Computational & Multiscale
Mechanics of Materials

CM3
www.ltas-cm3.ulg.ac.be

A Self-consistent Reinforced minimal Gated Recurrent Unit for surrogate modelling of elasto-plastic multi-scale problems

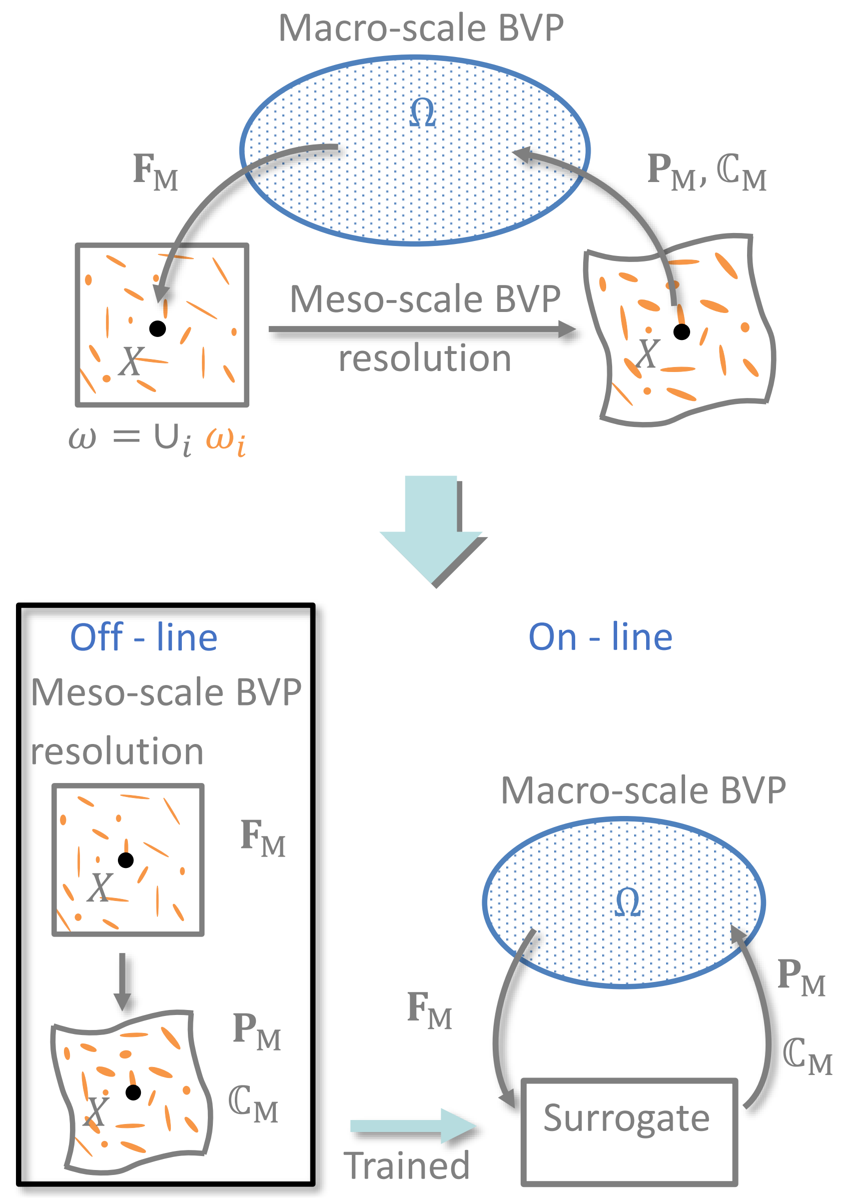
Wu Ling, Noels Ludovic

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

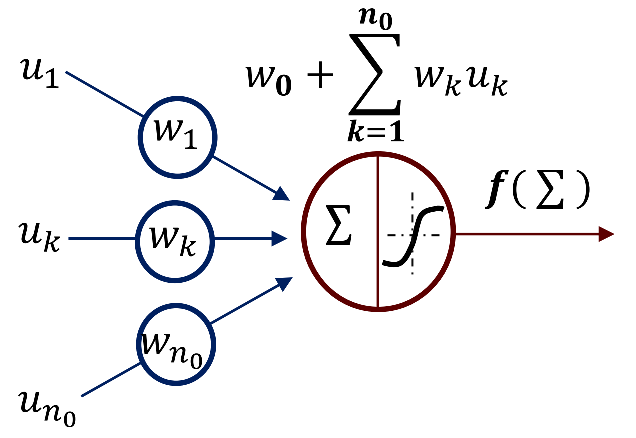
LIÈGE université

AI-accelerated multi-scale simulations

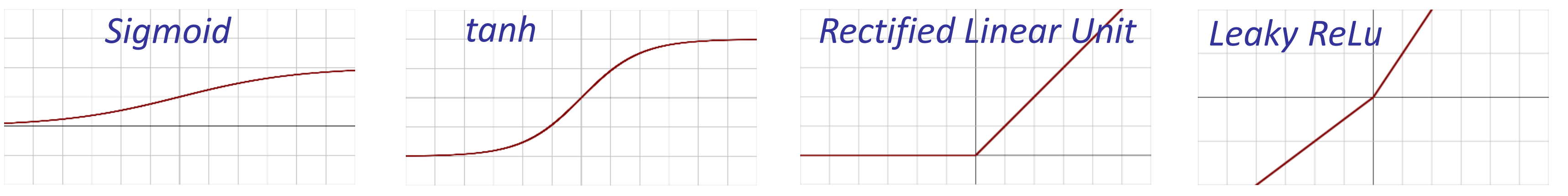
- Introduction to non-linear multi-scale simulations
  - FE multi-scale simulations
    - Problems to be solved at two scales
    - Require Newton-Raphson iterations at both scales
  - Use of surrogate models
    - Train a meso-scale surrogate model (off-line)
      - Requires extensive data
