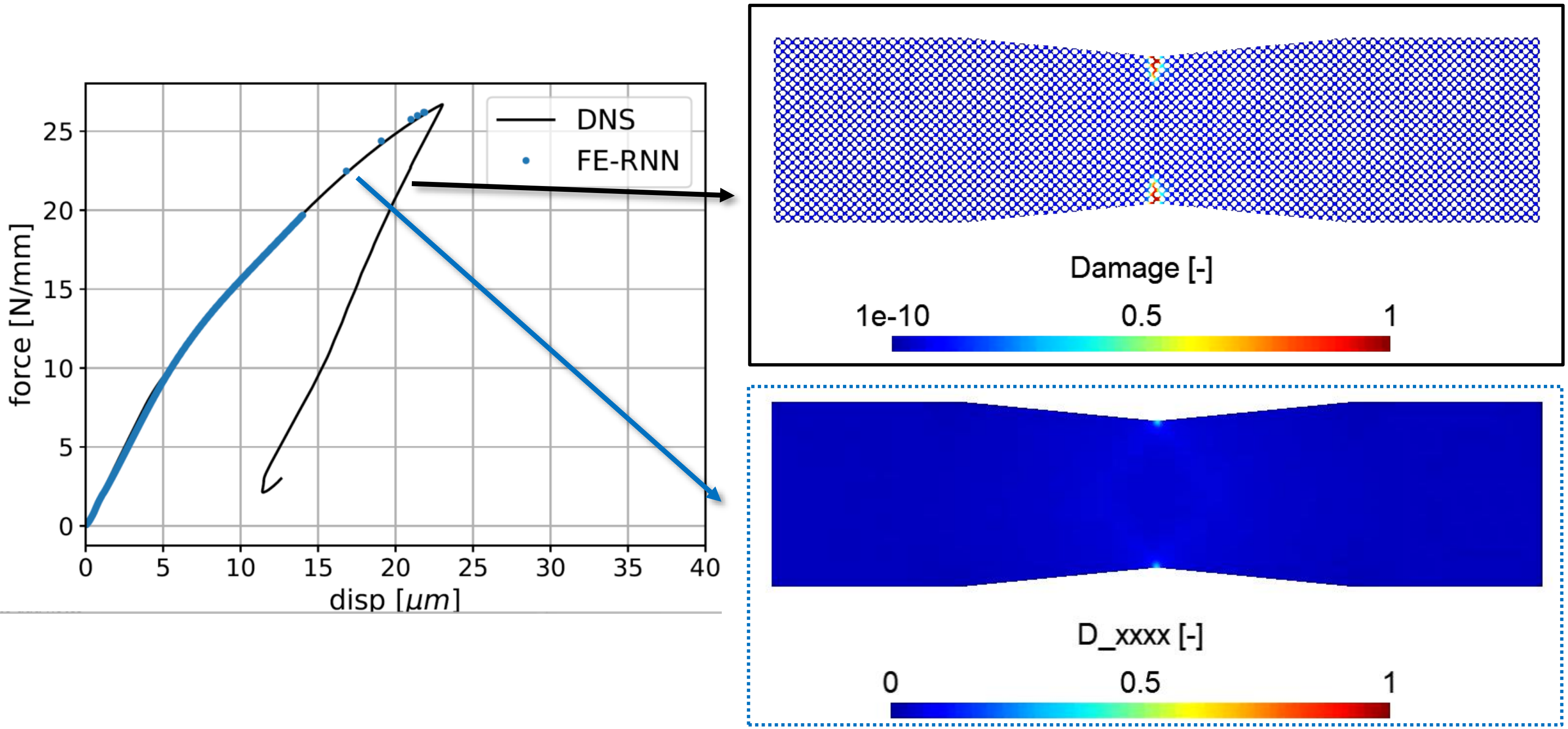
AI-accelerated multi-scale analysis of composite failure

- Non-local lengths
  - $$\begin{cases} D_{\mathrm{M}_a} - l_d^2 \nabla^2 \widetilde{D}_{\mathrm{M}_a} = D_{\mathrm{M}_a} \\ \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{\mathrm{M}_b}^{\mathrm{plast}} = \varepsilon_{\mathrm{M}_b}^{\mathrm{plast}} \end{cases}$$
  - Calibration from DNS



AI-accelerated multi-scale analysis of composite failure

- Larger sample
  - DNS vs. RNN
    - RNN trained on volume elements
    - Calibrated macro-scale non-local lengths



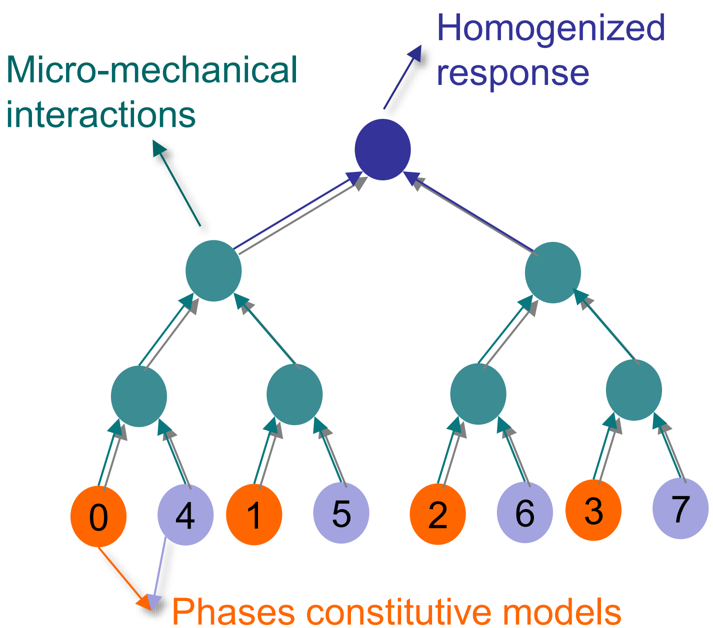
- Use of RNN
  - Good accuracy/performance for RNN
    - When designed for computational mechanics
    - Trained with adequate synthetic data base
  - Limitations: No extrapolation capabilities
    - Requires extensive synthetic data base
    - Generalizing for arbitrary micro-structure/phase response requires unreachable data bases
    - Validity range is limited

AI-accelerated multi-scale analysis of composites

- Use of RNN
  - Good accuracy/performance for RNN
    - When designed for computational mechanics
    - Trained with adequate synthetic data base
  - Limitations: No extrapolation capabilities
    - Requires extensive synthetic data base
    - Generalizing for arbitrary micro-structure/phase response requires unreachable data bases
    - Validity range is limited
- Can we limit the size of the data base?
  - Introduce Physics in neural networks
    - Predicting energy potentials (Yvonnet, Masi, Stefanou, …)
    - Introducing material laws (Maia, van der Meer, …
    - Modifying loss function (Cueto, …)
    - ….

AI-accelerated multi-scale analysis of composites

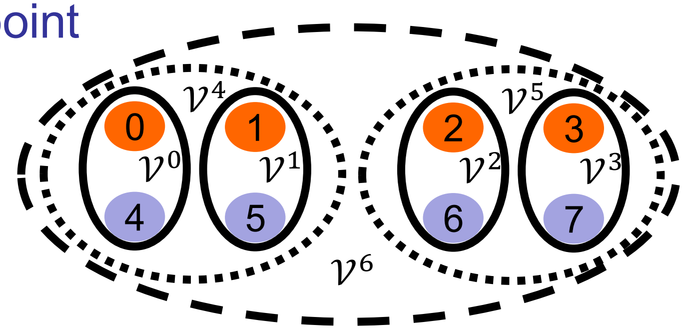
- Use of RNN
  - Good accuracy/performance for RNN
    - When designed for computational mechanics
    - Trained with adequate synthetic data base
  - Limitations: No extrapolation capabilities
    - Requires extensive synthetic data base
    - Generalizing for arbitrary micro-structure/phase response requires unreachable data bases
    - Validity range is limited
- Can we limit the size of the data base?
  - Introduce Physics in neural networks
    - Predicting energy potentials (Yvonnet, Masi, Stefanou, …)
    - Introducing material laws (Maia, van der Meer, …
    - Modifying loss function (Cueto, …)
    - ….
  - Alternative solution: build a network of physical blocks
    - Deep Material Network (Liu, …)*



*Z. Liu, C. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, Comput. Methods Appl. Mech. Engrg. 345 (2019) 1138–1168

- Deep Material Networks from the interaction viewpoint
  - Strain/Stress averaging

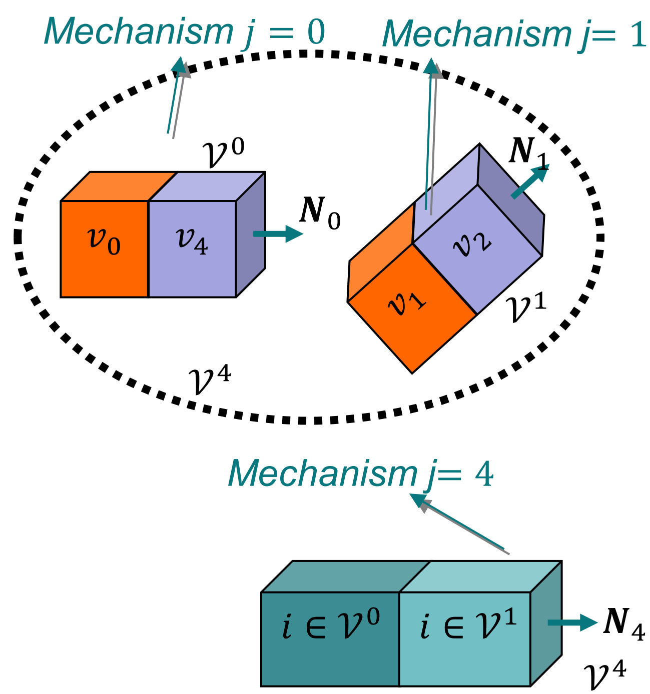
$$\mathbf{P}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{P}_i(t)$$
$$\mathbf{F}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{F}_i(t)$$
$$v_i = \frac{V_i}{V_{\mathrm{M}}}$$



  - Fluctuation filed $\boldsymbol{u}'$

$$\mathbf{F}_{\mathrm{m}}(\boldsymbol{X}_{\mathrm{m}}) = \mathbf{F}_{\mathrm{M}} + \boldsymbol{u}'(\boldsymbol{X}_{\mathrm{m}}) \otimes \nabla_0$$

