

Data-driven multi-scale simulations of composite material failure

L. Noels, L. Wu, C. Anglade
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

This project has received funding from the European Union's Horizon Europe Framework Programme under grant agreement No. 101056682 for the project "Digital DEsign strategies to certify and mAnufacture Robust cOMposite sTructures (DIDEAROT)". 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



- 2-Scale problem

- Idea:

- Substitute the explicit constitutive model

$$\sigma_M(X, t) = \sigma_M(\varepsilon_M(X, t); \mathbf{z}_M(X, \tau), \tau \in [0, t])$$

- By an implicit relation resulting from the lower-scale BVP resolution

- Macro-scale state variables correspond to the set of RVE state variables

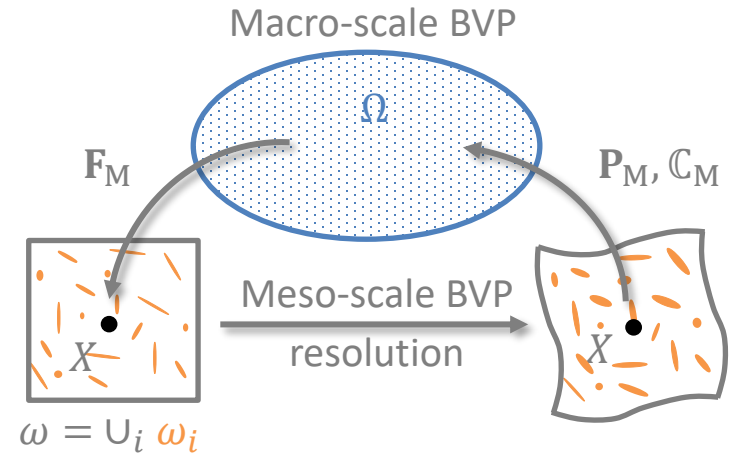
$$\mathbf{z}_M(X, t) = \{\mathbf{z}_m(\mathbf{x}, t) \forall \mathbf{x} \in \omega(X)\}$$

- Basic assumption: separation of scale

$$l_M \gg l_{RVE} \gg l_m$$

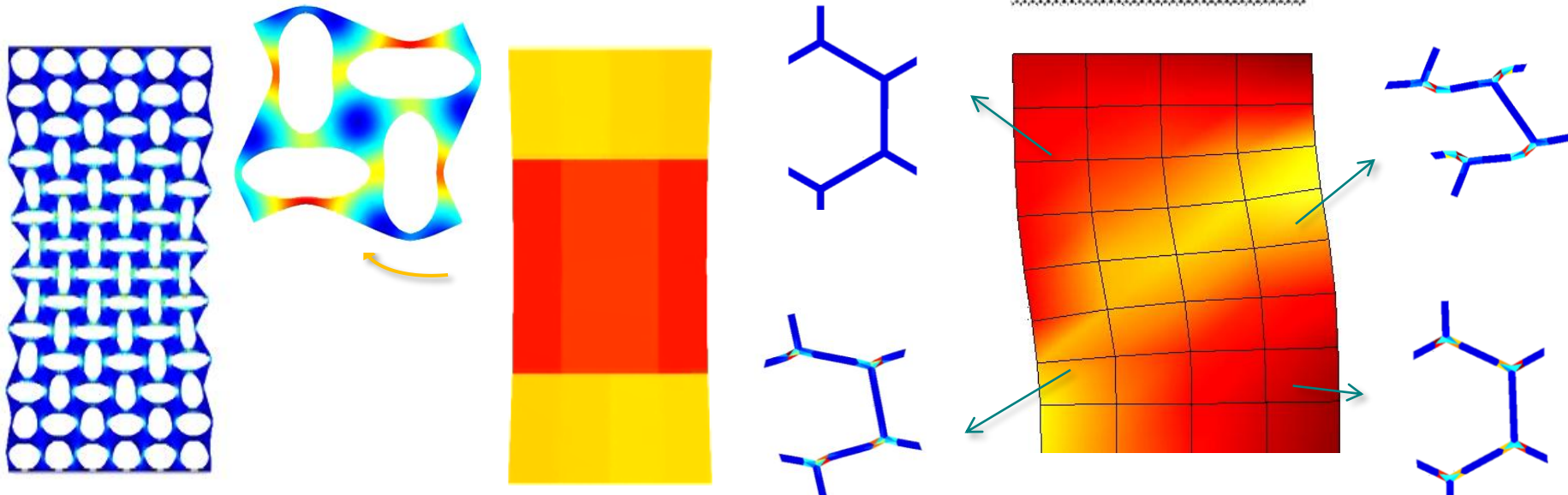
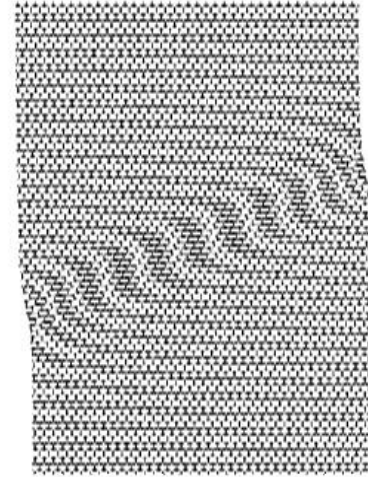
- Opportunities

- Complex simulations (accounting for complex failure modes)
 - Stochastic analyses
 - ...



- Multi-Scale FE² resolution
 - Method has been applied to a wide variety of problems
 - Several drawbacks
 - Overwhelming computation time
 - Fixed micro-structure

➔ Not manageable for real applications

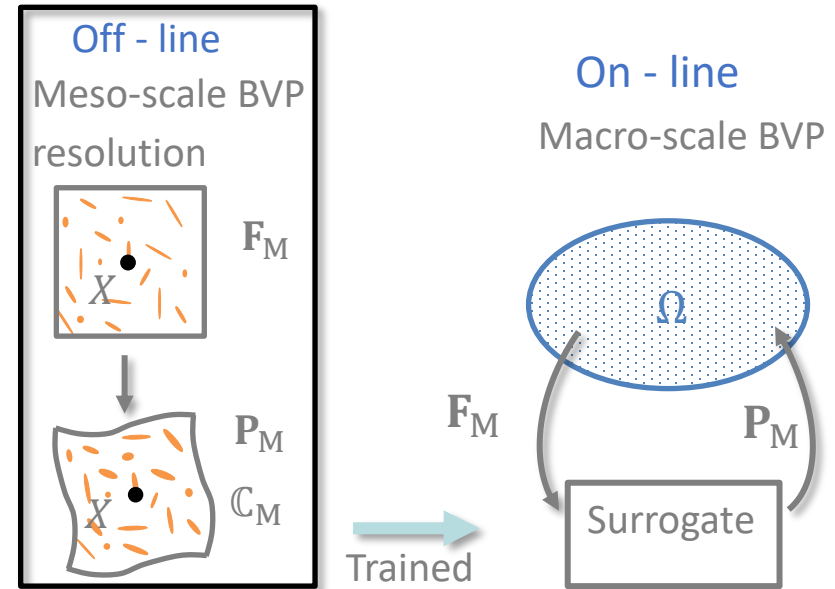


- Opportunities

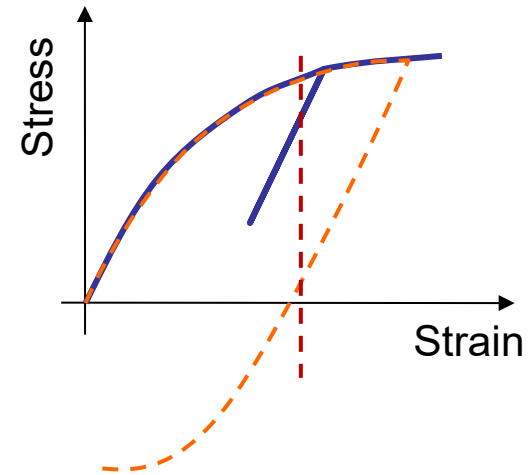
- Account for micro-structural effects on macro-scale responses
 - Failure origin
 - Micro-structural variation
- Accelerate simulations
 - Makes virtual testing possible

- Challenges

- Which surrogate/ML tool?
- How to get a relevant database?
- Quid extrapolating capabilities?
- How to account for fracture?

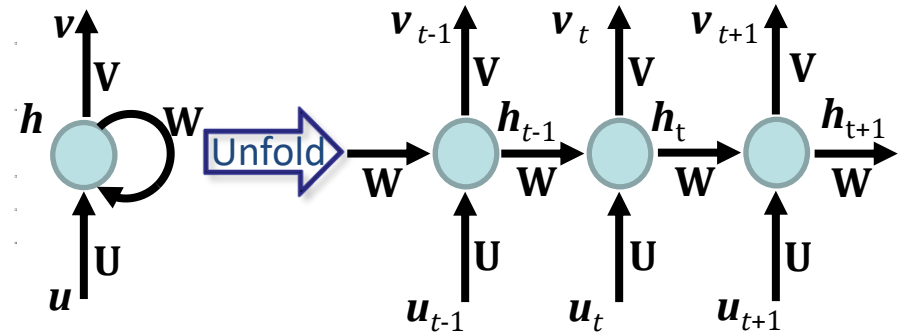
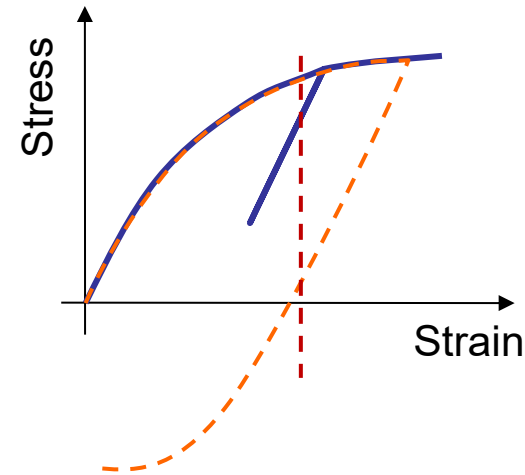


- Input / output definition
 - Input: Strain (history): $\mathbf{F}_M$
 - Output: Stress (history): $\mathbf{P}_M$
- Elasto-plastic material behaviour
 - No bijective strain-stress relation
 - Feed-forward NNW cannot be used
 - History should be accounted for

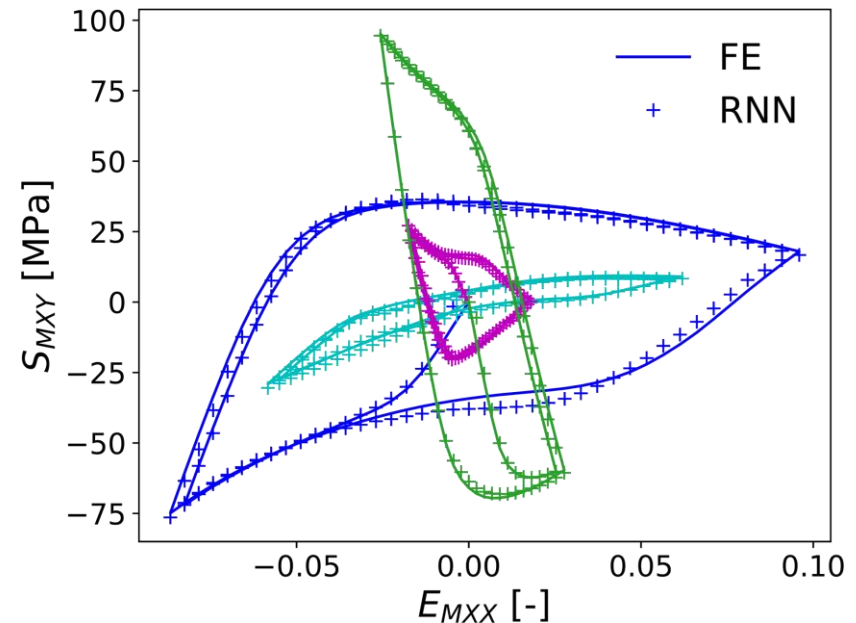
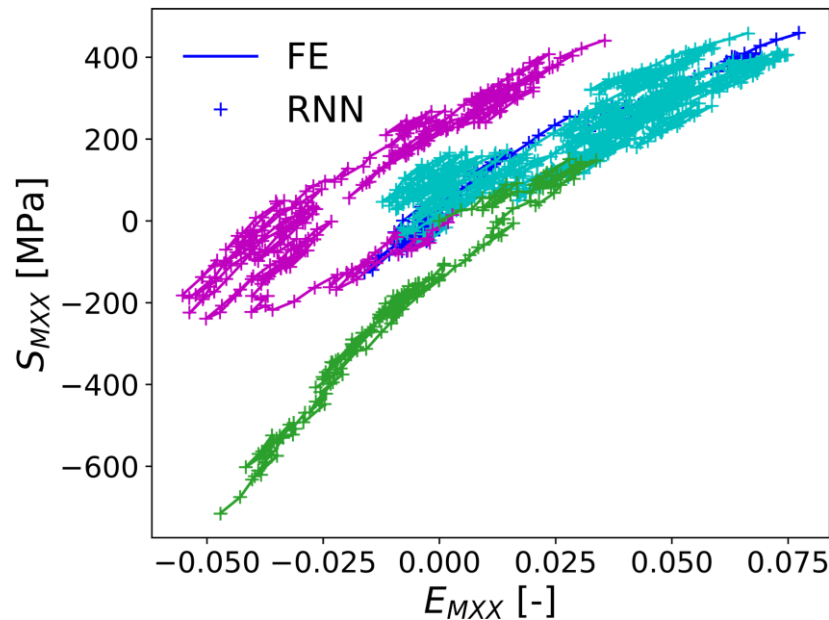
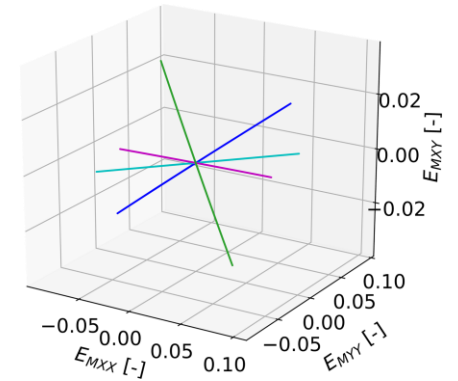
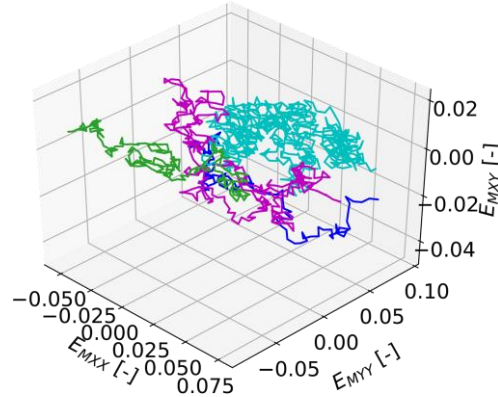
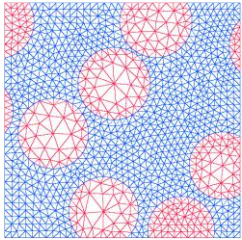


Recurrent Neural Network for multi-scale simulations

- Input / output definition
 - Input: Strain (history): $\mathbf{F}_M$
 - Output: Stress (history): $\mathbf{P}_M$
- 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 $\mathbf{u}_t$
 - Output: sequence $\mathbf{v}_t = \mathbf{g}(\mathbf{u}_t, \mathbf{h}_{t-1})$
 - Internal variable $\mathbf{h}_t = \mathbf{g}(\mathbf{u}_t, \mathbf{h}_{t-1})$
 - Existing recurrent units
 - GRU, LSTM
 - Both developed for Nature Language Processing
 - ...