      - Obtained from RVE simulations
    - Use the trained surrogate model during analyses (on-line)
      - Surrogate acts as a homogenised constitutive law
      - Expected speed-up of several orders



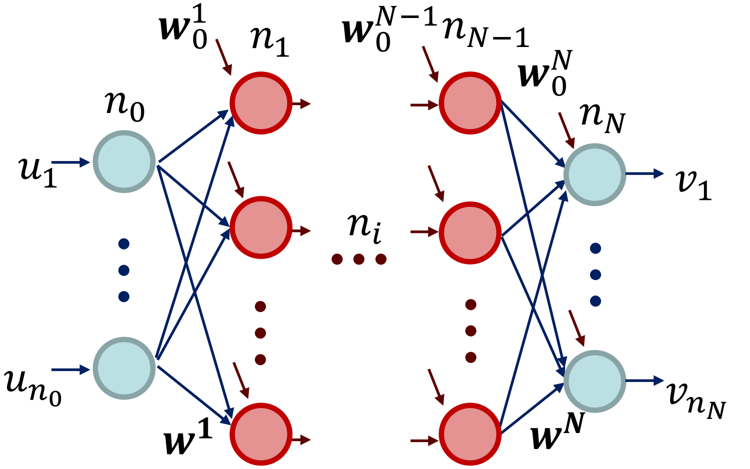
- Definition of the surrogate model
  - Artificial neuron
    - Non-linear function on n_0 inputs u_k
    - Requires evaluation of weights w_k
    - Requires definition of activation function f



  - Activation functions f

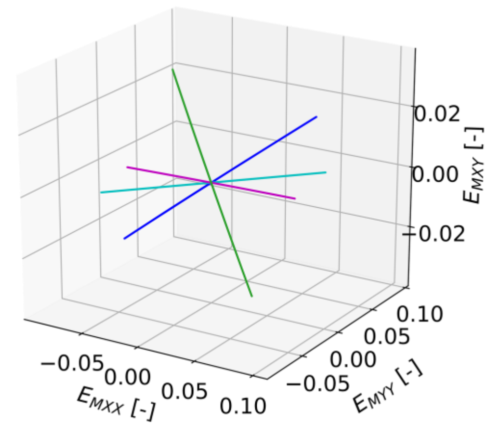


  - Feed-Forward Neuron Network
    - Simplest architecture
    - Layers of neurons
      - Input layer
      - $N-1$ hidden layers
      - Output layers
    - Mapping $\Re^{n_0} \rightarrow \Re^{n_N}: \boldsymbol{v} = \boldsymbol{g}(\boldsymbol{u})$