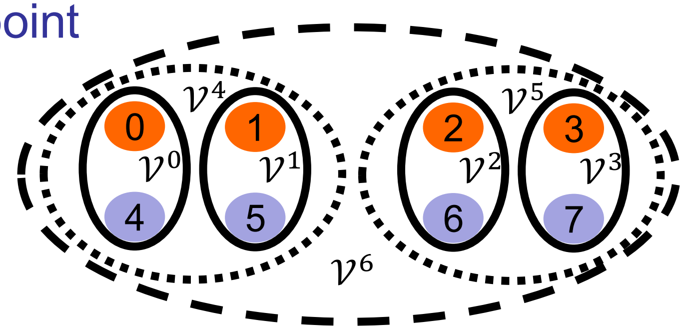
$$\mathbf{F}_i = \mathbf{F}_{\mathrm{M}} + \frac{1}{V_i} \sum_{(i,j)} s_j^i \, \boldsymbol{u}_j' \otimes \boldsymbol{N}_j^i \ , \quad i = 0..7$$



- Deep Material Networks from the interaction viewpoint
  - Strain/Stress averaging

$$\mathbf{P}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{P}_i(t) \qquad v_i = \frac{V_i}{V_{\mathrm{M}}}$$

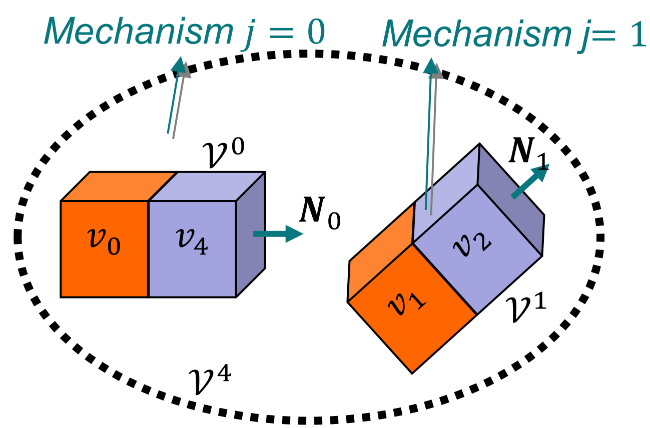
$$\mathbf{F}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{F}_i(t)$$



  - Fluctuation filed $\boldsymbol{u}'$

$$\mathbf{F}_{\mathrm{m}}(\boldsymbol{X}_{\mathrm{m}}) = \mathbf{F}_{\mathrm{M}} + \boldsymbol{u}'(\boldsymbol{X}_{\mathrm{m}}) \otimes \nabla_0$$

$$\Rightarrow \mathbf{F}_i = \mathbf{F}_{\mathrm{M}} + \frac{1}{V_i} \sum_{(i,j)} s_j^i \, \boldsymbol{u}_j' \otimes \boldsymbol{N}_j^i \,, \quad i = 0..7$$

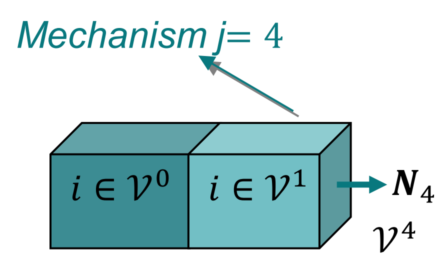


  - Weak form from Hill Mandel condition

$$\mathbf{P}_{\mathrm{M}} : \delta \mathbf{F}_{\mathrm{M}} = \sum_i v_i \, \mathbf{P}_i : \delta \mathbf{F}_i$$

$$\Rightarrow \left[\sum_j \left(\sum_i \frac{v_i s_j^i}{V_i} \mathbf{P}_i(\boldsymbol{u}_k') \right) \cdot \boldsymbol{N}_j^i \right] \cdot \delta \boldsymbol{u}_j' = 0$$

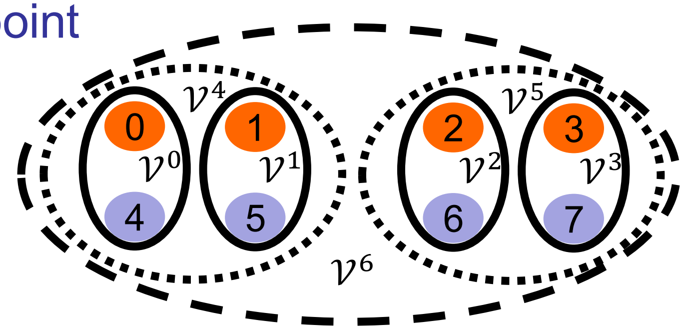
$$\Rightarrow \sum_i s_j^i \, \mathbf{P}_i(\boldsymbol{u}_k') \cdot \boldsymbol{N}_j^i = 0, \qquad j = 0, \ldots, 6$$



- Deep Material Networks from the interaction viewpoint
  - Strain/Stress averaging

$$\mathbf{P}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{P}_i(t)$$
$$\mathbf{F}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{F}_i(t)$$
$$v_i = \frac{V_i}{V_{\mathrm{M}}}$$

Weight of node i (parameter)



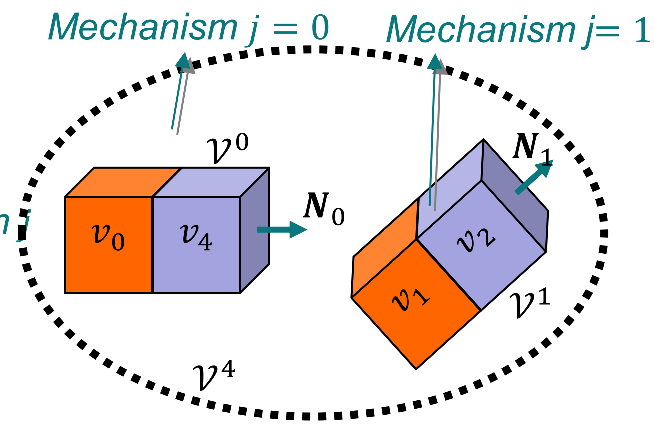
  - Fluctuation filed $\boldsymbol{u}'$

$$\mathbf{F}_{\mathrm{m}}(\boldsymbol{X}_{\mathrm{m}}) = \mathbf{F}_{\mathrm{M}} + \boldsymbol{u}'(\boldsymbol{X}_{\mathrm{m}}) \otimes \nabla_0$$

$$\mathbf{F}_i = \mathbf{F}_{\mathrm{M}} + \frac{1}{V_i} \sum_{(i,j)} s_j^i \, \boldsymbol{u}_j' \otimes \boldsymbol{N}_j^i \;, \quad i = 0..7$$

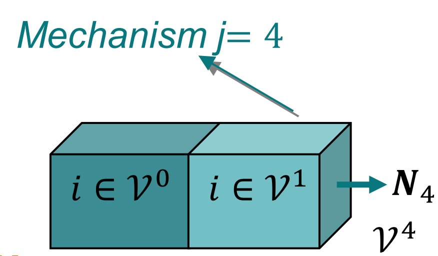
Contribution of node i in mechanism j (parameter)

Direction of mechanism j (parameter)



  - Offline stage: Parameters $(v_i, s_j^i, \boldsymbol{N}_j^i)$
    - Obtained by training ⇒ data driven approach
    - Should be independent from the constitutive behaviours
    - Should be independent from load paths

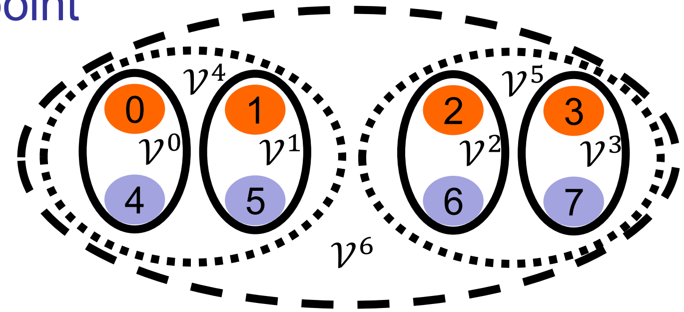
⇒ **Representative of the micro-structure organisation only**



AI-accelerated multi-scale analysis of composites

- Deep Material Networks from the interaction viewpoint
  - Strain/Stress averaging

$$\mathbf{P}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{P}_i(t)$$
$$\mathbf{F}_{\mathrm{M}}(t) = \sum_{i=0}^{7} v_i \, \mathbf{F}_i(t)$$
$$v_i = \frac{V_i}{V_{\mathrm{M}}}$$

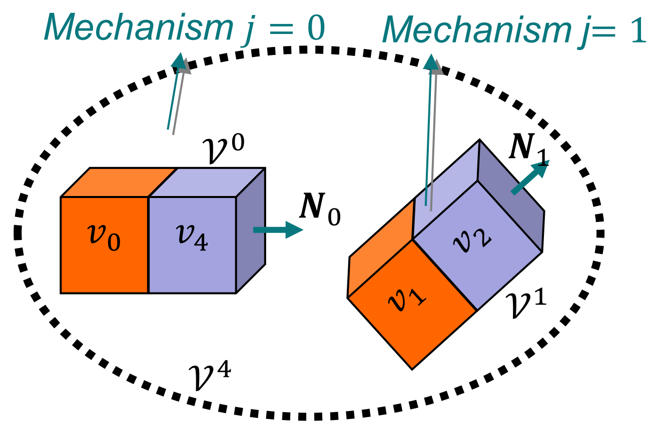


  - Fluctuation filed $\boldsymbol{u}'$

$$\mathbf{F}_{\mathrm{m}}(\boldsymbol{X}_{\mathrm{m}}) = \mathbf{F}_{\mathrm{M}} + \boldsymbol{u}'(\boldsymbol{X}_{\mathrm{m}}) \otimes \nabla_0$$

$$\mathbf{F}_i = \mathbf{F}_{\mathrm{M}} + \frac{1}{V_i} \sum_{(i,j)} s_j^i \, \boldsymbol{u}_j' \otimes \boldsymbol{N}_j^i \,, \quad i = 0..7$$

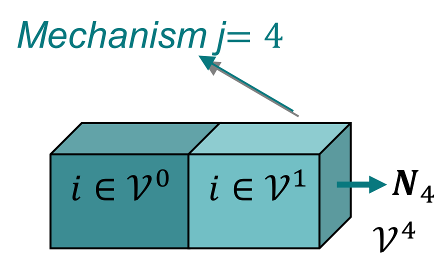
Degrees of freedom of mechanism j defining the strain fluctuation



  - Online stage: Weak form from Hill Mandel condition

$$\sum_i s_j^i \, \mathbf{P}_i(\boldsymbol{u}_k') \cdot \boldsymbol{N}_j^i = 0, \qquad j = 0, \ldots, 6$$

with $\mathbf{P}_i = \mathbf{P}^p(\mathbf{F}_i(\boldsymbol{u}'_k);\ \boldsymbol{Z}_i)$, for $i = 0, \ldots, 7$