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

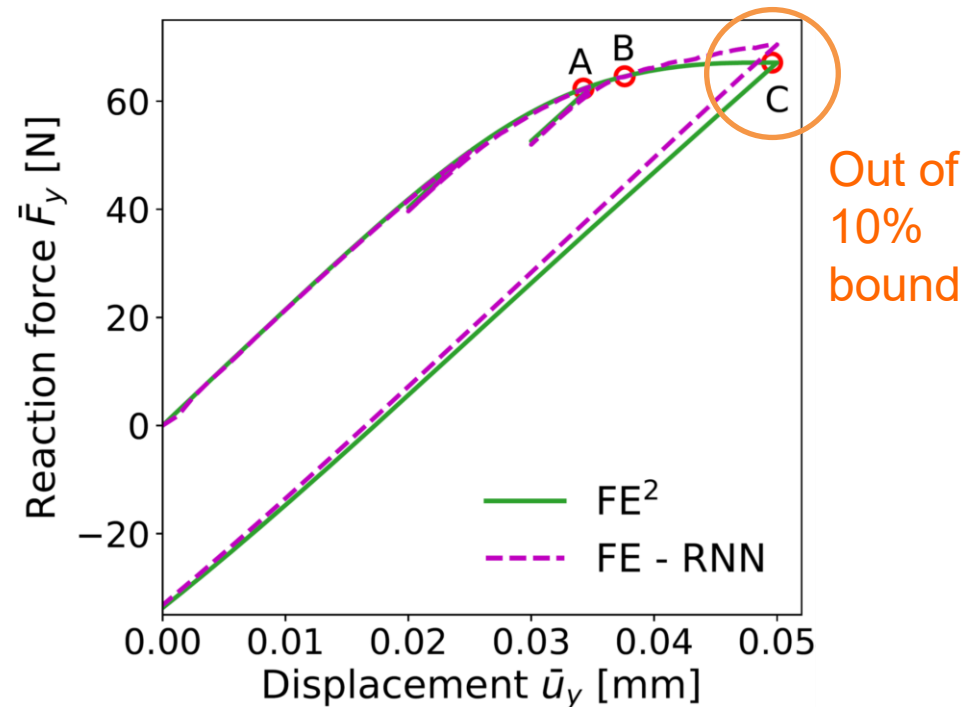
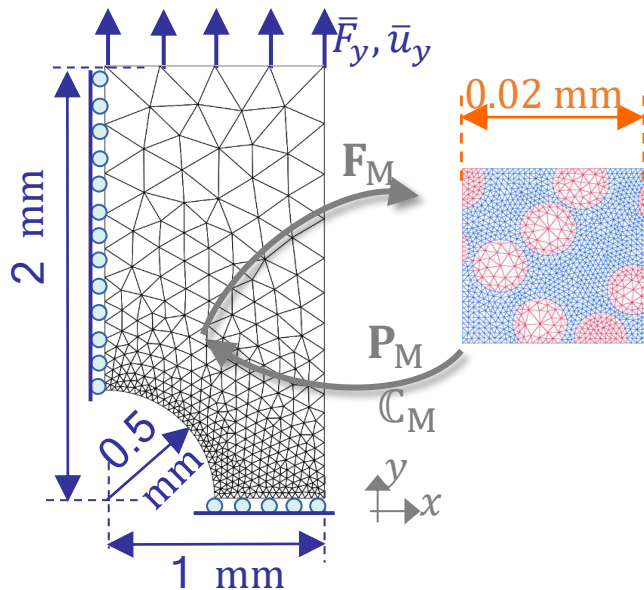


Recurrent Neural Network-accelerated multi-scale simulations

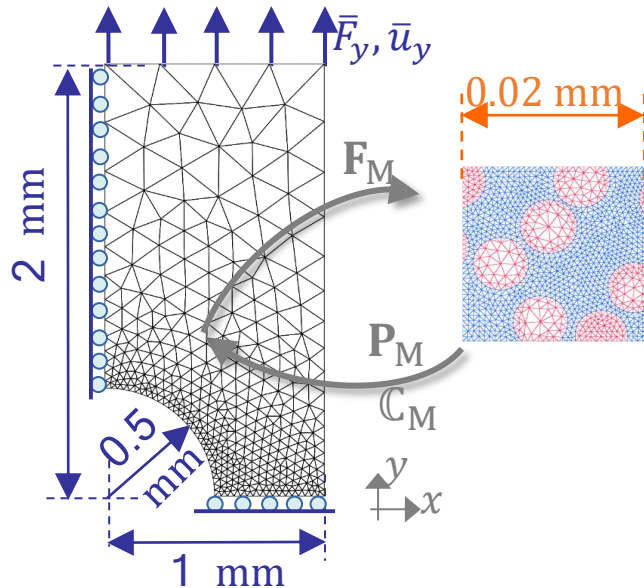
Multiscale simulation

- Use of a GRU as RNN
- Elasto-plastic composite RVE
- Comparison FE^2 vs. RNN-surrogate
- Training data
 - Bounded at 10% deformation

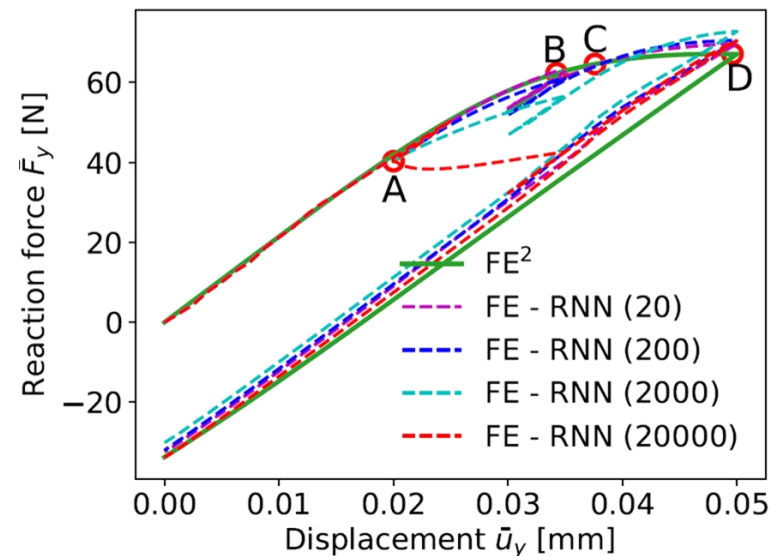
Off-line	FE^2	FE-RNN
Data generation	-	9000 x 2 h-cpu
Training	-	3 day-cpu
On-line	FE^2	FE-RNN
Simulation	18000 h-cpu	0.5 h-cpu



- 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



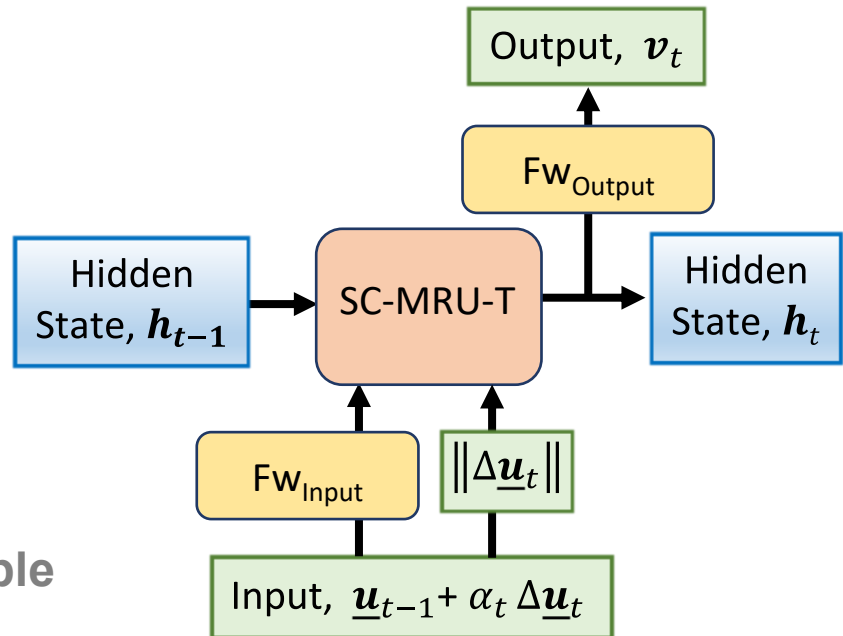
Simplified minimal recurrent unit



*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

- The Self-Consistent Recurrent Neural Network based on a SC unit
 - Oscillations / loss of accuracy can appear with GRU, LSTM* -> SC required



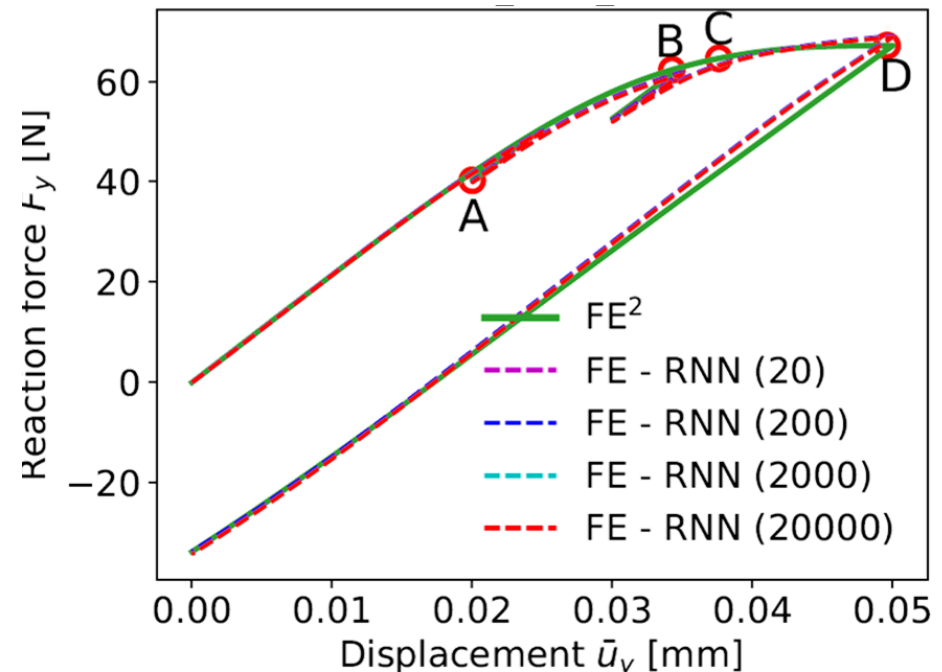
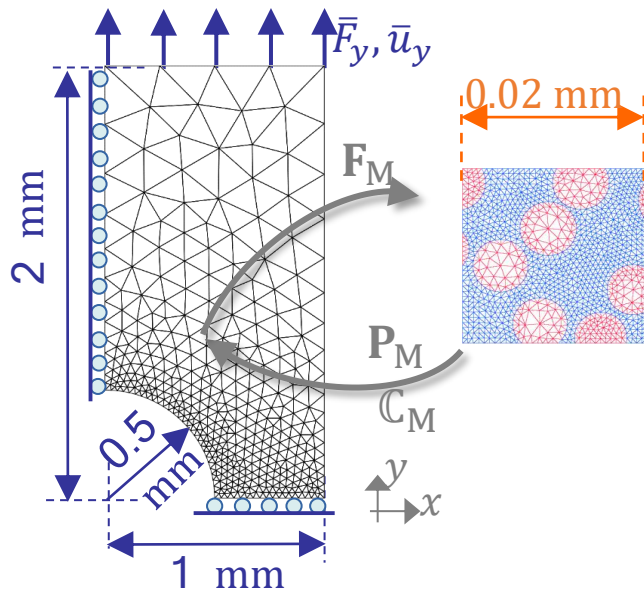
- The **total form of the input variable**
 - Use as input $\underline{u}_{t-1} + \alpha_t \Delta \underline{u}_t$
 - Lift the size of the input to the one of the hidden state with Fw_{input}
- & **increment norm** $\|\Delta \underline{u}_t\|$
 - For $\|\Delta \underline{u}_t\| \rightarrow 0$ the hidden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

*Colin Bonatti, Dirk Mohr, On the importance of self-consistency in recurrent neural network models representing elastoplastic 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