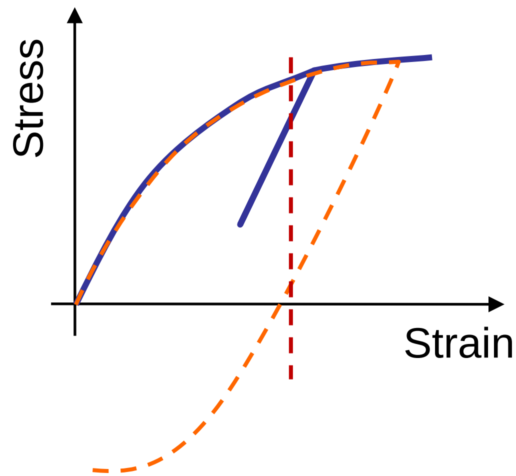


Recurrent Neural Network-accelerated multi-scale simulations

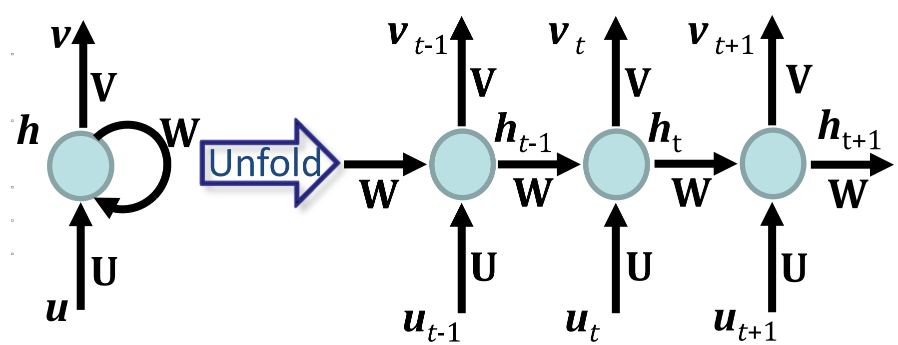
- Input / output definition
  - Input: Strain (history): $\mathbf{F}_{\mathrm{M}}$
  - Output: Stress (history): $\mathbf{P}_{\mathrm{M}}$



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

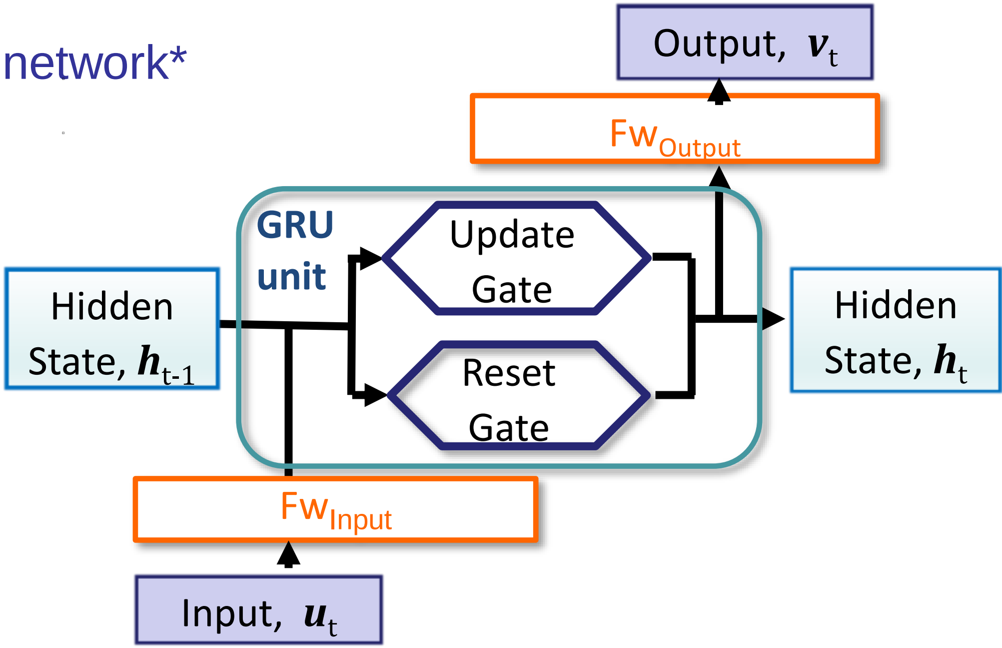


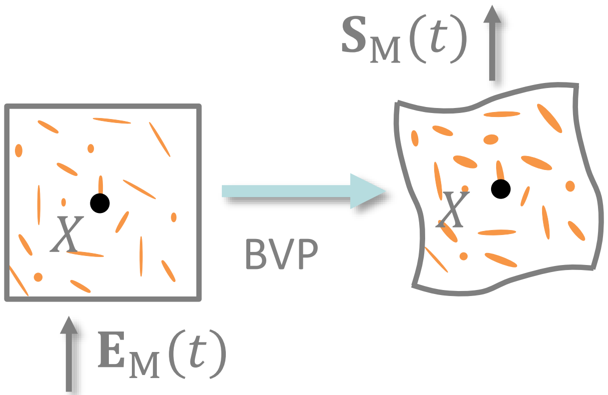
- Recurrent neural network
  - Allows a history dependent relation
    - Input $\boldsymbol{u}_t$
    - Output $\boldsymbol{v}_t = \boldsymbol{g}(\boldsymbol{u}_t, \boldsymbol{h}_{t-1})$
    - Internal variable $\boldsymbol{h}_t = \boldsymbol{g}(\boldsymbol{u}_t, \boldsymbol{h}_{t-1})$
  - Weights matrices $\mathbf{U}, \mathbf{W}, \mathbf{V}$
    - Trained using sequences
      - Inputs $\boldsymbol{u}_{t-n}^{(p)}, \ldots, \boldsymbol{u}_t^{(p)}$
      - Output $\boldsymbol{v}_{t-n}^{(p)}, \ldots, \boldsymbol{v}_t^{(p)}$