⇒ Non-linear problem for given loading/material models

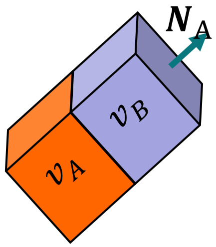
- Training of Interaction-Based Deep Material Networks
  - Summary

$$\begin{cases} \mathbf{F}_i = \mathbf{F}_\mathrm{M} + \frac{1}{V_i}\sum_{(i,j)} s_j^i\, \boldsymbol{u}'_j \otimes \boldsymbol{N}_j^i\;, \quad i = 0..7 \\ \sum_i s_j^i\, \mathbf{P}_i\,(\boldsymbol{u}'_k) \cdot \boldsymbol{N}_j^i = 0, \qquad j = 0,\ldots,6 \\ \mathbf{P}_i = \mathbf{P}^p(\mathbf{F}_i(\boldsymbol{u}'_k);\; \boldsymbol{Z}_i)\;, \text{ for } i = 0,\ldots,7 \\ \mathbf{P}_\mathbf{M}(t) = \sum_{i=0}^{7} v_i\, \mathbf{P}_i(t) \end{cases}$$

- Training of Interaction-Based Deep Material Networks
  - Summary

$$\begin{cases} \mathbf{F}_i = \mathbf{F}_\mathrm{M} + \frac{1}{V_i}\sum_{(i,j)} s_j^i\, \boldsymbol{u}'_j \otimes \boldsymbol{N}_j^i \;, \quad i = 0..7 \\ \sum_i s_j^i\, \mathbf{P}_i\,(\boldsymbol{u}'_k) \cdot \boldsymbol{N}_j^i = 0, \qquad j = 0,\dots,6 \\ \mathbf{P}_i = \mathbf{P}^p(\mathbf{F}_i(\boldsymbol{u}'_k);\; \boldsymbol{Z}_i) \text{ , for } i = 0, \dots, 7 \\ \mathbf{P}_\mathbf{M}(t) = \sum_{i=0}^{7} v_i\, \mathbf{P}_i(t) \end{cases}$$

  - Elasticity with only two pseudo-grains (1-level)
    - Analytical solution



New dof: $\hat{\mathrm{a}} = \frac{s_\mathrm{AB}}{V_\mathrm{M}}\boldsymbol{u}'$

$$\begin{cases} \boldsymbol{\varepsilon}_\mathrm{A} = \boldsymbol{\varepsilon}_\mathrm{M} + \frac{1}{v_\mathrm{A}}\frac{s_\mathrm{AB}}{V_\mathrm{M}}\boldsymbol{u}' \otimes^s \boldsymbol{N}_\mathrm{A} \text{ and} \\ \boldsymbol{\varepsilon}_\mathrm{B} = \boldsymbol{\varepsilon}_\mathrm{M} - \frac{1}{v_\mathrm{B}}\frac{s_\mathrm{AB}}{V_\mathrm{M}}\boldsymbol{u}' \otimes^s \boldsymbol{N}_\mathrm{A} \\ \mathbb{C}_\mathrm{M} = v_\mathrm{A}\mathbb{C}_\mathrm{A} + (1.0 - v_\mathrm{A})\mathbb{C}_\mathrm{B} + (\mathbb{C}_\mathrm{A} - \mathbb{C}_\mathrm{B}) \cdot \boldsymbol{N}_\mathrm{A} \otimes^s [\mathcal{K}^{-1}(v_\mathrm{A}, \boldsymbol{N}_\mathrm{A}) \cdot \mathcal{F}_\mathrm{M}(v_\mathrm{A}, \boldsymbol{N}_\mathrm{A})] \end{cases}$$

Topology parameters to be defined $\mathcal{G}^2 = \{v_\mathrm{A}, \boldsymbol{N}_\mathrm{A}\}$

$$\mathbb{C}_\mathrm{M} = \mathrm{FUN}\left(\mathbb{C}_\mathrm{A}, \mathbb{C}_\mathrm{B}, \mathcal{G}^2(v_\mathrm{A}, \boldsymbol{N}_\mathrm{A})\right)$$

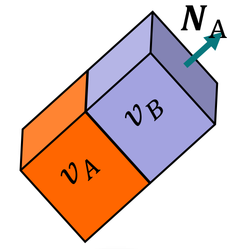
AI-accelerated multi-scale analysis of composites

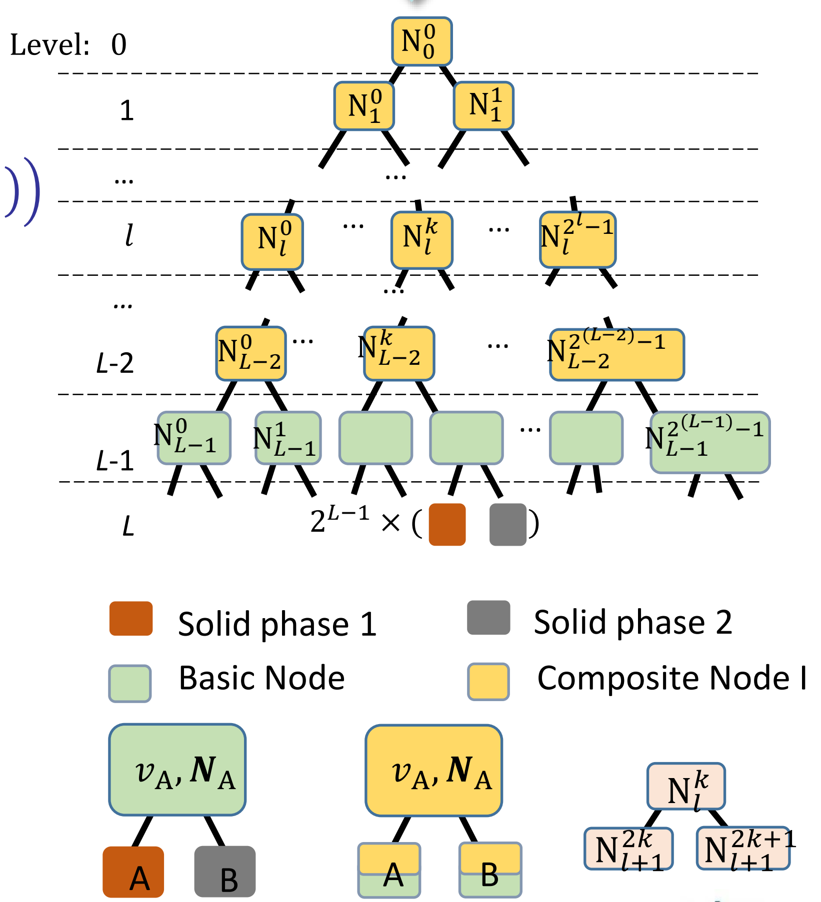
- Training of Interaction-Based Deep Material Networks
  - Elasticity with multi-level tree
    - Generalizing $\mathbb{C}_{\mathrm{M}} = \mathrm{FUN}\left(\mathbb{C}_{\mathrm{A}}, \mathbb{C}_{\mathrm{B}}, \mathcal{G}^2(v_{\mathrm{A}}, \boldsymbol{N}_{\mathrm{A}})\right)$

$$\mathbb{C}(\mathrm{N}_l^k) = \mathrm{FUN}\left(l, \mathbb{C}_0, \mathbb{C}_{\mathrm{I}}, \mathcal{G}^2(\mathrm{N}_l^k)\right)$$

with

$$\begin{cases} \mathbb{C}(\mathrm{N}_{l+1}^{2k}) = \mathrm{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_{\mathrm{I}}, \mathcal{G}^2(\mathrm{N}_{l+1}^{2k})\right) \\ \mathbb{C}(\mathrm{N}_{l+1}^{2k+1}) = \mathrm{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_{\mathrm{I}}, \mathcal{G}^2(\mathrm{N}_{l+1}^{2k+1})\right) \end{cases}$$





- Training of Interaction-Based Deep Material Networks
  - Elasticity with multi-level tree
    - Generalizing $\mathbb{C}_M = \text{FUN}\left(\mathbb{C}_A, \mathbb{C}_B, \mathcal{G}^2(v_A, \boldsymbol{N}_A)\right)$

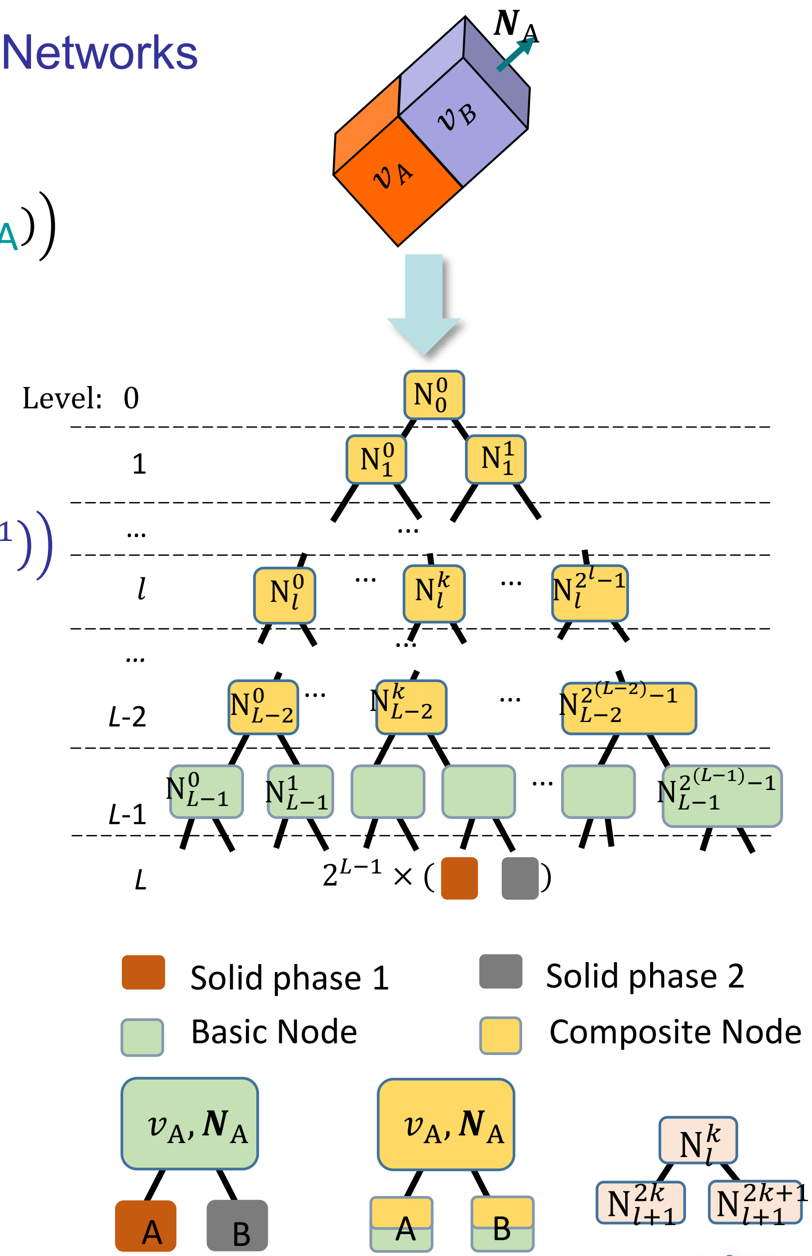
$$\mathbb{C}(\text{N}_l^k) = \text{FUN}\left(l, \mathbb{C}_0, \mathbb{C}_\text{I}, \mathcal{G}^2(\text{N}_l^k)\right)$$

with

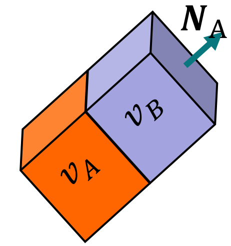
$$\begin{cases} \mathbb{C}(\text{N}_{l+1}^{2k}) = \text{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_\text{I}, \mathcal{G}^2(\text{N}_{l+1}^{2k})\right) \\ \mathbb{C}(\text{N}_{l+1}^{2k+1}) = \text{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_\text{I}, \mathcal{G}^2(\text{N}_{l+1}^{2k+1})\right) \end{cases}$$