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

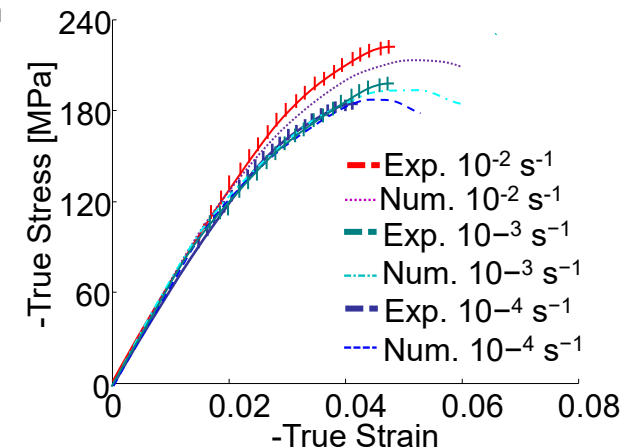
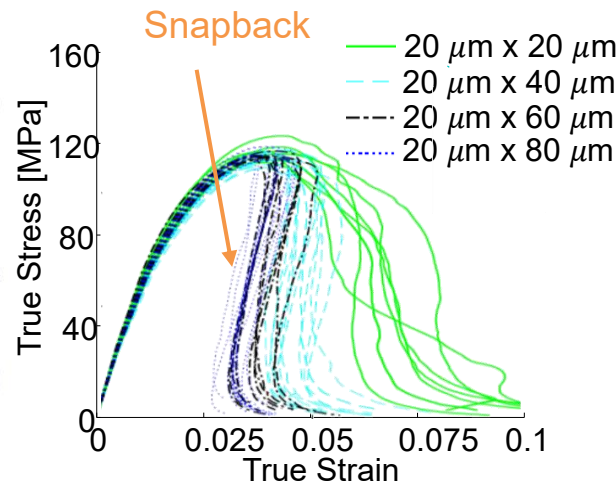
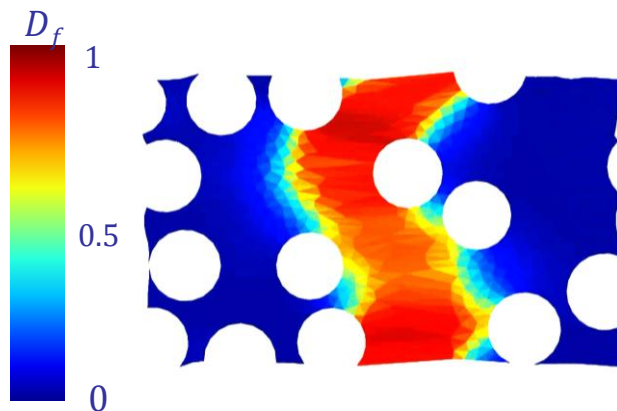
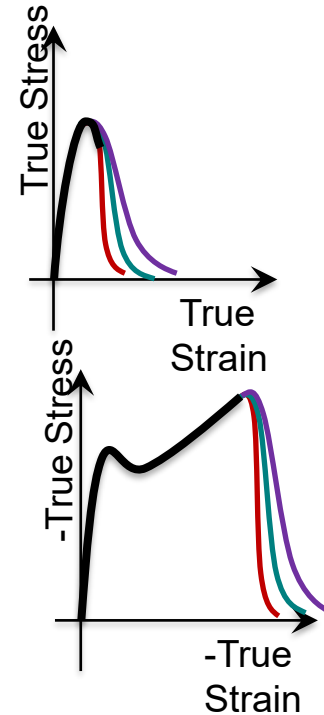
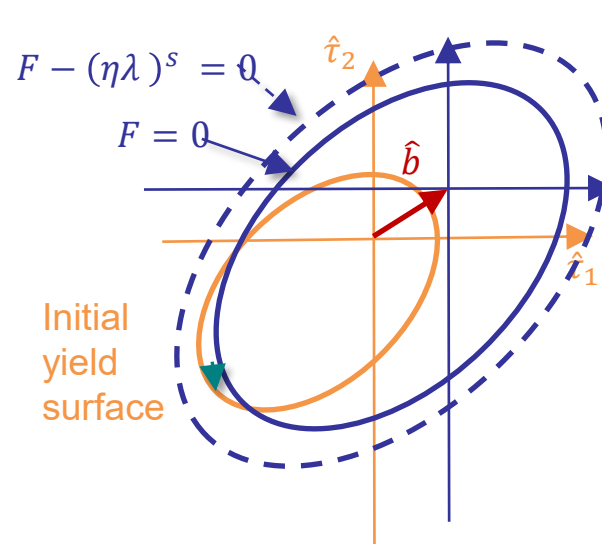
Off-line	FE ²	FE-GRU	FE-SC
Data generation	-	9000 x 2 h-cpu	23 500 h-cpu
Training	-	3 day-cpu	<10 h-cpu
On-line	FE ²	FE-GRU	FE-SC
Simulation	18000 h-cpu	0.5 h-cpu	0.4 h-cpu



AI-accelerated multi-scale analysis of composite failure

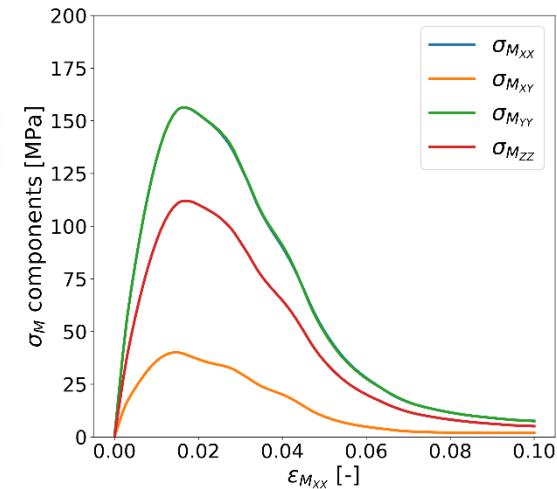
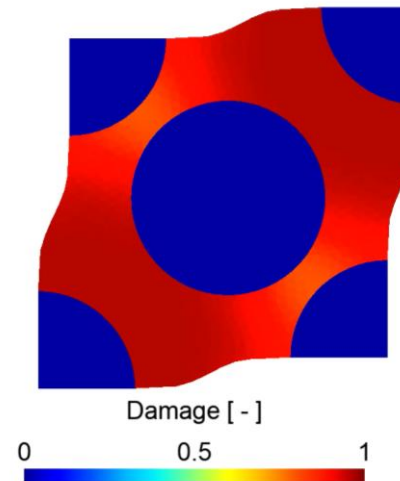
- RVE simulations:

- 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



- 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

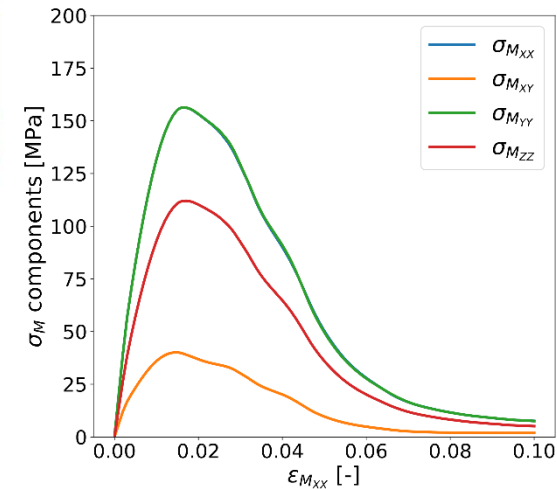
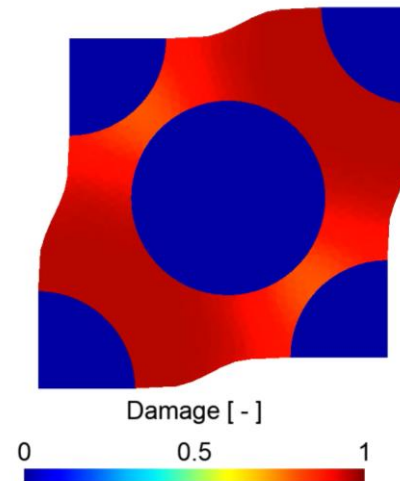
- Challenges to develop the AI surrogate:

- We need an extensive data-base

- Simulations should reach total failure

- Resin is brittle $\rightarrow$ snapback

$\rightarrow$ (R)VE size is reduced



- Homogenized curves are meaningless

- They exhibit a strain softening

- Softening depends on the RVE size

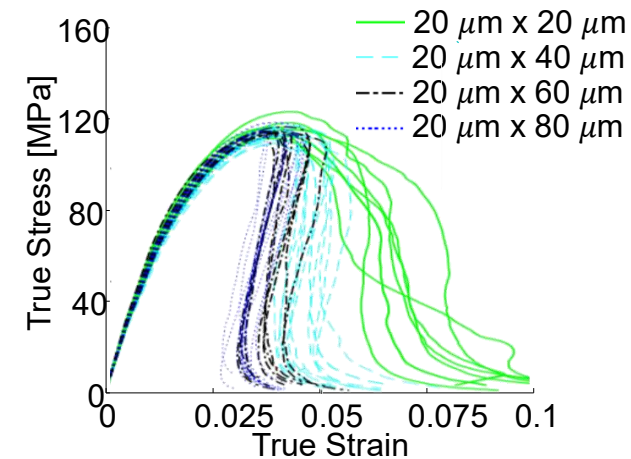
- Only physical value is fracture energy G_c

$\rightarrow$ 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)

$$\boldsymbol{\varepsilon}_M^{\text{plast}} = \boldsymbol{\varepsilon}_M - \mathbb{C}_M^{\text{el},D^{-1}} : \boldsymbol{\sigma}_M$$

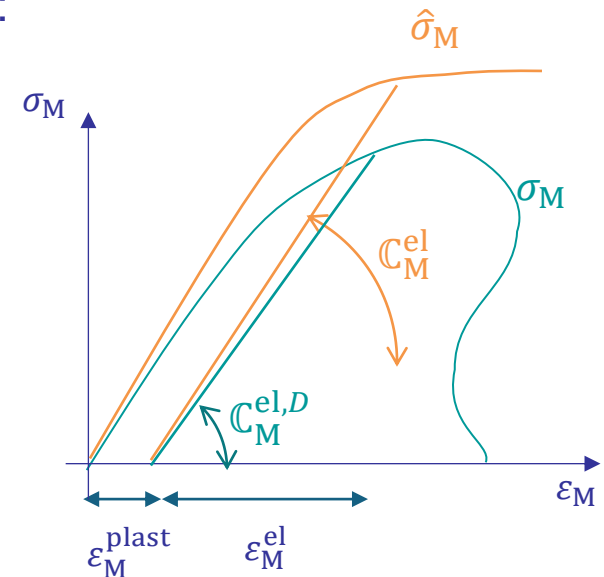
- Component-wise damage (to be reduced*)

$$D_{M_{ijkl}} = 1 - \frac{C_{M_{ijkl}}^{\text{el},D}}{C_{M_{ijkl}}^{\text{el}}}$$

- New macro-scale Helmholtz equations

$$D_{M_a} - l_d^2 \nabla^2 \tilde{D}_{M_a} = D_{M_a}$$

$$\tilde{\varepsilon}_{M_b}^{\text{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{M_b}^{\text{plast}} = \varepsilon_{M_b}^{\text{plast}}$$



*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

- Development of an embedding non-local framework:

- Identification of local/non-local variables

- Plastic strain tensor (from unloading in apparent space)

$$\boldsymbol{\varepsilon}_M^{\text{plast}} = \boldsymbol{\varepsilon}_M - \mathbb{C}_M^{\text{el},D^{-1}} : \boldsymbol{\sigma}_M$$

- Component-wise damage (to be reduced*)

$$D_{M_{ijkl}} = 1 - \frac{C_{M_{ijkl}}^{\text{el},D}}{C_{M_{ijkl}}^{\text{el}}}$$

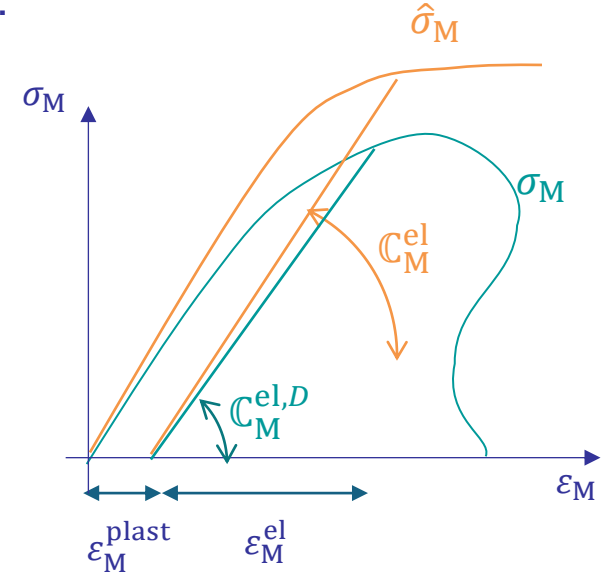
- New macro-scale Helmholtz equations

$$D_{M_a} - l_d^2 \nabla^2 \tilde{D}_{M_a} = D_{M_a}$$

$$\tilde{\varepsilon}_{M_b}^{\text{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{M_b}^{\text{plast}} = \varepsilon_{M_b}^{\text{plast}}$$

- Reconstructed (mesh-insensitive) apparent stress

$$\boldsymbol{\sigma}_{M_{\text{rec}}} = [(1 - \tilde{D}_M) \odot \mathbb{C}_M^{\text{el}}] : (\boldsymbol{\varepsilon}_M - \tilde{\boldsymbol{\varepsilon}}_M^{\text{plast}})$$



- Development of an embedding non-local framework:

- Identification of local/non-local variables

- Plastic strain tensor (from unloading in apparent space)

$$\boldsymbol{\varepsilon}_M^{\text{plast}} = \boldsymbol{\varepsilon}_M - \mathbb{C}_M^{\text{el},D^{-1}} : \boldsymbol{\sigma}_M$$

- Component-wise damage (to be reduced*)

$$D_{Mijkl} = 1 - \frac{C_{Mijkl}^{\text{el},D}}{C_{Mijkl}^{\text{el}}}$$

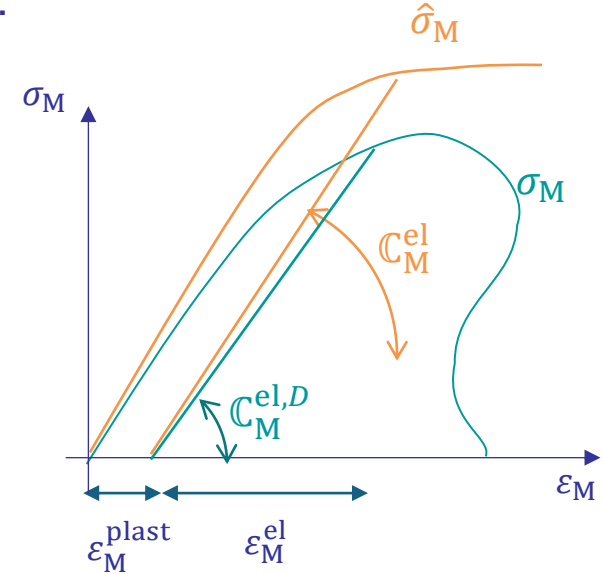
- New macro-scale Helmholtz equations

$$D_{M_a} - l_d^2 \nabla^2 \tilde{D}_{M_a} = D_{M_a}$$

$$\tilde{\varepsilon}_{M_b}^{\text{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{M_b}^{\text{plast}} = \varepsilon_{M_b}^{\text{plast}}$$

- Reconstructed (mesh-insensitive) apparent stress

$$\boldsymbol{\sigma}_{M_{\text{rec}}} = [(1 - \tilde{D}_M) \odot \mathbb{C}_M^{\text{el}}] : (\boldsymbol{\varepsilon}_M - \tilde{\boldsymbol{\varepsilon}}_M^{\text{plast}})$$



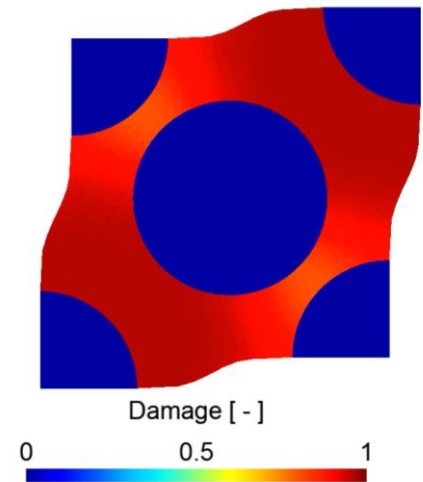
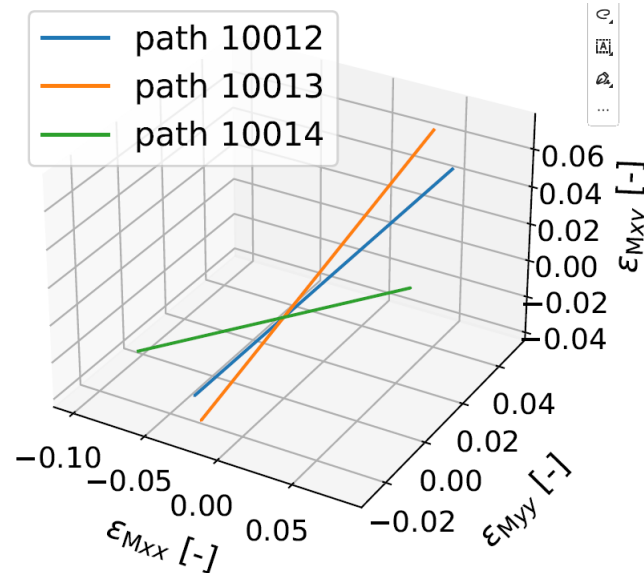
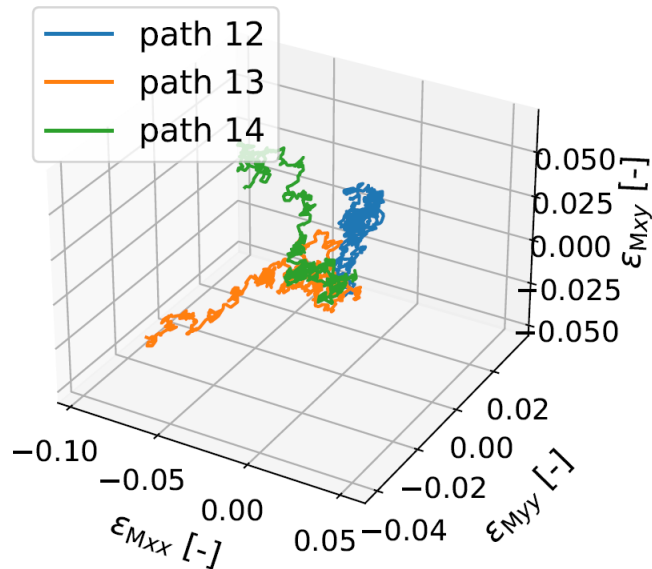
➡ Neural networks to be developed:

- Inputs: $\underline{\mathbf{u}}_t = \{\boldsymbol{\varepsilon}_M\}_t$
- Outputs: $\underline{\mathbf{v}}_t = \{\boldsymbol{\sigma}_M, C_M^{\text{el},D}, C_M^{\text{el}}\}_t$

- Training data-base

- ~10 000 random paths

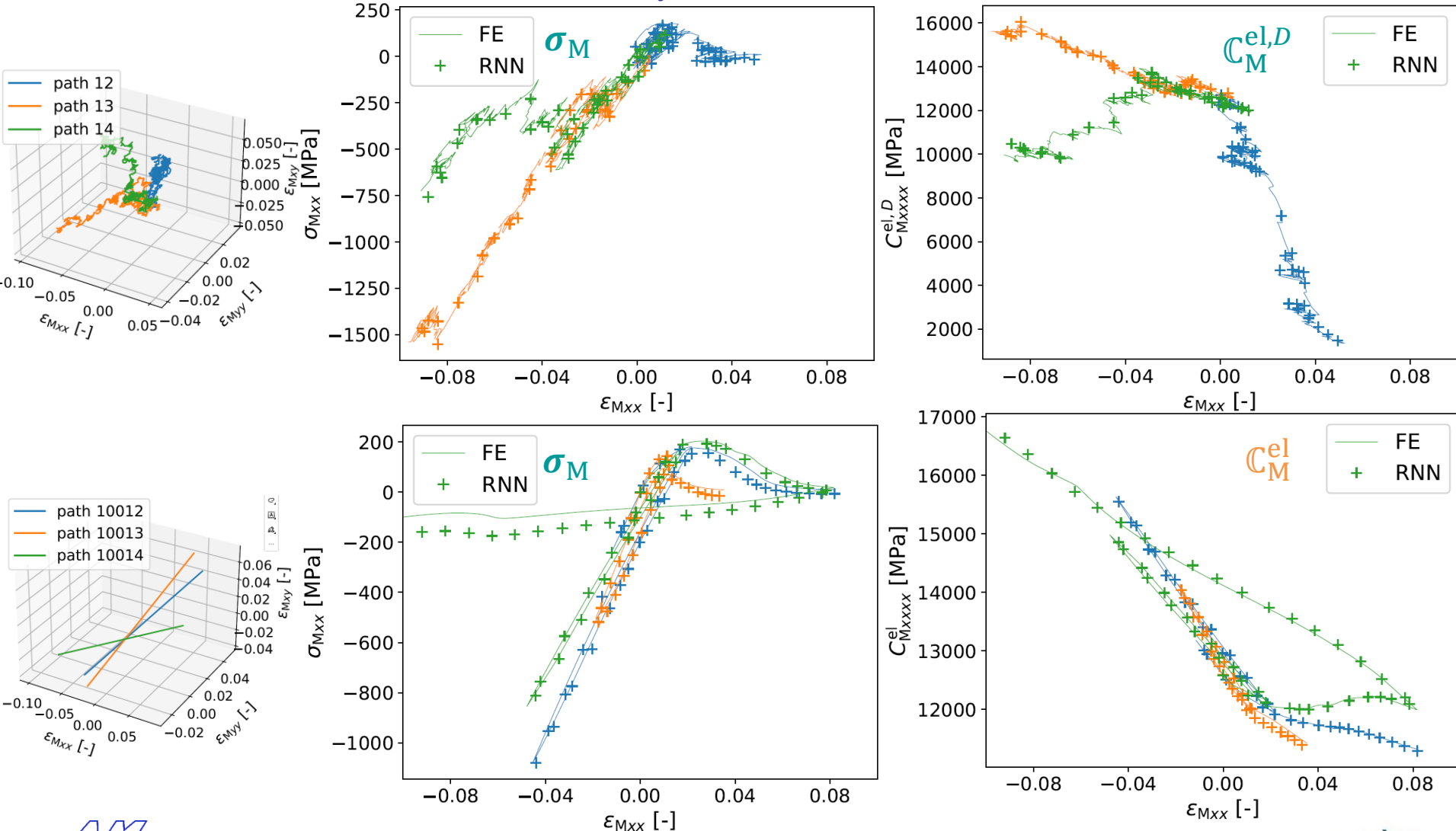
- ~ 5000 cyclic paths



AI-accelerated multi-scale analysis of composite failure

- Testing the neural networks

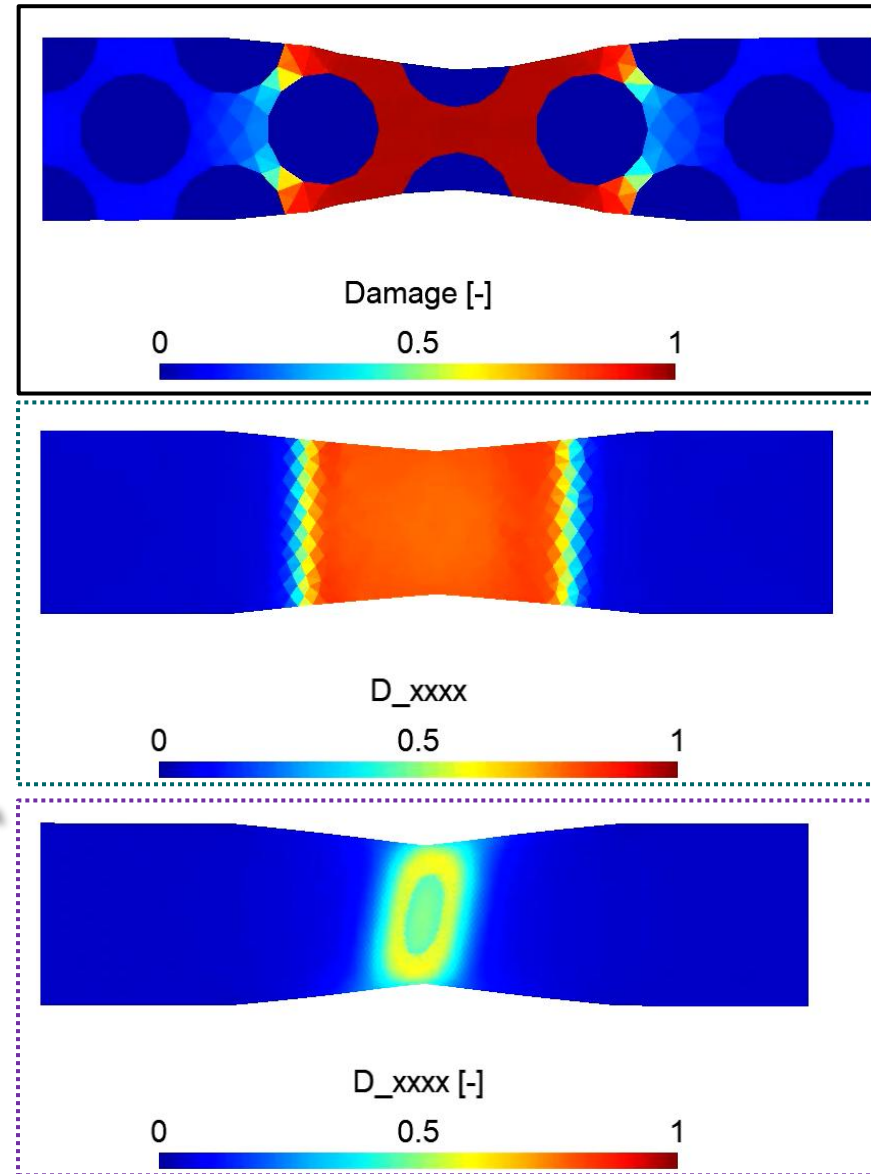
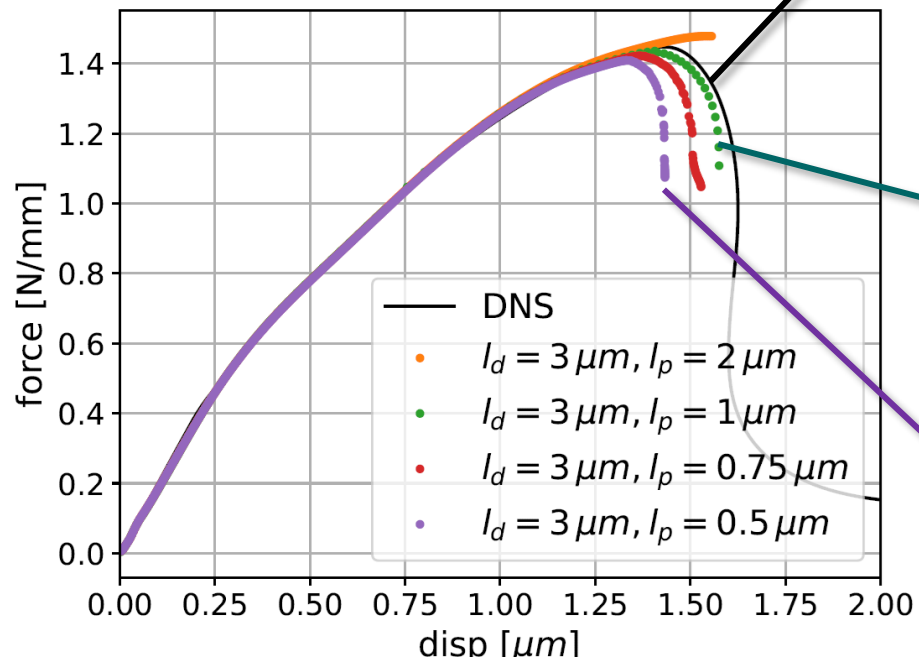
– $\underline{u}_t = \{\boldsymbol{\varepsilon}_M\}_t$ vs. $\underline{v}_t = \{\boldsymbol{\sigma}_M, \mathbb{C}_M^{\text{el},D}, \mathbb{C}_M^{\text{el}}\}_t$