- Recursive analytical solution

$$\mathbb{C}_M = \text{FUN}\left(l = 0, \mathbb{C}_0, \mathbb{C}_\text{I}, \mathcal{G}^2(\text{N}_0^0)\right)$$



- Training of Interaction-Based Deep Material Networks
  - Elasticity with multi-level tree



- Generalizing $\mathbb{C}_\mathrm{M} = \mathrm{FUN}\left(\mathbb{C}_\mathrm{A}, \mathbb{C}_\mathrm{B}, \mathcal{G}^2(v_\mathrm{A}, \boldsymbol{N}_\mathrm{A})\right)$

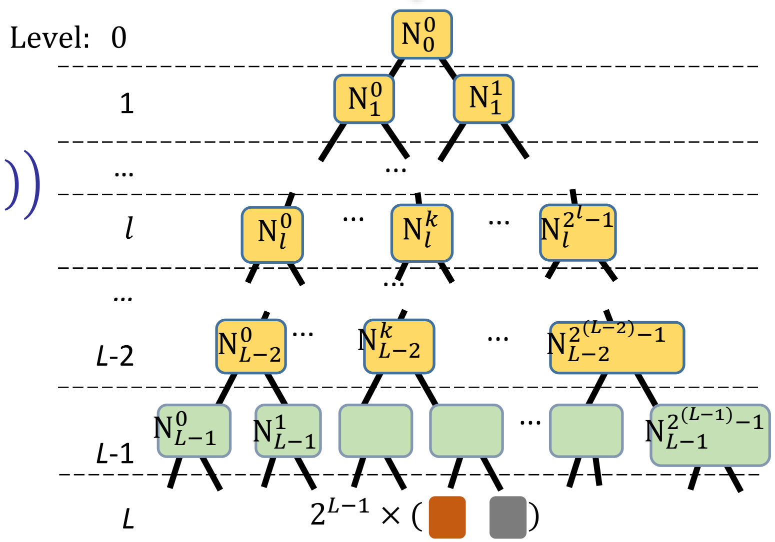
$$\mathbb{C}(\mathrm{N}_l^k) = \mathrm{FUN}\left(l, \mathbb{C}_0, \mathbb{C}_\mathrm{I}, \mathcal{G}^2(\mathrm{N}_l^k)\right)$$

with

$$\begin{cases} \mathbb{C}(\mathrm{N}_{l+1}^{2k}) = \mathrm{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_\mathrm{I}, \mathcal{G}^2(\mathrm{N}_{l+1}^{2k})\right) \\ \mathbb{C}(\mathrm{N}_{l+1}^{2k+1}) = \mathrm{FUN}\left(l+1, \mathbb{C}_0, \mathbb{C}_\mathrm{I}, \mathcal{G}^2(\mathrm{N}_{l+1}^{2k+1})\right) \end{cases}$$

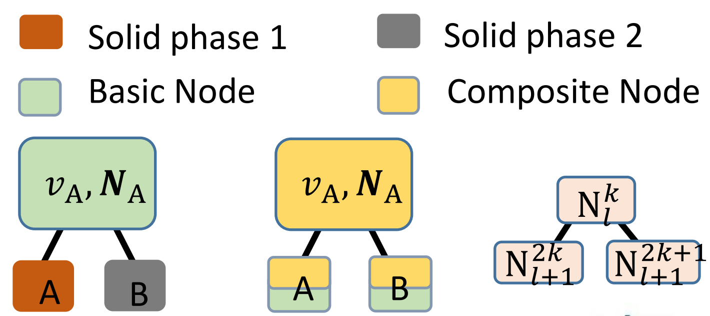
- Recursive analytical solution

$$\mathbb{C}_\mathrm{M} = \mathrm{FUN}\left(l=0, \mathbb{C}_0, \mathbb{C}_\mathrm{I}, \mathcal{G}^2(\mathrm{N}_0^0)\right)$$



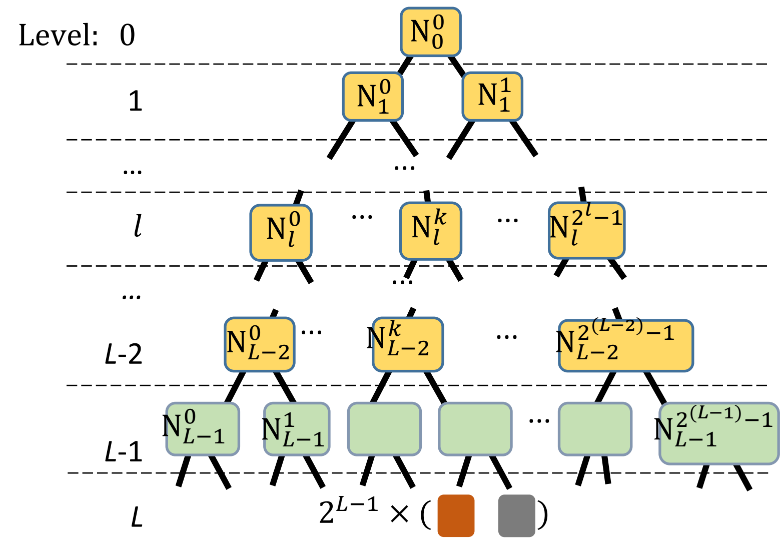
Topology parameters to be defined:

$\mathcal{G}^2(\mathrm{N}_0^0), \mathcal{G}^2(\mathrm{N}_1^0), \mathcal{G}^2(\mathrm{N}_1^1), \dots, \mathcal{G}^2\left(\mathrm{N}_{L-1}^{2^{(L-1)}-1}\right)$



- Training of Interaction-Based Deep Material Networks
  - Summary

$$
\begin{cases}
\mathbf{F}_i = \mathbf{F}_\mathrm{M} + \frac{1}{V_i}\sum_{(i,j)} s_j^i \, \boldsymbol{u}'_j \otimes \boldsymbol{N}_j^i \ , \quad i = 0..7 \\
\sum_i s_j^i \, \mathbf{P}_i\left(\boldsymbol{u}'_k\right) \cdot \boldsymbol{N}_j^i = 0, \qquad j = 0, \ldots, 6 \\
\mathbf{P}_i = \mathbf{P}^p\left(\mathbf{F}_i(\boldsymbol{u}'_k);\ \boldsymbol{Z}_i\right) \text{, for } i = 0, \ldots, 7 \\
\mathbf{P}_\mathrm{M}(t) = \sum_{i=0}^{7} v_i \, \mathbf{P}_i(t)
\end{cases}
$$



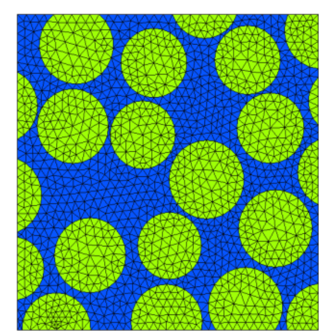
  - Analytical solution in linear elasticity
    - Recursively

$$\mathbb{C}_\mathrm{M} = \mathrm{FUN}\left(l = 0, \mathbb{C}_0, \mathbb{C}_\mathrm{I}, \mathcal{G}^2(\mathrm{N}_0^0)\right)$$

    - Topological parameters

$$\mathcal{G}^2(\mathrm{N}_0^0), \mathcal{G}^2(\mathrm{N}_1^0), \mathcal{G}^2(\mathrm{N}_1^1), \ldots, \mathcal{G}^2\left(\mathrm{N}_{L-1}^{2^{(L-1)}-1}\right)$$

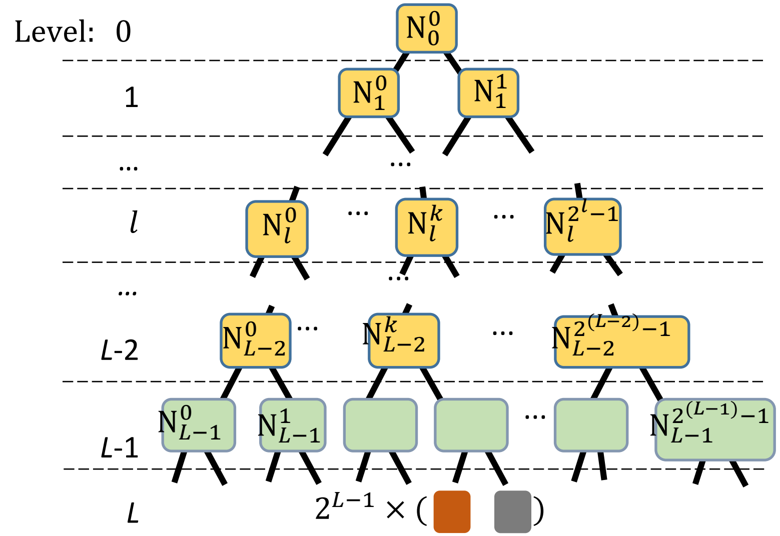
To be identified for a given micro-structure