- Non-local lengths

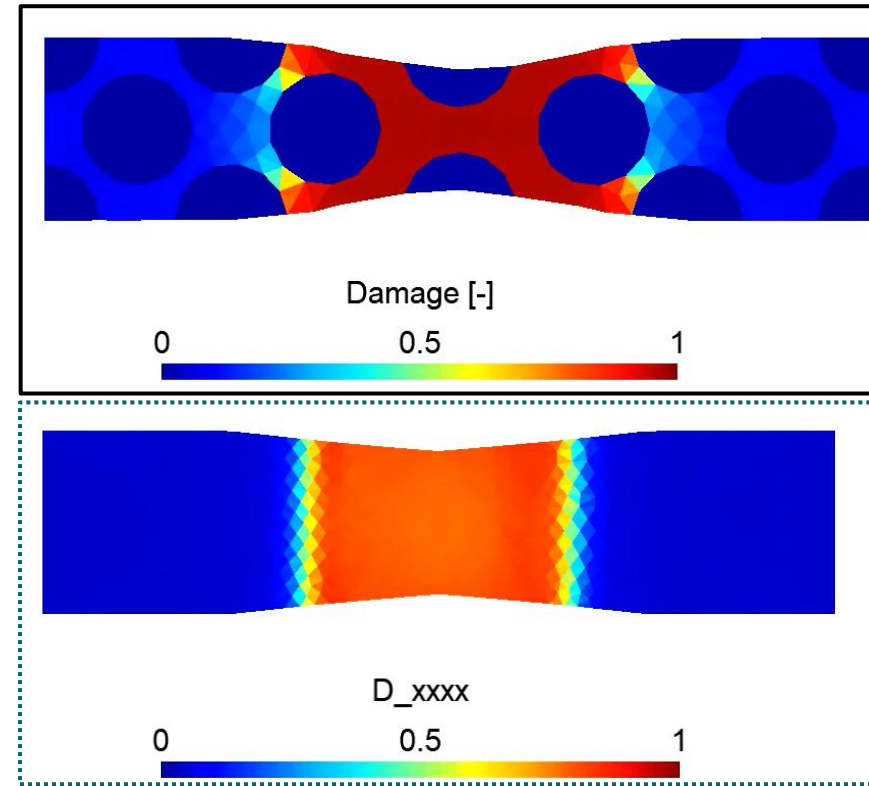
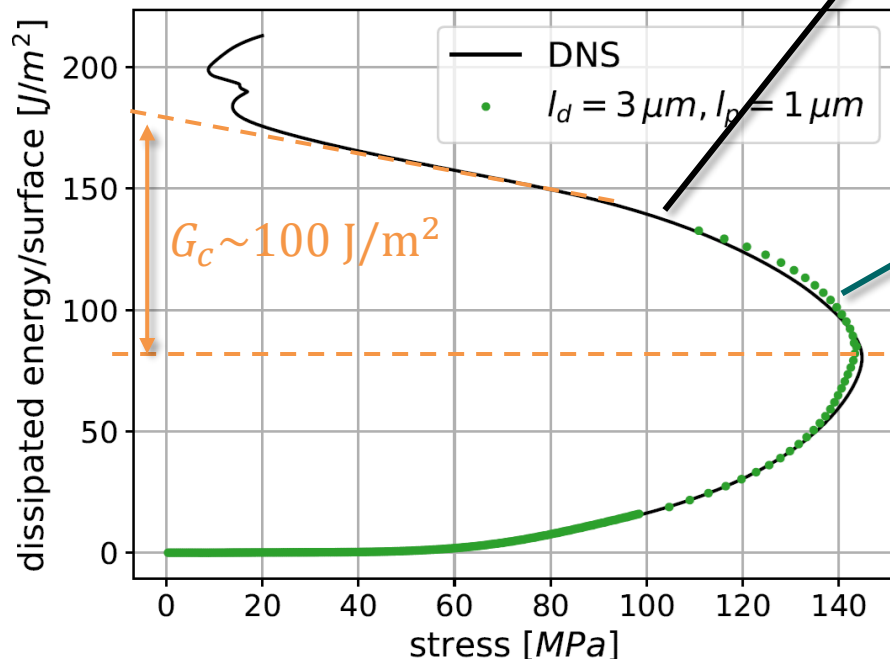
- $$\begin{cases} D_{M_a} - l_d^2 \nabla^2 \tilde{D}_{M_a} = D_{M_a} \\ \tilde{\varepsilon}_{M_b}^{\text{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{M_b}^{\text{plast}} = \varepsilon_{M_b}^{\text{plast}} \end{cases}$$

- Calibration from DNS



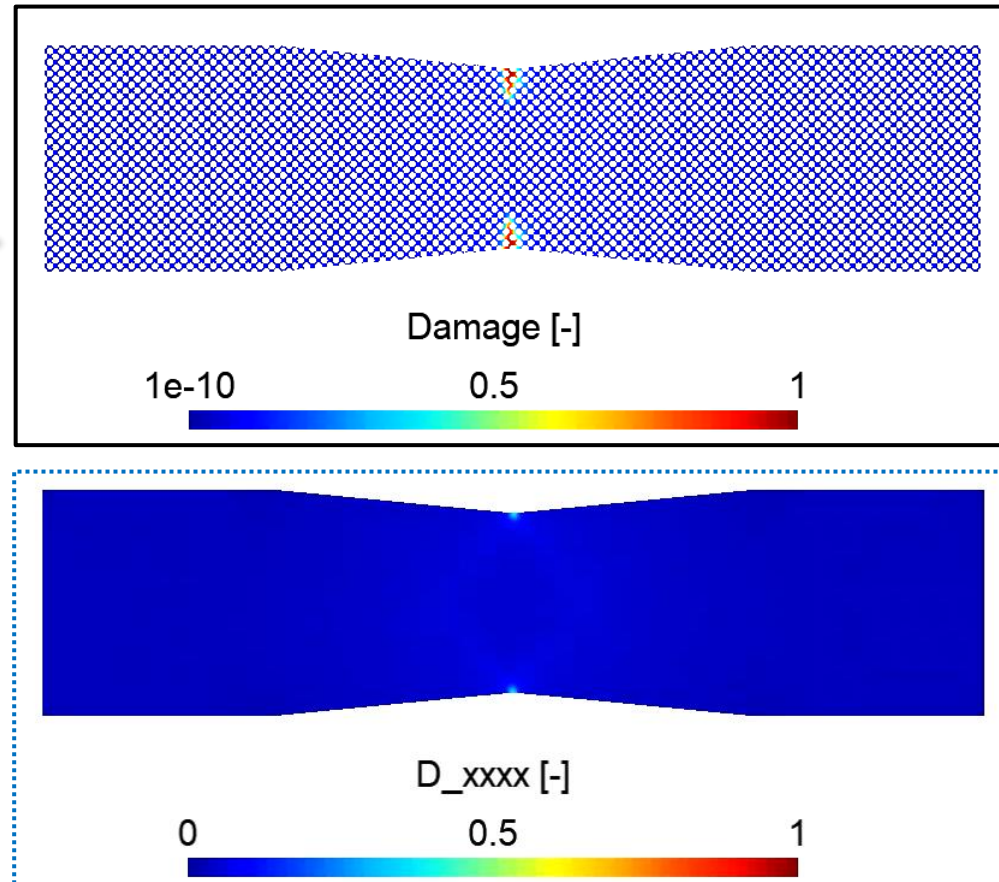
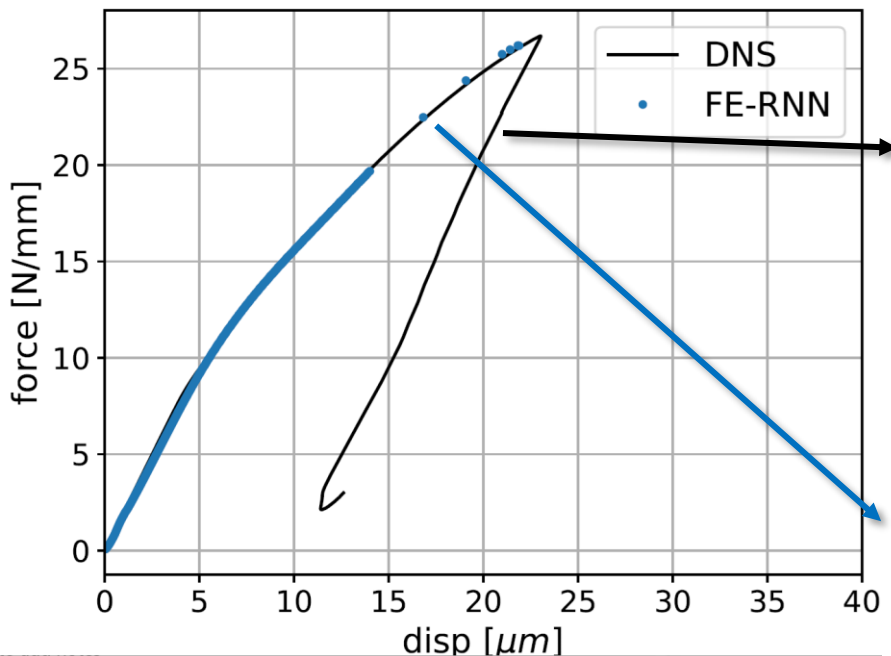
- Non-local lengths

- $$\begin{cases} D_{M_a} - l_d^2 \nabla^2 \tilde{D}_{M_a} = D_{M_a} \\ \tilde{\varepsilon}_{M_b}^{\text{plast}} - l_p^2 \nabla^2 \tilde{\varepsilon}_{M_b}^{\text{plast}} = \varepsilon_{M_b}^{\text{plast}} \end{cases}$$
- 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

- 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, ...)
 -

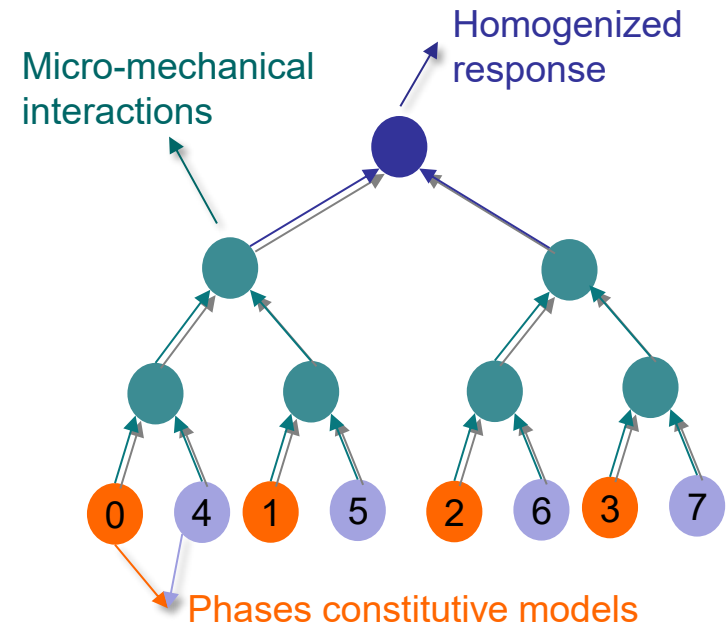
- Use of RNN

- 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

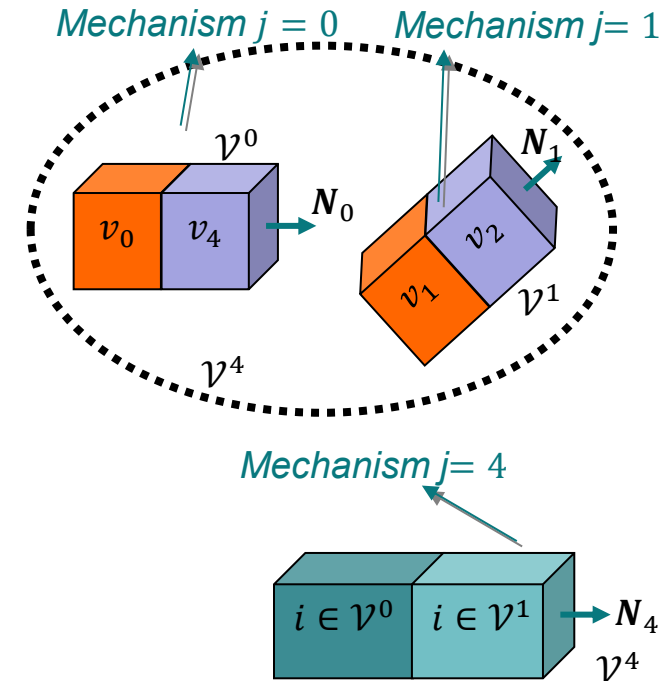
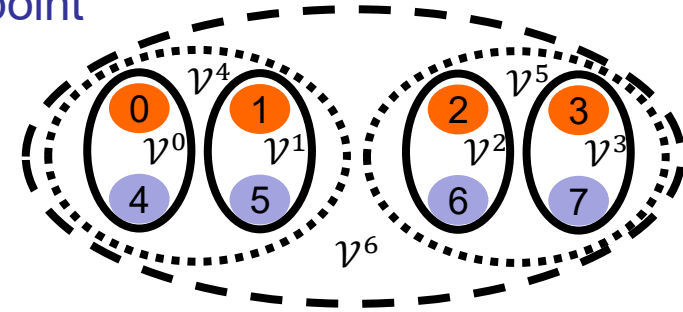
$$\mathbf{P}_M(t) = \sum_{i=0}^7 v_i \mathbf{P}_i(t) \quad v_i = \frac{V_i}{V_M}$$

$$\mathbf{F}_M(t) = \sum_{i=0}^7 v_i \mathbf{F}_i(t)$$

- Fluctuation field $\mathbf{u}'$

$$\mathbf{F}_m(\mathbf{X}_m) = \mathbf{F}_M + \mathbf{u}'(\mathbf{X}_m) \otimes \nabla_0$$

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



- Deep Material Networks from the interaction viewpoint

- Strain/Stress averaging

$$\mathbf{P}_M(t) = \sum_{i=0}^7 v_i \mathbf{P}_i(t) \quad v_i = \frac{V_i}{V_M}$$

$$\mathbf{F}_M(t) = \sum_{i=0}^7 v_i \mathbf{F}_i(t)$$

- Fluctuation filed $\mathbf{u}'$

$$\mathbf{F}_m(\mathbf{X}_m) = \mathbf{F}_M + \mathbf{u}'(\mathbf{X}_m) \otimes \nabla_0$$

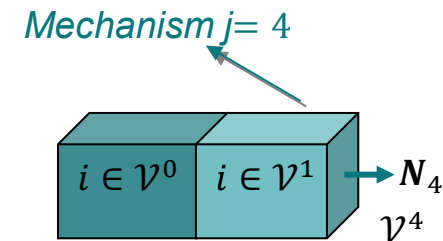
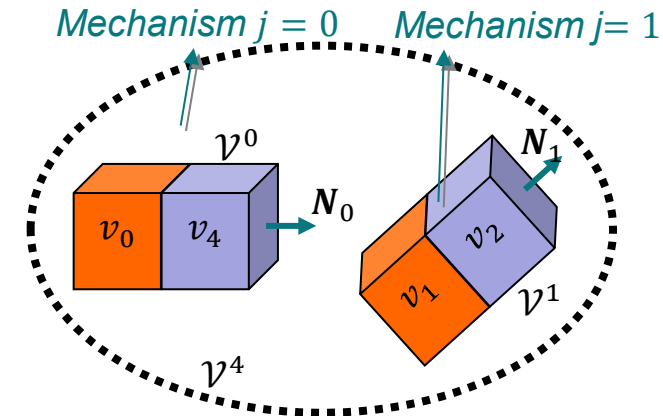
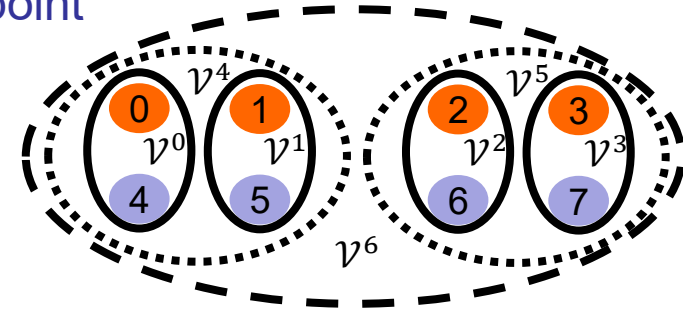
➡ $\mathbf{F}_i = \mathbf{F}_M + \frac{1}{V_i} \sum_{(i,j)} s_j^i \mathbf{u}'_j \otimes \mathbf{N}_j^i, \quad i = 0..7$

- Weak form from Hill Mandel condition

$$\mathbf{P}_M : \delta \mathbf{F}_M = \sum_i v_i \mathbf{P}_i : \delta \mathbf{F}_i$$

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

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



• Deep Material Networks from the interaction viewpoint

- Strain/Stress averaging

$$\mathbf{P}_M(t) = \sum_{i=0}^7 v_i \mathbf{P}_i(t)$$

$$\mathbf{F}_M(t) = \sum_{i=0}^7 v_i \mathbf{F}_i(t)$$

$$v_i = \frac{V_i}{V_M}$$

Weight of node i (parameter)

- Fluctuation filed $\mathbf{u}'$

$$\mathbf{F}_m(\mathbf{X}_m) = \mathbf{F}_M + \mathbf{u}'(\mathbf{X}_m) \otimes \nabla_0$$

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

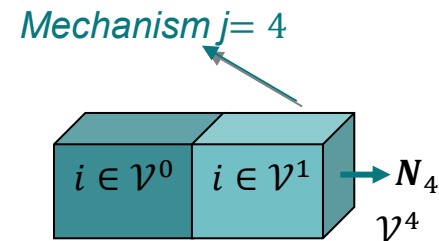
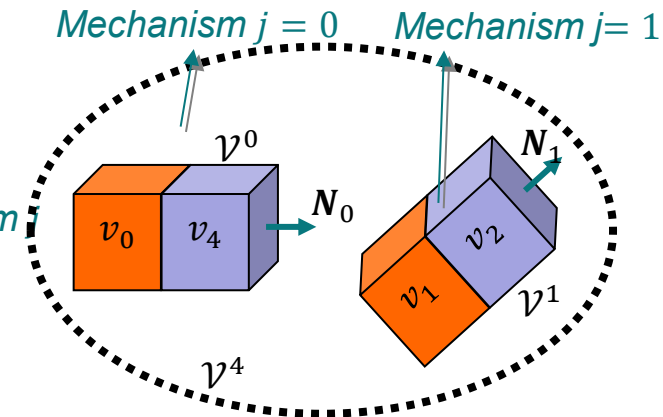
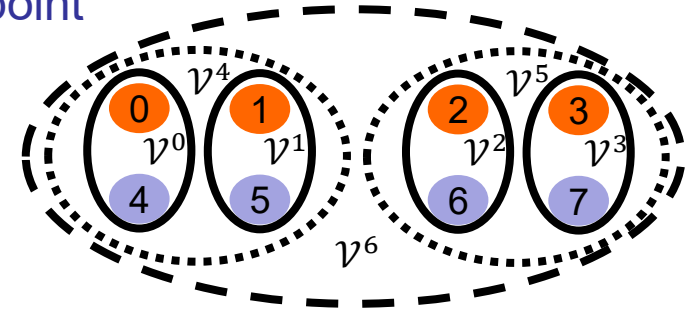
Contribution of node i in mechanism j (parameter)

Direction of mechanism (parameter)

- Offline stage: Parameters $(v_i, s_j^i, \mathbf{N}_j^i)$

- Obtained by training
- Should be independent from the constitutive behaviours
- Should be independent from load paths

Representative of the micro-structure organisation only



- Deep Material Networks from the interaction viewpoint

- Strain/Stress averaging

$$\mathbf{P}_M(t) = \sum_{i=0}^7 v_i \mathbf{P}_i(t)$$

$$\mathbf{F}_M(t) = \sum_{i=0}^7 v_i \mathbf{F}_i(t)$$

$$v_i = \frac{V_i}{V_M}$$

- Fluctuation filed $\mathbf{u}'$

$$\mathbf{F}_m(\mathbf{X}_m) = \mathbf{F}_M + \mathbf{u}'(\mathbf{X}_m) \otimes \nabla_0$$

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

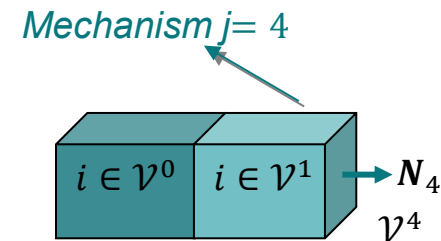
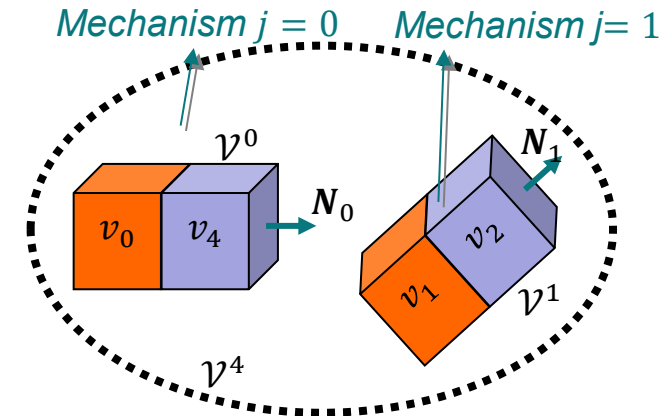
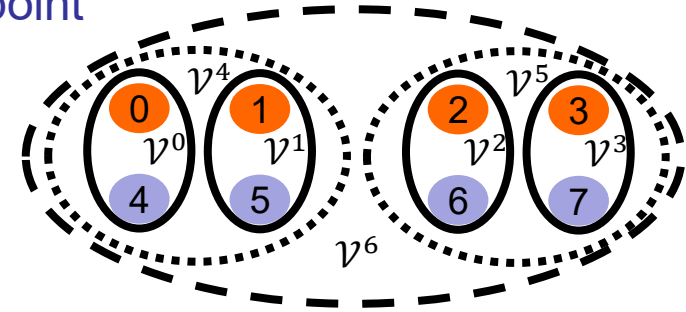
Degrees of freedom of mechanism j defining the strain fluctuation

- Online stage: Weak form from Hill Mandel condition

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

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

➡ Non-linear problem for given loading/material models



- Training of Interaction-Based Deep Material Networks
 - Summary

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

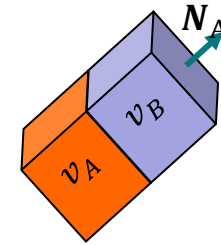
- Training of Interaction-Based Deep Material Networks

- Summary

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

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

- Analytical solution



New dof: $\hat{\mathbf{a}} = \frac{s_{AB}}{V_M} \mathbf{u}'$

$$\left\{ \begin{array}{l} \boldsymbol{\varepsilon}_A = \boldsymbol{\varepsilon}_M + \frac{1}{v_A} \frac{s_{AB}}{V_M} \mathbf{u}' \otimes^s \mathbf{N}_A \text{ and} \\ \boldsymbol{\varepsilon}_B = \boldsymbol{\varepsilon}_M - \frac{1}{v_B} \frac{s_{AB}}{V_M} \mathbf{u}' \otimes^s \mathbf{N}_A \\ \mathbb{C}_M = v_A \mathbb{C}_A + (1.0 - v_A) \mathbb{C}_B + (\mathbb{C}_A - \mathbb{C}_B) \cdot \mathbf{N}_A \otimes^s [\mathcal{K}^{-1}(v_A, \mathbf{N}_A) \cdot \mathcal{F}_M(v_A, \mathbf{N}_A)] \end{array} \right.$$

Topology parameters to be defined $\mathcal{G}^2 = \{v_A, \mathbf{N}_A\}$

$$\mathbb{C}_M = \text{FUN}\left(\mathbb{C}_A, \mathbb{C}_B, \mathcal{G}^2(v_A, \mathbf{N}_A)\right)$$



AI-accelerated multi-scale analysis of composites

- Training of Interaction-Based Deep Material Networks

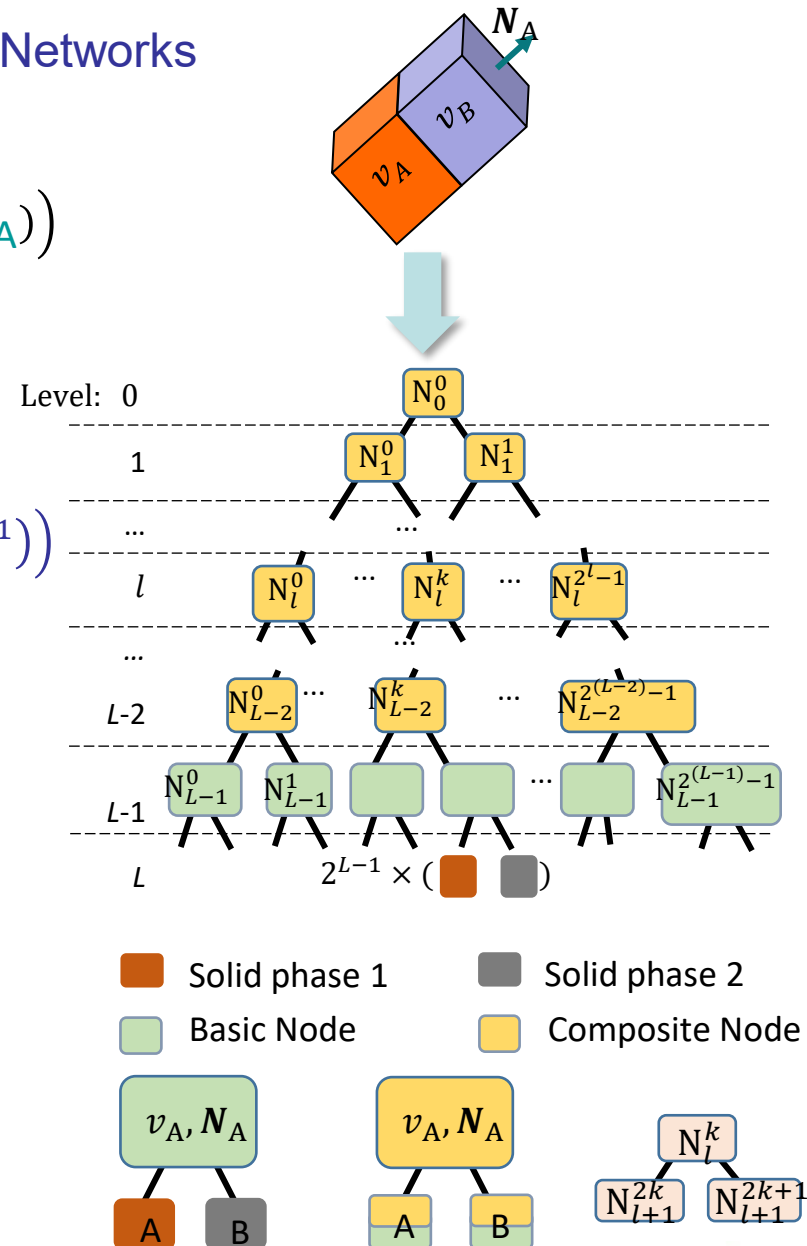
- Elasticity with multi-level tree

- Generalizing $\mathbb{C}_M = \text{FUN}(\mathbb{C}_A, \mathbb{C}_B, \mathcal{G}^2(v_A, N_A))$

➔ $\mathbb{C}(N_l^k) = \text{FUN}(l, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_l^k))$

with

$$\begin{cases} \mathbb{C}(N_{l+1}^{2k}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k})) \\ \mathbb{C}(N_{l+1}^{2k+1}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k+1})) \end{cases}$$



AI-accelerated multi-scale analysis of composites

- Training of Interaction-Based Deep Material Networks

- Elasticity with multi-level tree

- Generalizing $\mathbb{C}_M = \text{FUN}(\mathbb{C}_A, \mathbb{C}_B, \mathcal{G}^2(v_A, N_A))$

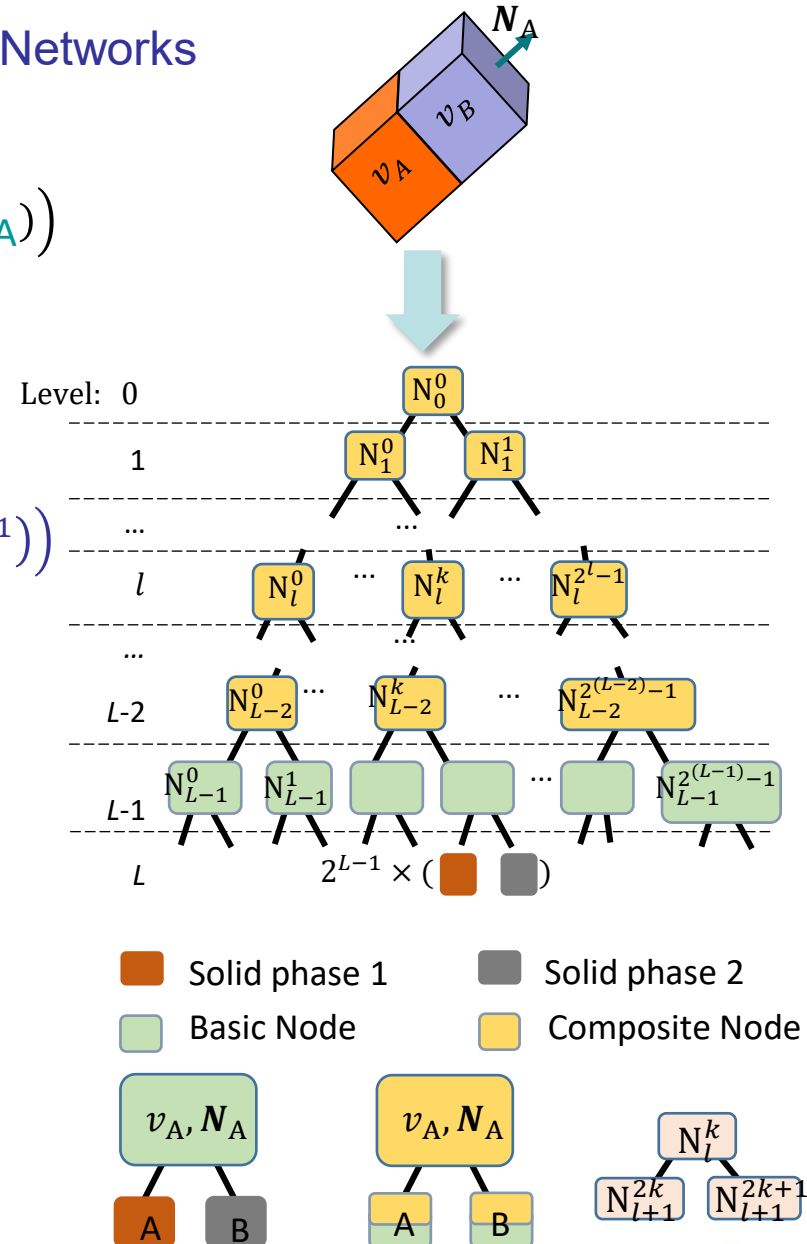
➡ $\mathbb{C}(N_l^k) = \text{FUN}(l, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_l^k))$

with

$$\begin{cases} \mathbb{C}(N_{l+1}^{2k}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k})) \\ \mathbb{C}(N_{l+1}^{2k+1}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k+1})) \end{cases}$$

- Recursive analytical solution

➡ $\mathbb{C}_M = \text{FUN}(l=0, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_0^0))$