AI-accelerated multi-scale analysis of composites

- Training of Interaction-Based Deep Material Networks
  - Summary

$$\begin{cases} \mathbf{F}_i = \mathbf{F}_\mathrm{M} + \frac{1}{V_i}\sum_{(i,j)} s_j^i\, \boldsymbol{u}'_j \otimes \boldsymbol{N}_j^i \;, \quad i = 0..7 \\ \sum_i s_j^i\, \mathbf{P}_i\,(\boldsymbol{u}'_k)\cdot \boldsymbol{N}_j^i = 0, \qquad j = 0,\dots,6 \\ \mathbf{P}_i = \mathbf{P}^p(\mathbf{F}_i(\boldsymbol{u}'_k);\; \boldsymbol{Z}_i)\;, \text{ for } i = 0, \dots, 7 \\ \mathbf{P}_\mathrm{M}(t) = \sum_{i=0}^{7} v_i\, \mathbf{P}_i(t) \end{cases}$$

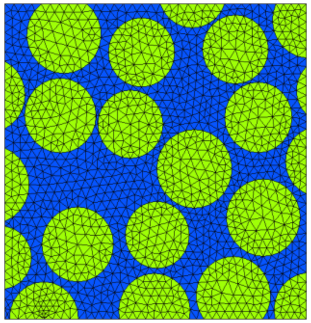


  - Analytical solution in linear elasticity
    - Recursively

$$\mathbb{C}_\mathrm{M} = \mathrm{FUN}\left(l = 0,\, \mathbb{C}_0,\, \mathbb{C}_\mathrm{I},\, \mathcal{G}^2(\mathrm{N}_0^0)\right)$$

    - Topological parameters

$$\mathcal{G}^2(\mathrm{N}_0^0), \mathcal{G}^2(\mathrm{N}_1^0), \mathcal{G}^2(\mathrm{N}_1^1), \dots, \mathcal{G}^2\left(\mathrm{N}_{L-1}^{2^{(L-1)}-1}\right)$$



  - Data driven approach:
    - Generate observations $\{\hat{\mathbb{C}}_\mathrm{M}(\mathbb{C}_0^s, \mathbb{C}_\mathrm{I}^s)\}$ by full DNS using random $\mathbb{C}_0^s$, $\mathbb{C}_\mathrm{I}^s$
    - Identify topological parameters from loss function (knowing real volume fraction $\hat{\mathrm{v}}_\mathrm{I}$ of phase I)

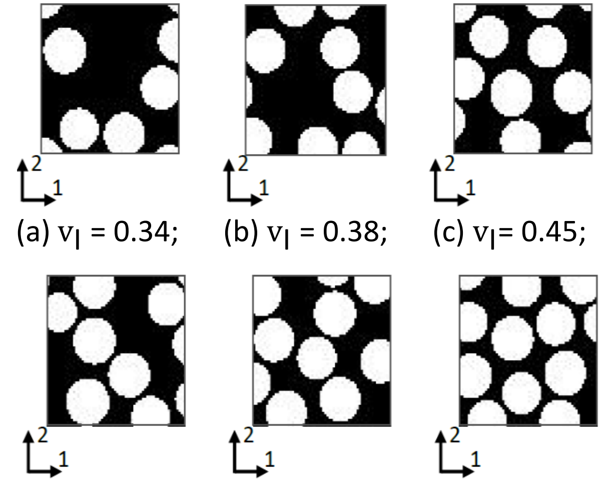
$$Loss(\hat{\mathbb{C}}_\mathrm{M}, \mathbb{C}_\mathrm{M}) = \frac{1}{n}\sum_{s=0}^{n-1} \frac{\|\hat{\mathbb{C}}_\mathrm{M}(\mathbb{C}_0^s, \mathbb{C}_\mathrm{I}^s) - \mathbb{C}_\mathrm{M}(\mathbb{C}_0^s, \mathbb{C}_\mathrm{I}^s; \mathcal{G}^2)\|}{\|\hat{\mathbb{C}}_\mathrm{M}(\mathbb{C}_0^s, \mathbb{C}_\mathrm{I}^s)\|} + \frac{\lambda}{2}(\hat{\mathrm{v}}_\mathrm{I} - \mathrm{v}_\mathrm{I})^2$$

AI-accelerated multi-scale analysis of composites

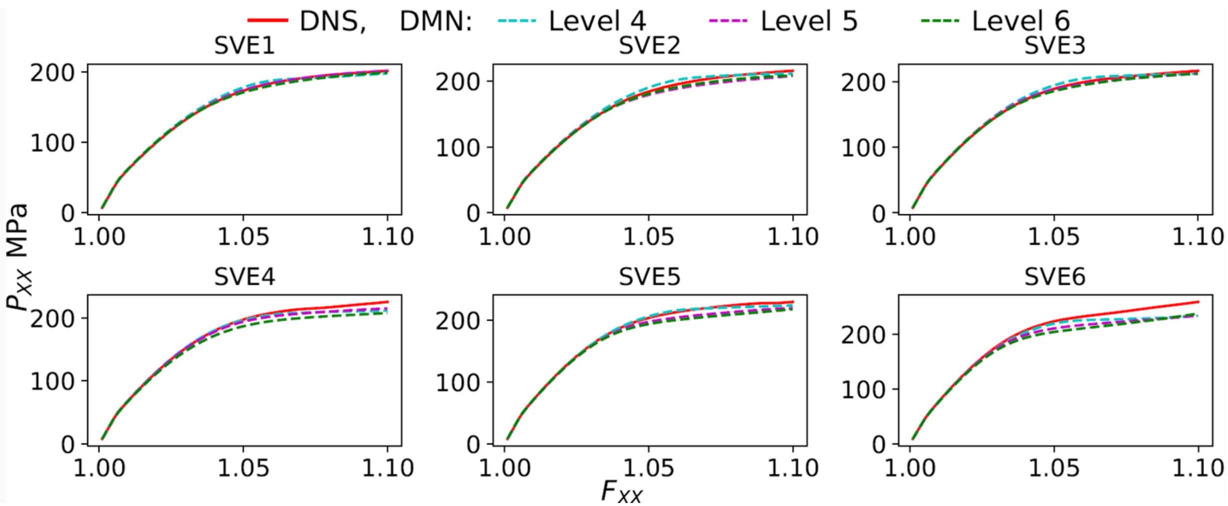
- Testing of Interaction-Based Deep Material Networks
  - Six micro-structure realizations
  - ➡ Six IB-DMNs to be trained in linear elasticity
    - For each micro-structure new set of topological parameters

$$\left\{ \mathcal{G}^2(\mathrm{N}_0^0), \mathcal{G}^2(\mathrm{N}_1^0), \mathcal{G}^2(\mathrm{N}_1^1), \ldots, \mathcal{G}^2\left(\mathrm{N}_{L-1}^{2^{(L-1)}-1}\right) \right\}$$

  - Testing on complex elasto-plasticity
    - Non-associated pressure-dependent plasticity
    - Extrapolation in terms of loading path
    - Extrapolation in terms of constitutive model



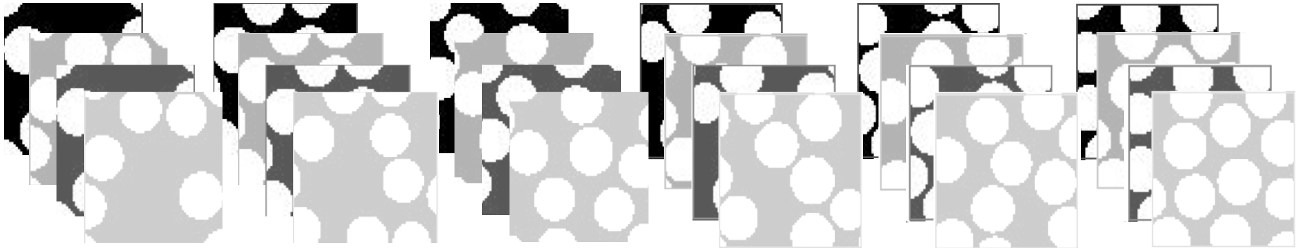
(a) $v_I = 0.34$; (b) $v_I = 0.38$; (c) $v_I = 0.45$; (d) $v_I = 0.47$; (e) $v_I = 0.52$; (f) $v_I = 0.58$



➡ For a new micro-structure: new training required

- Stochastic Interaction-Based Deep Material Networks
  - For a new micro-structure:
    - New training needed
  - When separation of scale not satisfied

$$l_{\mathrm{M}} >> l_{\mathrm{SVE}} \sim l_{\mathrm{m}}$$



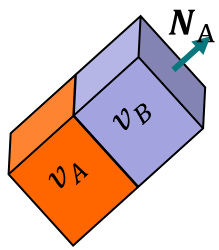
    - For the same composite volume fraction v_{I} the response depends on the micro-structure realisation

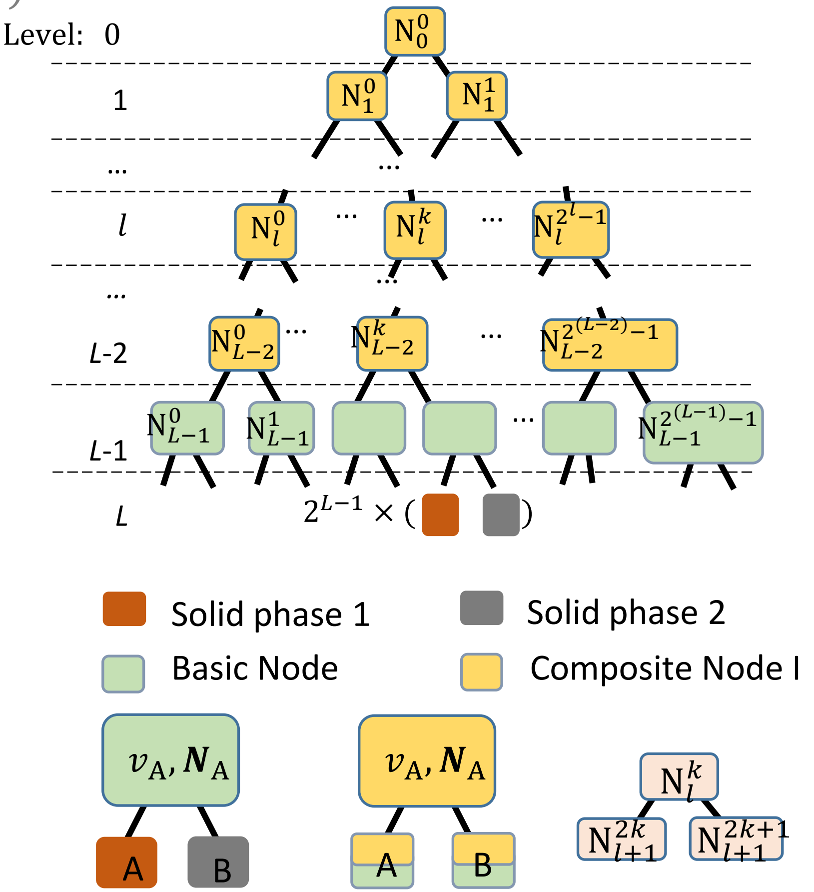
Can we interpolate the IB-DMN in terms of the micro-structure?