AI-accelerated multi-scale analysis of composites

• Training of Interaction-Based Deep Material Networks

– Elasticity with multi-level tree

- Generalizing $\mathbb{C}_M = \text{FUN}(\mathbb{C}_A, \mathbb{C}_B, \mathcal{G}^2(v_A, N_A))$

→ $\mathbb{C}(N_l^k) = \text{FUN}(l, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_l^k))$

with

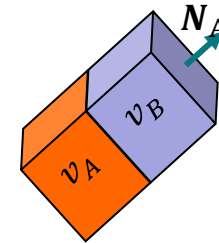
$$\begin{cases} \mathbb{C}(N_{l+1}^{2k}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k})) \\ \mathbb{C}(N_{l+1}^{2k+1}) = \text{FUN}(l+1, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_{l+1}^{2k+1})) \end{cases}$$

- Recursive analytical solution

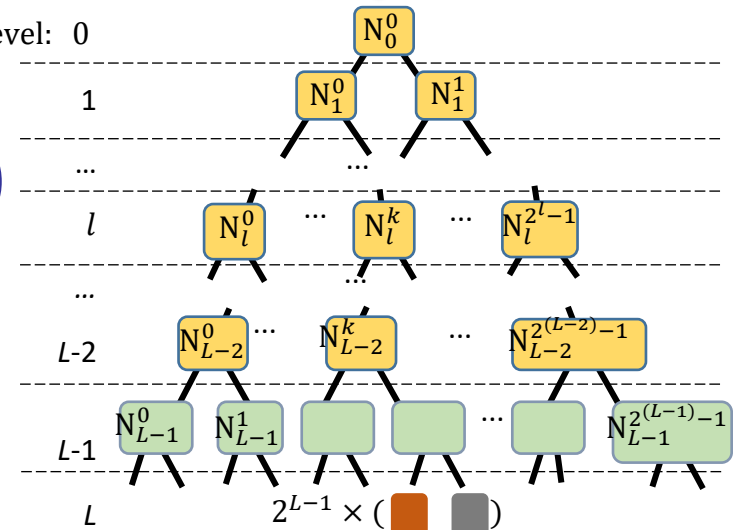
→ $\mathbb{C}_M = \text{FUN}(l=0, \mathbb{C}_0, \mathbb{C}_l, \mathcal{G}^2(N_0^0))$

→ Topology parameters to be defined:

$$\mathcal{G}^2(N_0^0), \mathcal{G}^2(N_1^0), \mathcal{G}^2(N_1^1), \dots, \mathcal{G}^2(N_{L-1}^{2^{(L-1)}-1})$$



Level: 0

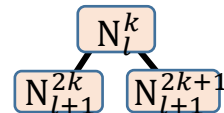
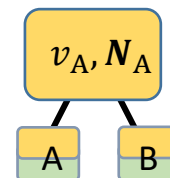
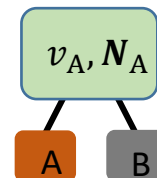


■ Solid phase 1

■ Solid phase 2

■ Basic Node

■ Composite Node l



• Training of Interaction-Based Deep Material Networks

– Summary

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

– Analytical solution in linear elasticity

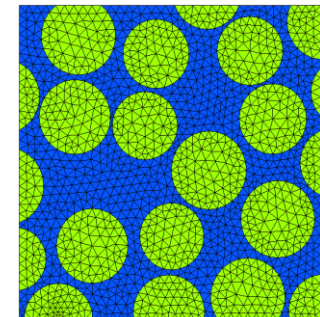
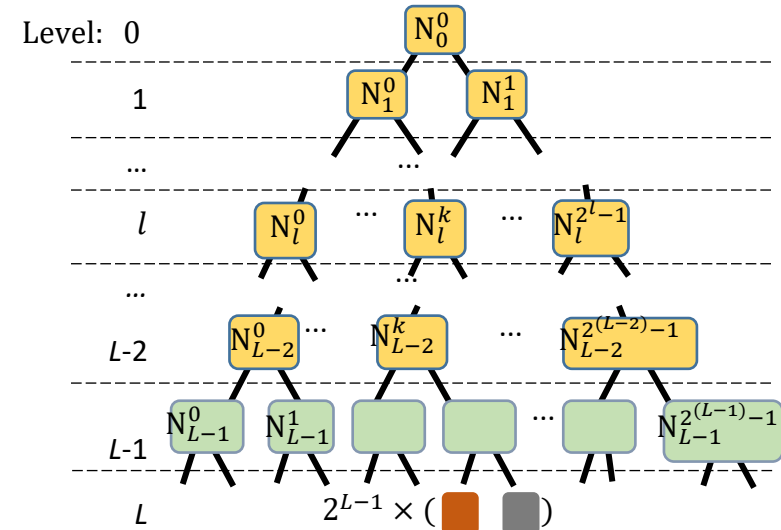
• Recursively

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

• Topological parameters

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

To be identified for a given micro-structure



• Training of Interaction-Based Deep Material Networks

– Summary

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

– Analytical solution in linear elasticity

• Recursively

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

• Topological parameters

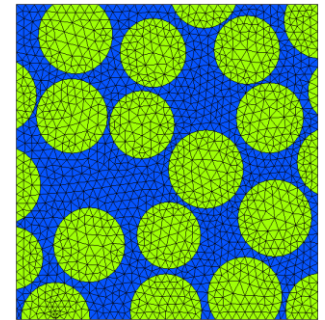
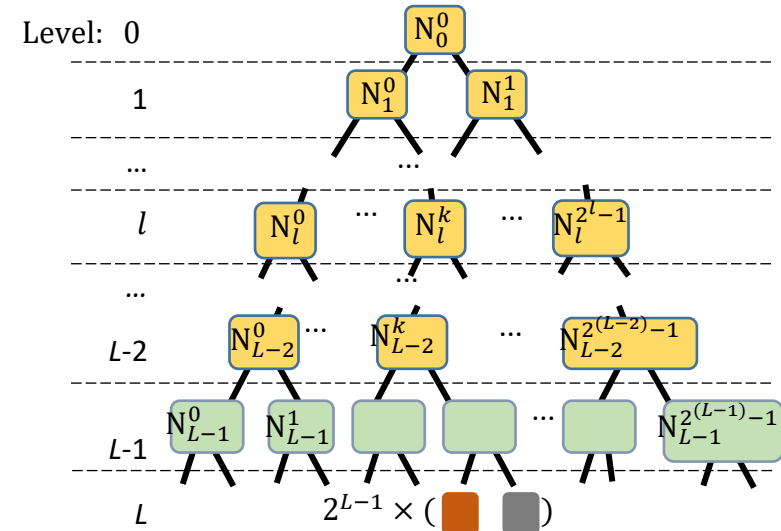
$$\mathcal{G}^2(\mathbf{N}_0^0), \mathcal{G}^2(\mathbf{N}_1^0), \mathcal{G}^2(\mathbf{N}_1^1), \dots, \mathcal{G}^2(\mathbf{N}_{L-1}^{2^{(L-1)}-1})$$

– Data driven approach:

• Generate observations $\{\hat{\mathbb{C}}_M(\mathbb{C}_0^s, \mathbb{C}_l^s)\}$ by full DNS using random $\mathbb{C}_0^s, \mathbb{C}_l^s$

• Identify topological parameters from loss function (knowing real volume fraction $\hat{v}_l$ of phase I)

$$\text{Loss}(\hat{\mathbb{C}}_M, \mathbb{C}_M) = \frac{1}{n} \sum_{s=0}^{n-1} \frac{\|\hat{\mathbb{C}}_M(\mathbb{C}_0^s, \mathbb{C}_l^s) - \mathbb{C}_M(\mathbb{C}_0^s, \mathbb{C}_l^s; \mathcal{G}^2)\|}{\|\hat{\mathbb{C}}_M(\mathbb{C}_0^s, \mathbb{C}_l^s)\|} + \frac{\lambda}{2} (\hat{v}_l - v_l)^2$$



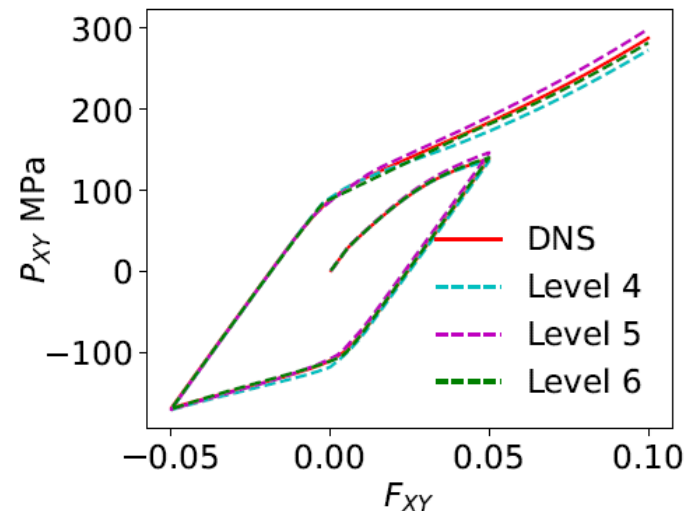
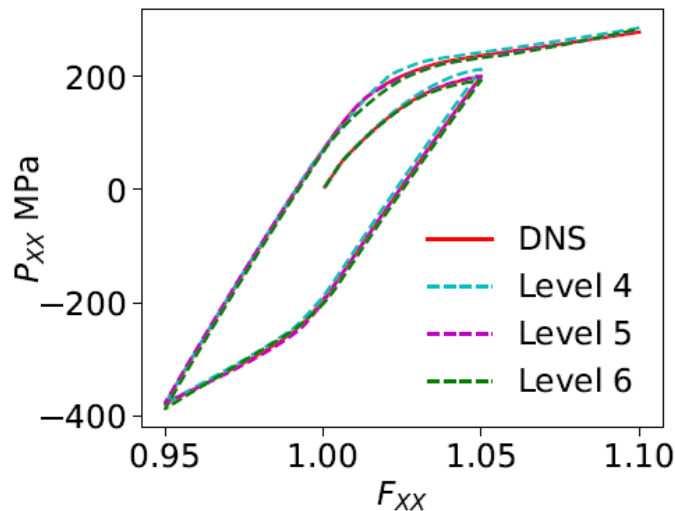
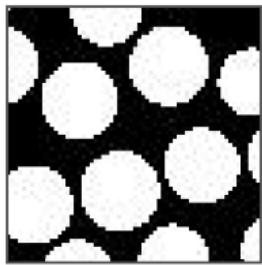
- Testing of Interaction-Based Deep Material Networks

- For one micro-structure infer one set of topological parameters

➡ $\left\{ \mathcal{G}^2(N_0^0), \mathcal{G}^2(N_1^0), \mathcal{G}^2(N_1^1), \dots, \mathcal{G}^2\left(N_{L-1}^{2^{(L-1)}-1}\right) \right\}$

- Testing on complex elasto-plasticity

- Non-associated pressure-dependent plasticity
- Extrapolation in terms of loading path
- Extrapolation in terms of constitutive model



Can we extrapolate with damage-enhanced model at the material nodes?

Damage-enhanced Interaction-Based Deep Material Network

- Can we extrapolate with damage-enhanced model at the material nodes?