- Stage 1: arbitrary volume fraction v_l by reformulating $\mathcal{G}^2 = \{v_A, \boldsymbol{N}_A\}$ at node N_l^k
  - Normal $\boldsymbol{N}_A(N_l^k)$ from angles $\theta_1(N_l^k)$ and $\theta_2(N_l^k)$

$$\theta_1 = 2\pi w_{\theta_1}(N_l^k),\ \theta_2 = \pi w_{\theta_2}(N_l^k)$$

$$w_{\theta_1}(N_l^k), w_{\theta_2}(N_l^k) \in [0,\ 1)$$

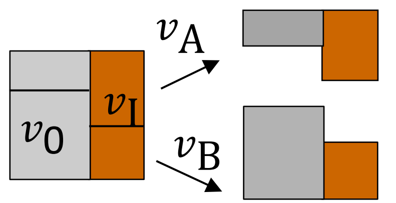




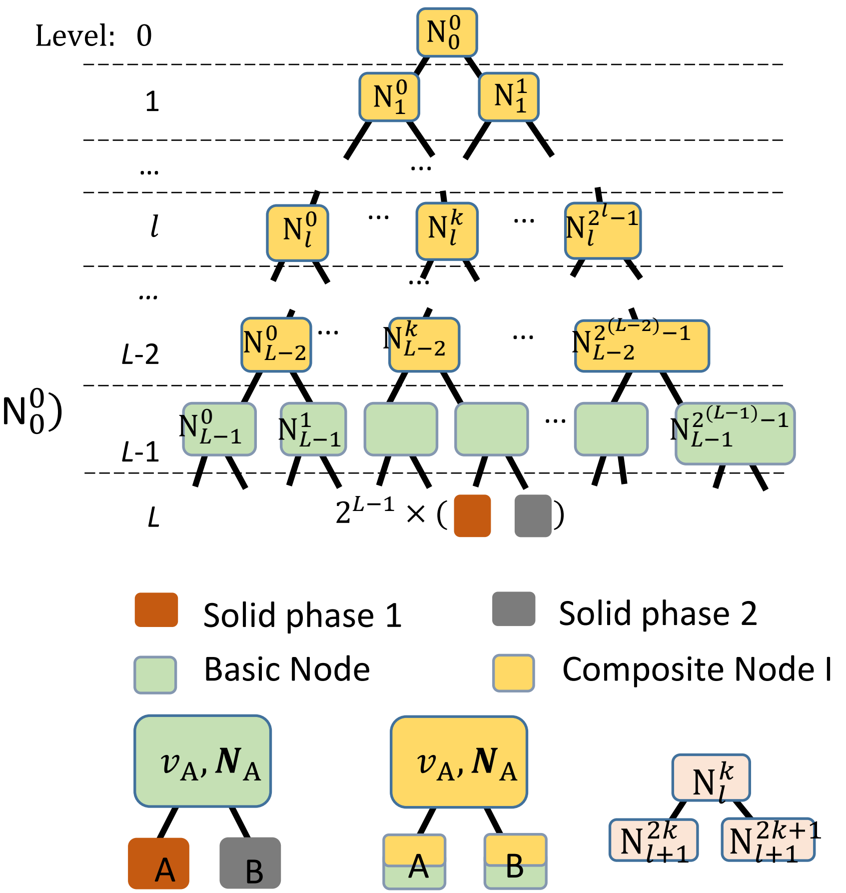
Redefinition of topological parameters to be defined:

$$\underline{w}_n = \{w_{\theta_1}, w_{\theta_2}\} = \text{sigmoid}(\mathcal{T}_n)$$

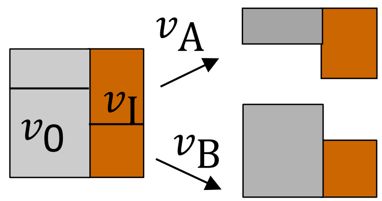
- Stage 1: arbitrary volume fraction v_I by reformulating $\mathcal{G}^2 = \{v_A, \boldsymbol{N}_A\}$ at node N_l^k
  - Volume fraction $v_A(N_l^k)$
    1) "Yellow" Node at level $l = 0$:
       $v_I(N_0^0) = v_I$ and $v_0(N_0^0) = 1 - v_I$ known



$$\begin{cases} v_A(N_0^0) = w_{v0}(N_0^0)\, v_0(N_0^0) + w_{vI}(N_0^0)\, v_I(N_0^0) \\ v_B(N_0^0) = 1.0 - v_A(N_0^0) \end{cases}$$



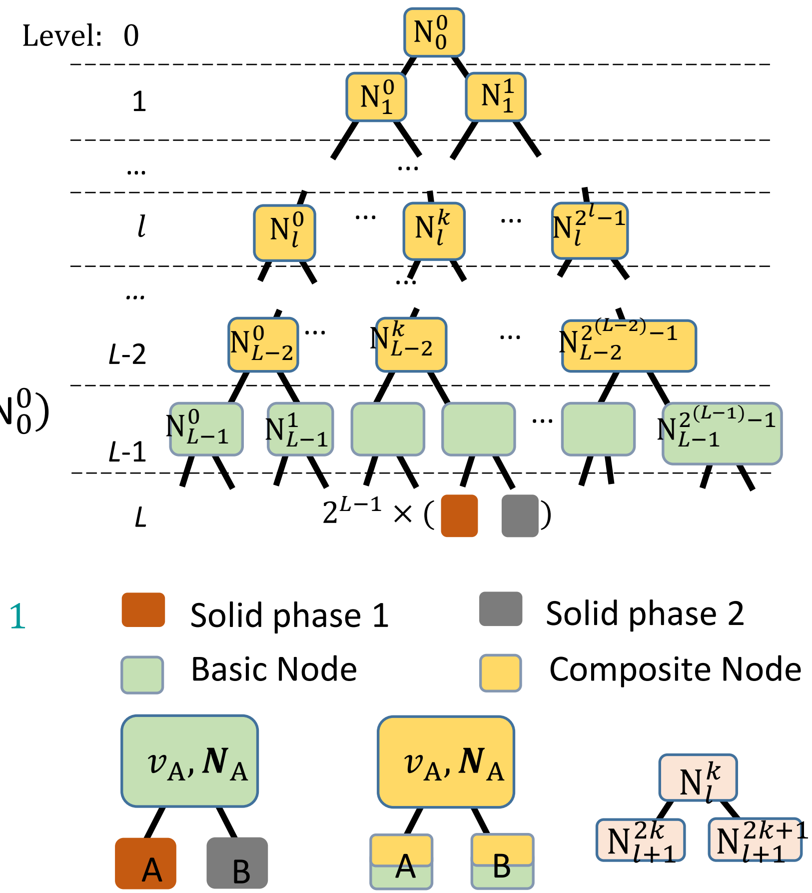
- Stage 1: arbitrary volume fraction v_I by reformulating $\mathcal{G}^2 = \{v_A, \boldsymbol{N}_A\}$ at node N_l^k
  - Volume fraction $v_A(N_l^k)$
    1) "Yellow" Node at level $l = 0$:
       $v_I(N_0^0) = v_I$ and $v_0(N_0^0) = 1 - v_I$ known



$$\begin{cases} v_A(N_0^0) = w_{v0}(N_0^0)v_0(N_0^0) + w_{vI}(N_0^0)v_I(N_0^0) \\ v_B(N_0^0) = 1.0 - v_A(N_0^0) \end{cases}$$

    2) Recursively for "Yellow" Node at level $0 < l < L - 1$

$$w_{v0}(N_l^k), w_{vI}(N_l^k) \in [0, 1)$$

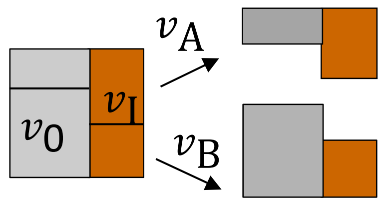


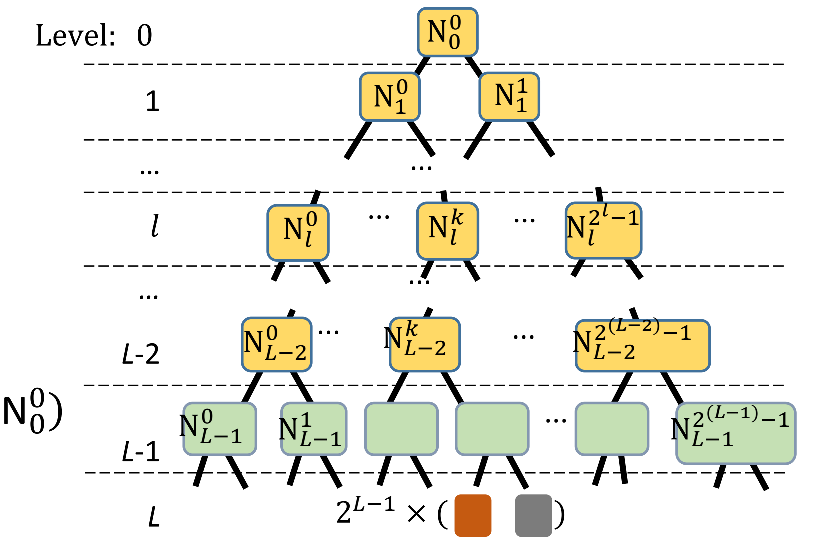
AI-accelerated stochastic multi-scale analysis of composites

- Stage 1: arbitrary volume fraction $\mathrm{v_I}$ by reformulating $\mathcal{G}^2 = \{v_\mathrm{A}, \boldsymbol{N}_\mathrm{A}\}$ at node N_l^k
  - Volume fraction $v_\mathrm{A}(\mathrm{N}_l^k)$

1) "Yellow" Node at level $l = 0$:

$v_\mathrm{I}(\mathrm{N}_0^0) = \mathrm{v_I}$ and $v_0(\mathrm{N}_0^0) = 1 - \mathrm{v_I}$ known