- Train 6 DMNs of different volume fractions

$$v_I = 0.330$$

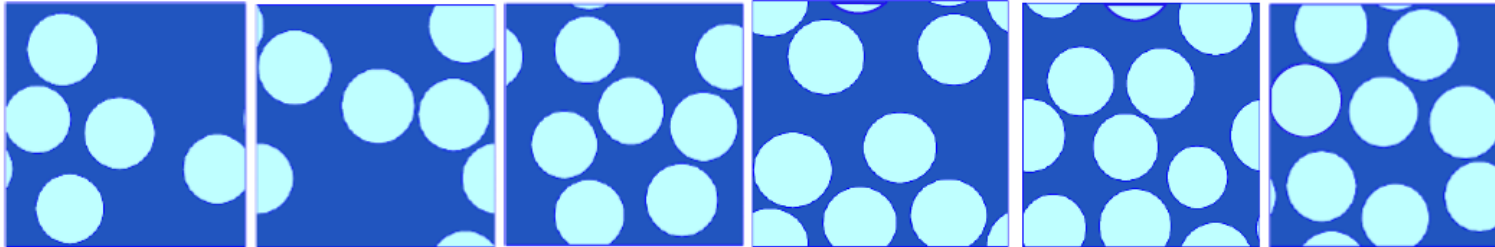
$$v_I = 0.361$$

$$v_I = 0.450$$

$$v_I = 0.486$$

$$v_I = 0.526$$

$$v_I = 0.554$$

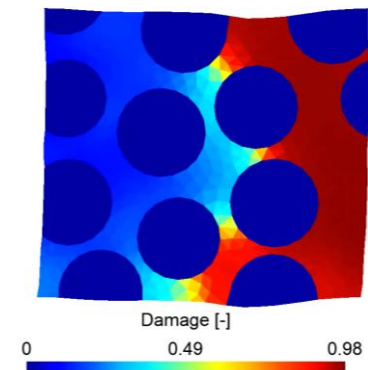
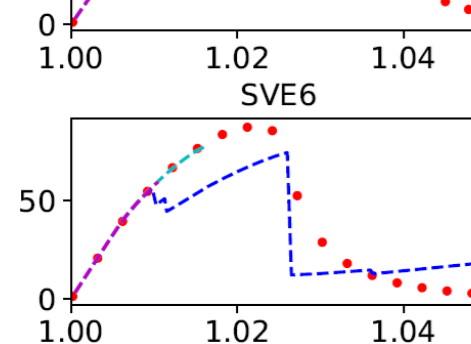
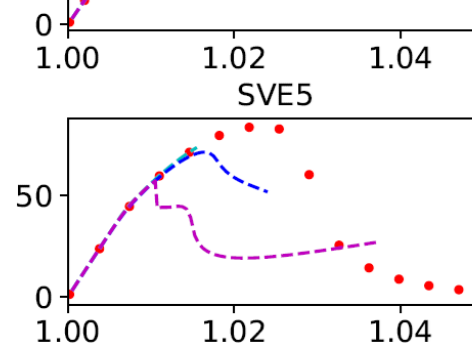
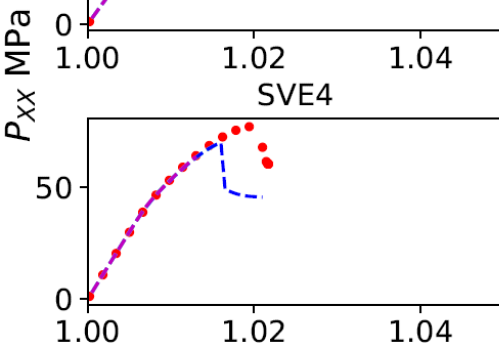
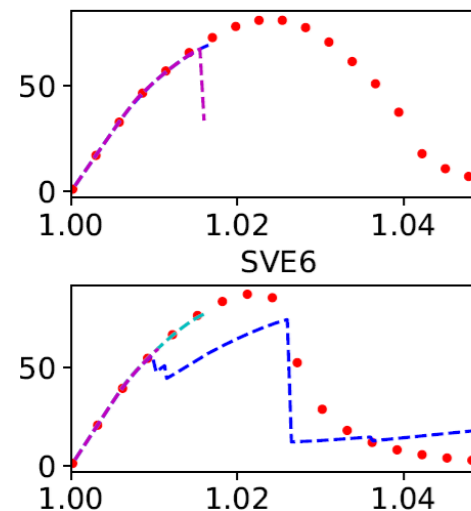
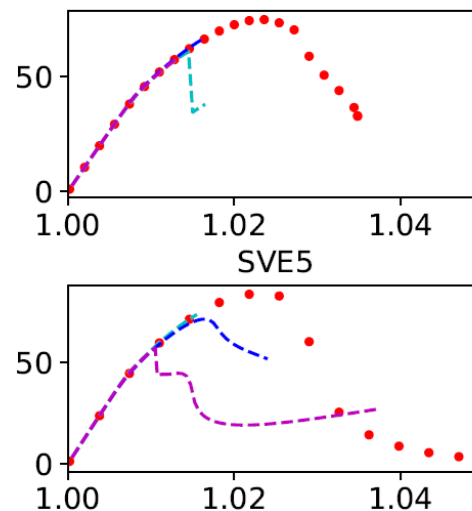
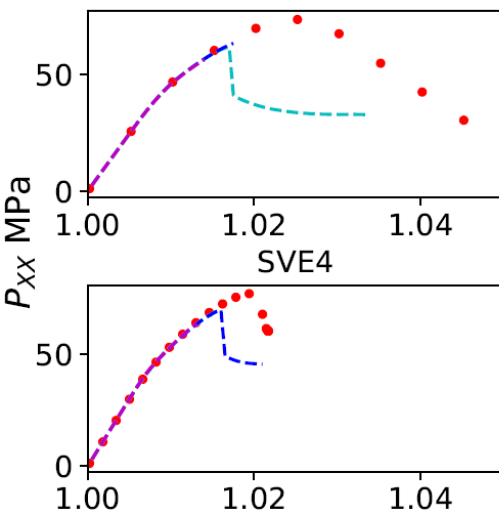


- Introduction of damage in the matrix model

DNS: $\bullet$ $l_c = 3\mu\text{m}$
SVE1

DMN of 5 Level with local damage: ---
SVE2

SVE3



DMN exhibits strain localization at some material nodes (like local FEM)

Discrete Non-Local Damage Deep Material Network

- Regularisation of the IB-DMN

- Classical non-local form

- $D(\tilde{\psi})$ with $\tilde{\psi}(x_0) = \int_V w(x - x_0) \psi(x) dx$

- $\mathbf{P} = \mathbf{P}^p(\mathbf{F}; \boldsymbol{\psi}) \rightarrow \mathbf{P} = \mathbf{P}^p(\mathbf{F}; \boldsymbol{\psi}, \tilde{\boldsymbol{\psi}})$

- Equivalent discrete form for the IB-DMB

$$\tilde{\psi}_i = \sum_{j=1}^n w_{ij} v_j \psi_j$$

- Under the constraints

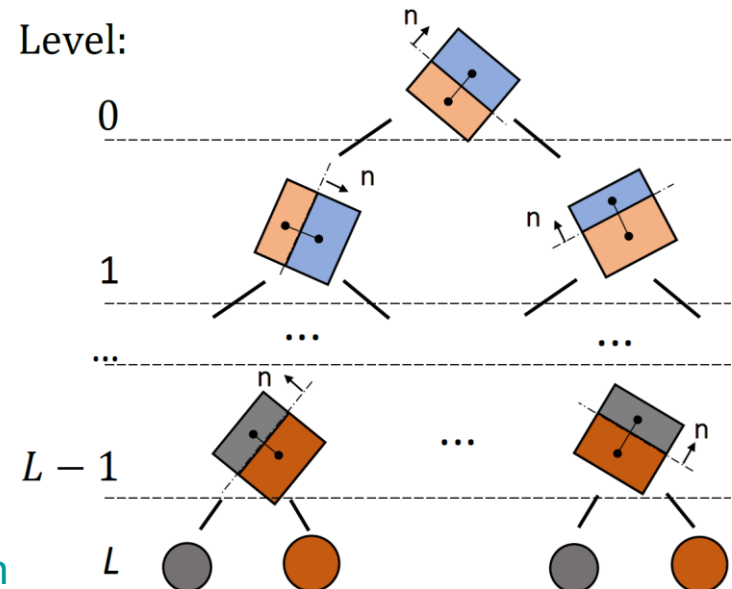
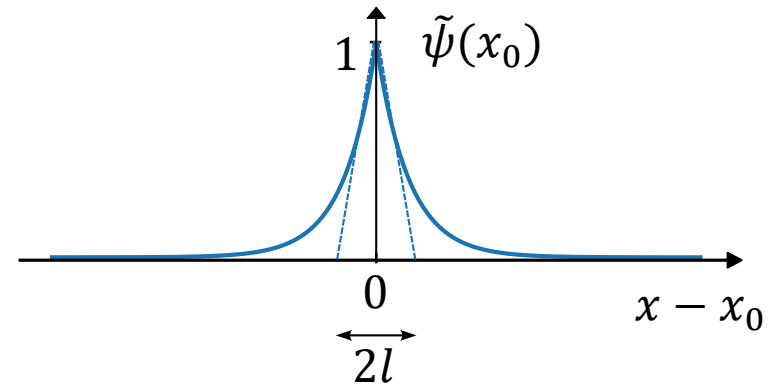
$$\left\{ \begin{array}{l} w_{ij} = w_{ji} \quad \sum_{j=1}^n w_{ij} v_j = 1 \\ \psi_i = \sum_{j=1}^n w_{ij} v_j \tilde{\psi}_j \end{array} \right.$$

- Satisfied by introducing auxiliary function

$$w_{ij} \propto \exp(-\sqrt{d_{ij} \cdot \mathbf{c}^{-1} \cdot d_{ij}})$$

Squared length scales

How to evaluate the distance in an abstract tree ?



- Implementation of discrete non-local IB-DMN

- Discrete Nonlocal form

- $$\tilde{\psi}_i = \sum_{j=1}^n w_{ij} v_j \psi_j \quad \Rightarrow \quad \mathbf{P} = \mathbf{P}^p(\mathbf{F}; \boldsymbol{\psi}, \tilde{\boldsymbol{\psi}})$$

- w_{ij} under constraints

- $w_{ij} \propto \exp(-\sqrt{\mathbf{d}_{ij} \cdot \mathbf{c}^{-1} \cdot \mathbf{d}_{ij}})$

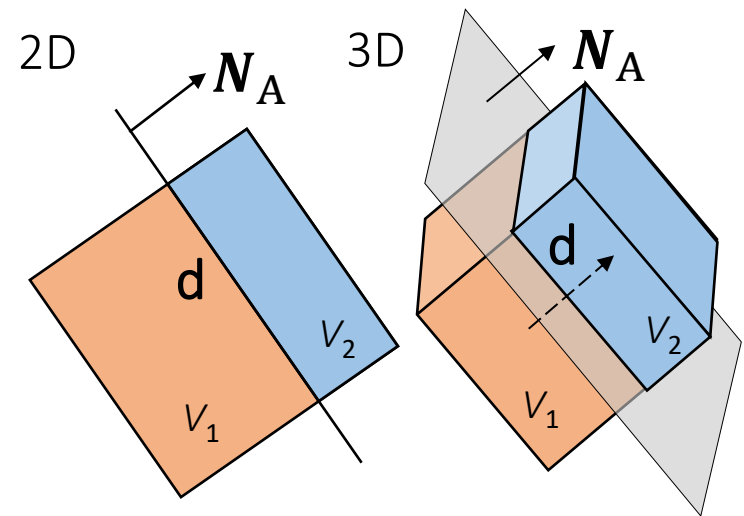
- Auxiliary function

- $g_{ij} = \exp(-\sqrt{\mathbf{d}_{ij} \cdot \mathbf{c}^{-1} \cdot \mathbf{d}_{ij}})$
 - Characteristic lengths $\mathbf{c} = \text{diag}(l_0^2, l_1^2, l_2^2)$
 - $\mathbf{d}_{ij} = \frac{\dim \sqrt{V_i + V_j}}{2} \mathbf{N}$

- Evaluation of the coefficients (only once)

$\Rightarrow w_{ij} = \alpha_i g_{ij} \alpha_j$

$\Rightarrow \alpha_i$ is solved to satisfy the constraints on w_{ij}



Damage-enhanced Interaction-Based Deep Material Network

- Test: Multi-damage-enhanced non-associated model in the matrix phase

- Train 6 (smaller) DMNs of different volume fractions

- Best training for each volume element

DNS: • DMN of 5 Level: — $l_c^s = 2, l_c^f = 3$; — $l_c^s = 3, l_c^f = 3\mu\text{m}$

SVE1

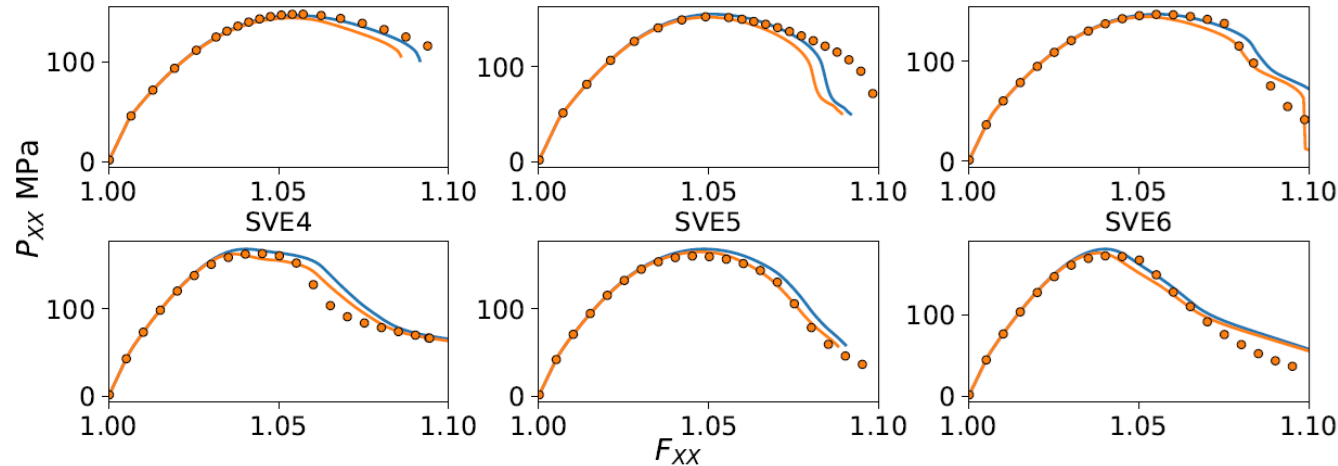
SVE2

SVE3

SVE4

SVE5

SVE6



DNS: • DMN of 5 Level: — $l_c^s = 2, l_c^f = 3$; — $l_c^s = 3, l_c^f = 3\mu\text{m}$

SVE1

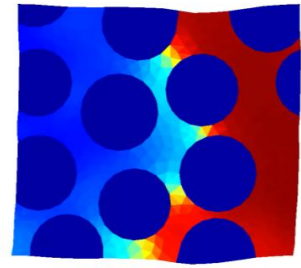
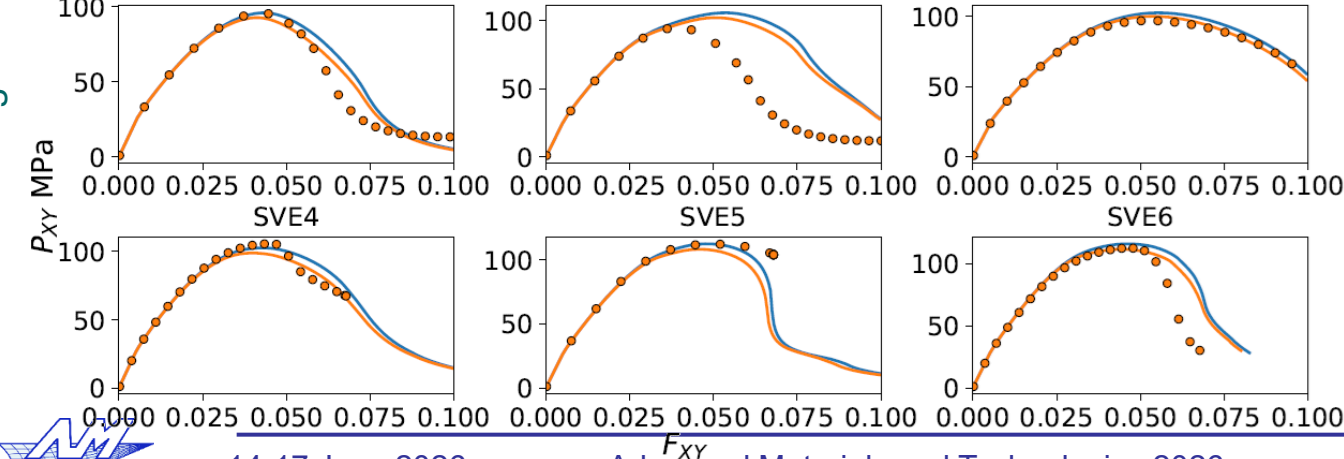
SVE2

SVE3

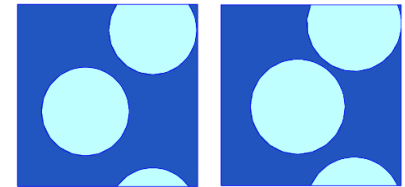
SVE4

SVE5

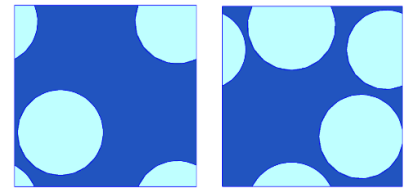
SVE6



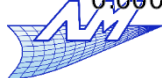
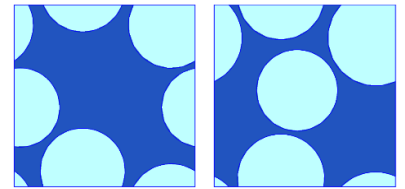
$v_I = 0.372$ $v_I = 0.432$



$v_I = 0.344$ $v_I = 0.510$



$v_I = 0.499$ $v_I = 0.556$



- Conclusions

- Data-driven method for failure analysis

- 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

- 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.