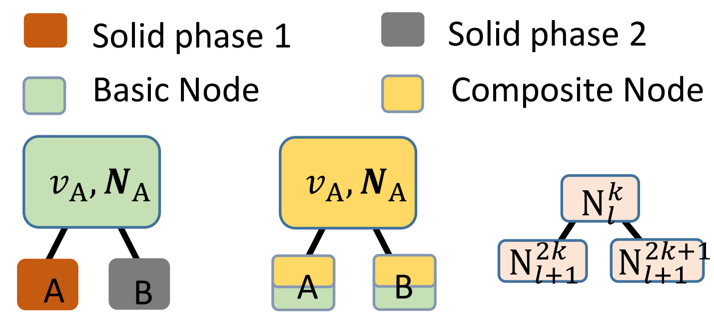




$$\begin{cases} v_\mathrm{A}(\mathrm{N}_0^0) = \mathrm{w}_{\mathrm{v}0}(\mathrm{N}_0^0) v_0(\mathrm{N}_0^0) + \mathrm{w}_{\mathrm{vI}}(\mathrm{N}_0^0) v_\mathrm{I}(\mathrm{N}_0^0) \\ v_\mathrm{B}(\mathrm{N}_0^0) = 1.0 - v_\mathrm{A}(\mathrm{N}_0^0) \end{cases}$$

2) Recursively for "Yellow" Node at level $0 < l < L-1$

$$\mathrm{w}_{\mathrm{v}0}(\mathrm{N}_l^k), \mathrm{w}_{\mathrm{vI}}(\mathrm{N}_l^k) \in [0, 1)$$



3) Basic "Green" Node at level $L - 1$:

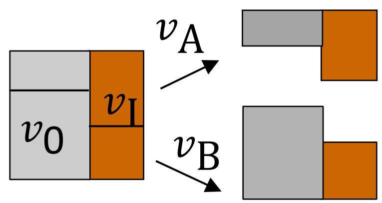
$$v_\mathrm{A} = 1.0 - v_\mathrm{I}(\mathrm{N}_{L-1}^k) \text{ and } v_\mathrm{B} = v_\mathrm{I}(\mathrm{N}_{L-1}^k)$$

AI-accelerated stochastic multi-scale analysis of composites

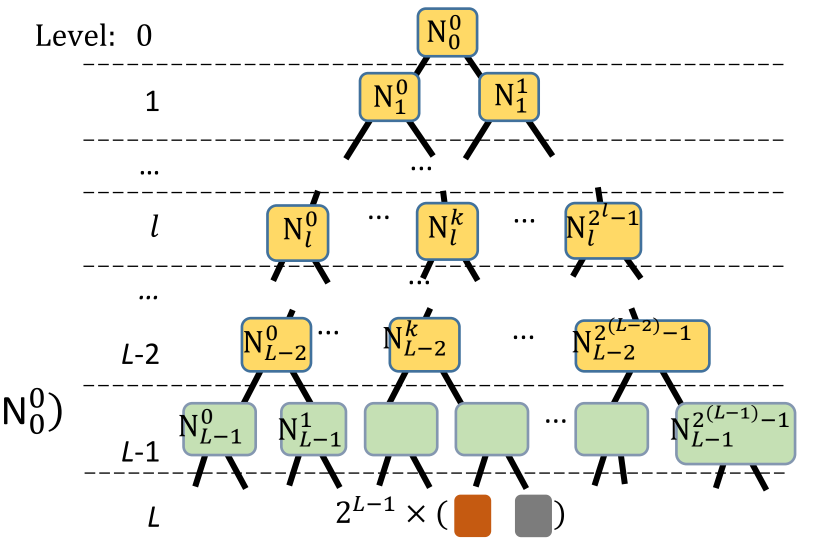
- Stage 1: arbitrary volume fraction v_I by reformulating $\mathcal{G}^2 = \{v_A, \boldsymbol{N}_A\}$ at node N_l^k
  - Volume fraction $v_A(N_l^k)$

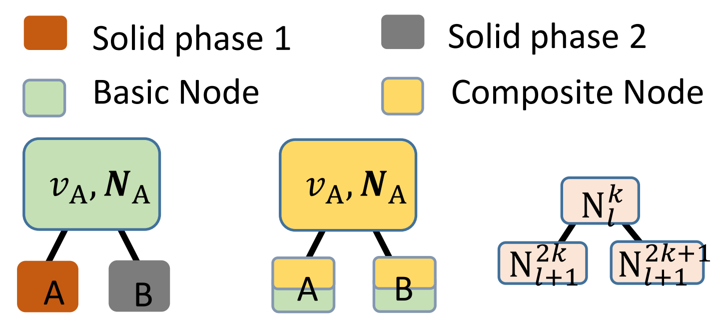
1) “Yellow” Node at level $l = 0$:

$v_I(N_0^0) = v_I$ and $v_0(N_0^0) = 1 - v_I$ known



$$\begin{cases} v_A(N_0^0) = w_{v0}(N_0^0)v_0(N_0^0) + w_{vI}(N_0^0)v_I(N_0^0) \\ v_B(N_0^0) = 1.0 - v_A(N_0^0) \end{cases}$$





2) Recursively for “Yellow” Node at level $0 < l < L - 1$

$$w_{v0}(N_l^k), w_{vI}(N_l^k) \in [0, 1)$$

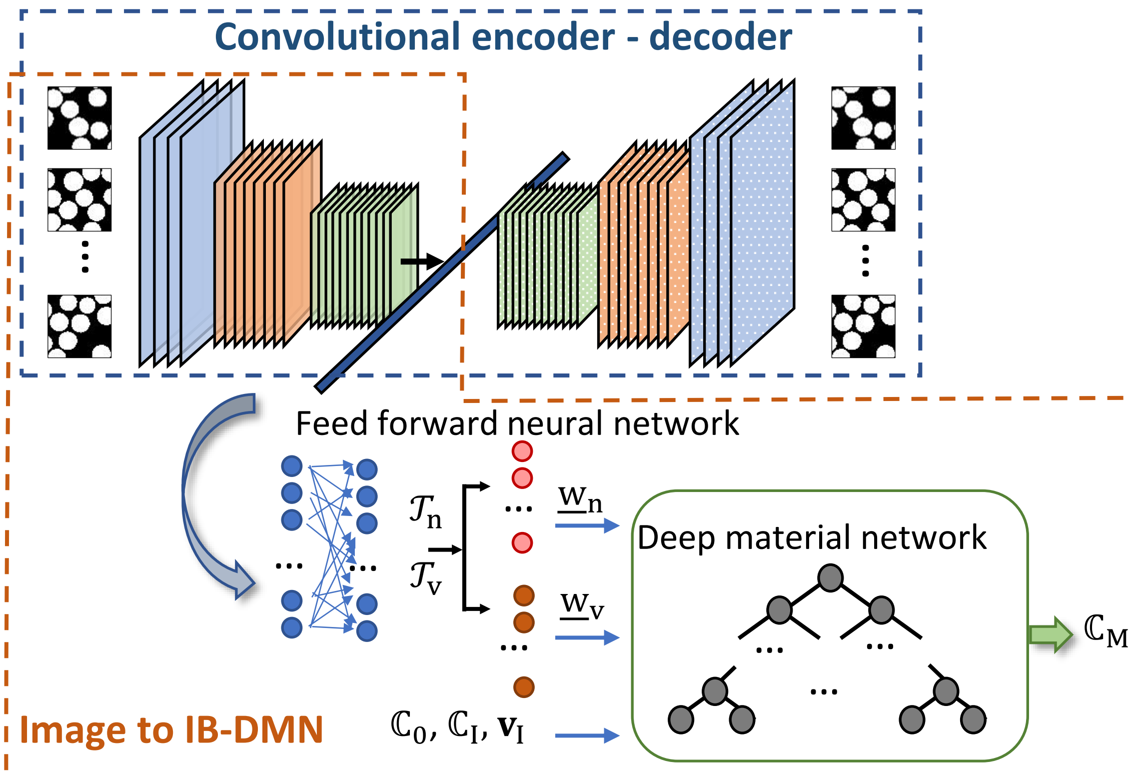
3) Basic “Green” Node at level $L - 1$:

$v_A = 1.0 - v_I(N_{L-1}^k)$ and $v_B = v_I(N_{L-1}^k)$

Redefinition of topological parameters to be defined: $\underline{w}_v = \{w_{v0}, w_{v1}\} = \text{sigmoid}(\mathcal{J}_v)$

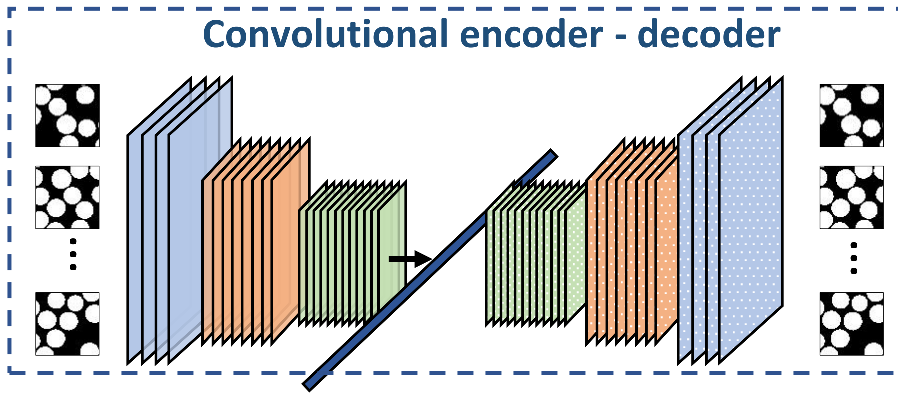
AI-accelerated stochastic multi-scale analysis of composites

- Stage 2: Predict topological parameters $\underline{w}_n = \text{sigmoid}(\mathcal{T}_n)$ & $\underline{w}_v = \text{sigmoid}(\mathcal{T}_v)$
  - Volume fraction v_I already extracted
  - $\mathcal{T}_n$ & $\mathcal{T}_v$ represent the micro-structure organisation → to be predicted from images
    - Trained Convolutional encoder-decoder: encoder predicts a features vector from image
    - Trained Feed-forward neural network: predicts topological parameters from features vector

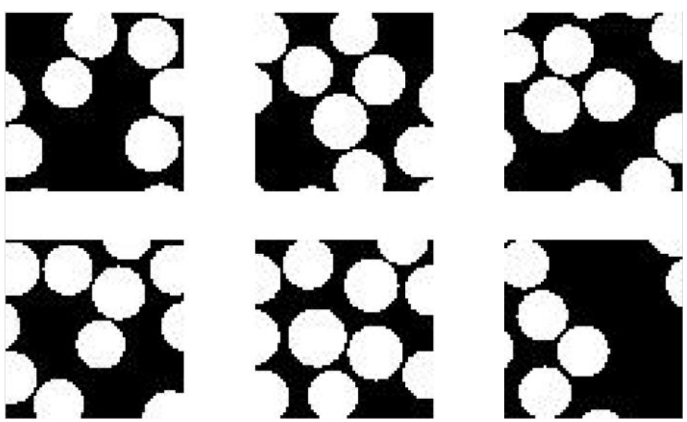


AI-accelerated stochastic multi-scale analysis of composites

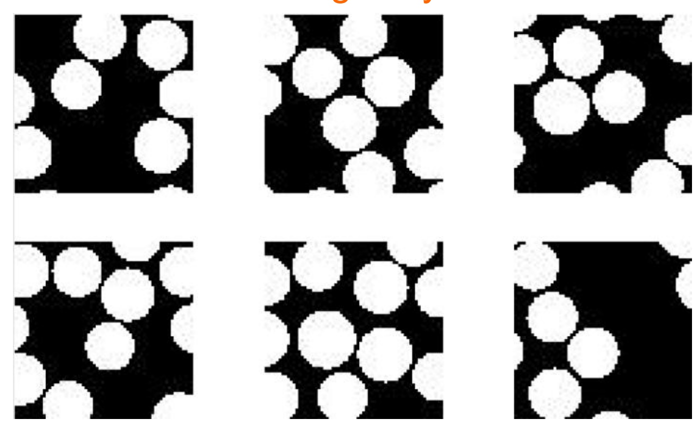
- Stage 2: Predict topological parameters $\underline{w}_n = \mathrm{sigmoid}(\mathcal{J}_n) \;\&\; \underline{w}_v = \mathrm{sigmoid}(\mathcal{J}_v)$
  - Convolutional encoder-decoder
    - Inputs: Reference images from SEM analyses
    - Output: reconstructed image
    - Trained with 10,000 reference images



Reference testing images



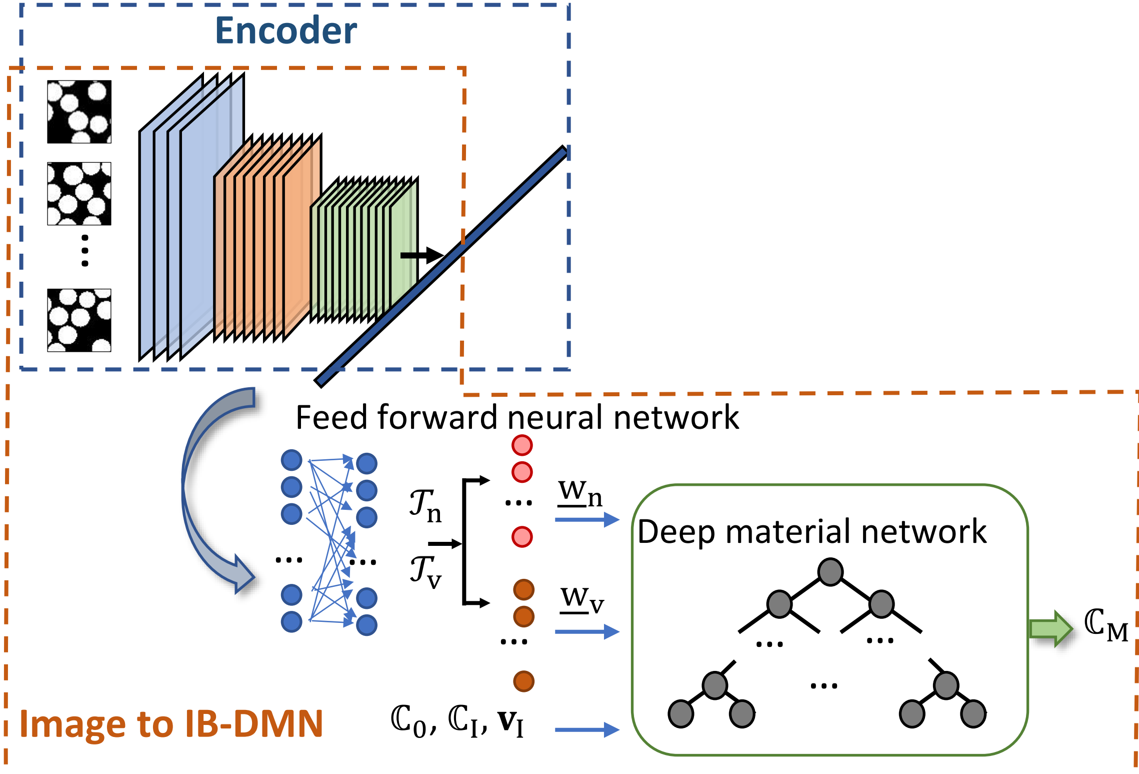
Predicted images by decoder

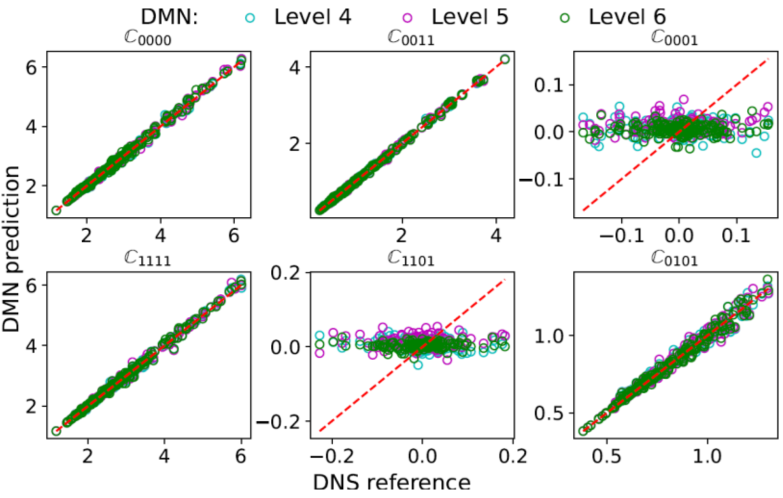


Once trained, the encoder gives a features vector of a micro-structure image

AI-accelerated stochastic multi-scale analysis of composites

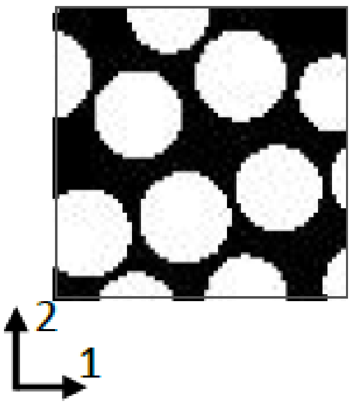
- Stage 2: Predict topological parameters $\underline{w_n} = \text{sigmoid}(\mathcal{T}_n)$ & $\underline{w_v} = \text{sigmoid}(\mathcal{T}_v)$
  - Feed forward neural network
    - Input: features vector of encoder
    - Output: topological parameters $\mathcal{T}_n$ & $\mathcal{T}_v$ representing the micro-structure
    - Trained so that the IB-DMN predicts the elasticity tensor of DNS

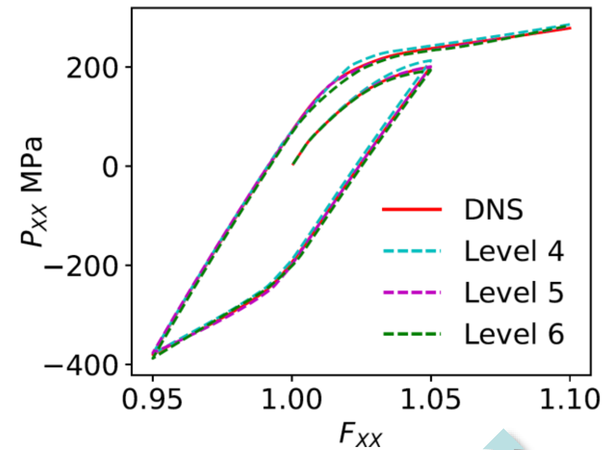


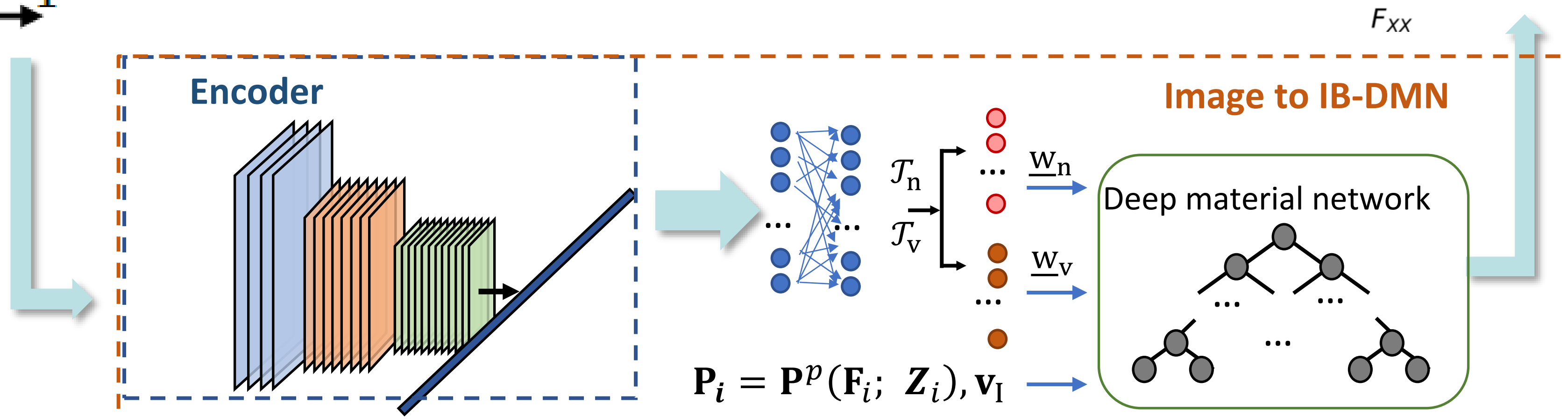


AI-accelerated stochastic multi-scale analysis of composites

- Stochastic IB-DMN prediction
  - New micro-structure realisation
  - Non-linear phase material model
  - Cyclic loading case







Conclusions & available materials

- Conclusions
  - Collection of powerful AI tools for many complex engineering applications
  - Right tool needs to be selected/developed for a given problem

- **Open positions (PhD students & post-doc)**

- Materials
  - Open source code : http://www.ltas-cm3.ulg.ac.be/openSource.htm
  - Publications http://www.ltas-cm3.ulg.ac.be/publications.htm
  - Data: http://www.ltas-cm3.ulg.ac.be/openSource.htm

- Credits

  - MOAMMM: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 862015 of the H2020-EU.1.2.1. - FET Open Programme
  - DIDEAROT: This project has received funding from the European Union’s Horizon Europe Framework Programme under grant agreement No. 101056682. The contents of this publication are the sole responsibility of ULiege and do not necessarily reflect the opinion of the European Union. Neither the European Union nor the granting authority can be held responsible for them.
  - CARBOBRAKE: This research has been funded by the Walloon Region under the agreement no. 2010092-CARBOBRAKE in the context of the M-ERA.Net Join Call 2020 funded by the European Union under the Grant Agreement no. 958174.