Recurrent Neural Network-accelerated multi-scale simulations

- Previous Work with recurrent neural network*
  - 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
    - $\mathrm{FW_{Input}}$ to treat inputs $\boldsymbol{u_t}$
    - $\mathrm{FW_{Output}}$ to produce outputs $\boldsymbol{v_t}$
  - Details
    - $\boldsymbol{u_t}$: homogenised GL strain $\mathbf{E}_\mathrm{M}$ (symmetric)
    - $\boldsymbol{v_t}$: homogenised 2nd PK stress $\mathbf{S}_\mathrm{M}$ (symmetric)
    - 100 hidden variables $\boldsymbol{h_t}$
    - $\mathrm{FW_{Input}}$ one hidden layer of 60 neurons
    - $\mathrm{FW_{Output}}$ two hidden layers of 100 neurons

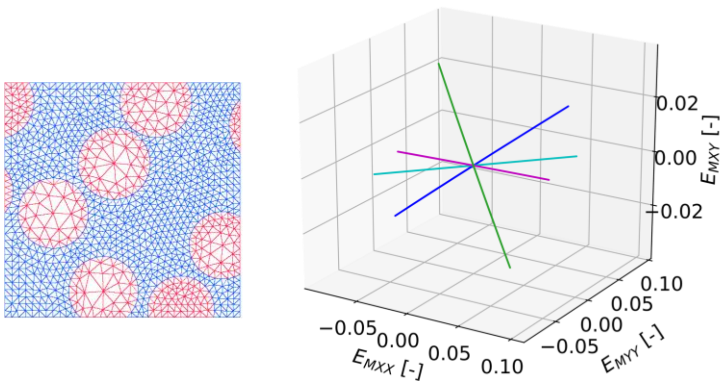


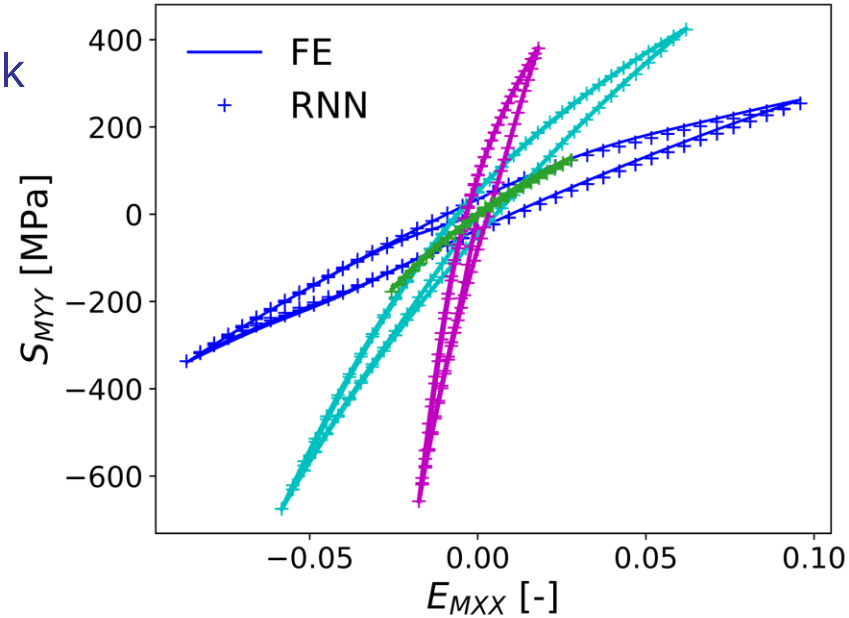


*L. Wu, V. D. Nguyen, N. G. Kilingar, and L. Noels. "A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths." Computer Methods in Applied Mechanics and Engineering, 369 (01 September 2020): 113234. doi:10.1016/j.cma.2020.113234

Recurrent Neural Network-accelerated multi-scale simulations

- Previous Work with recurrent neural network
  - Good accuracy on testing data



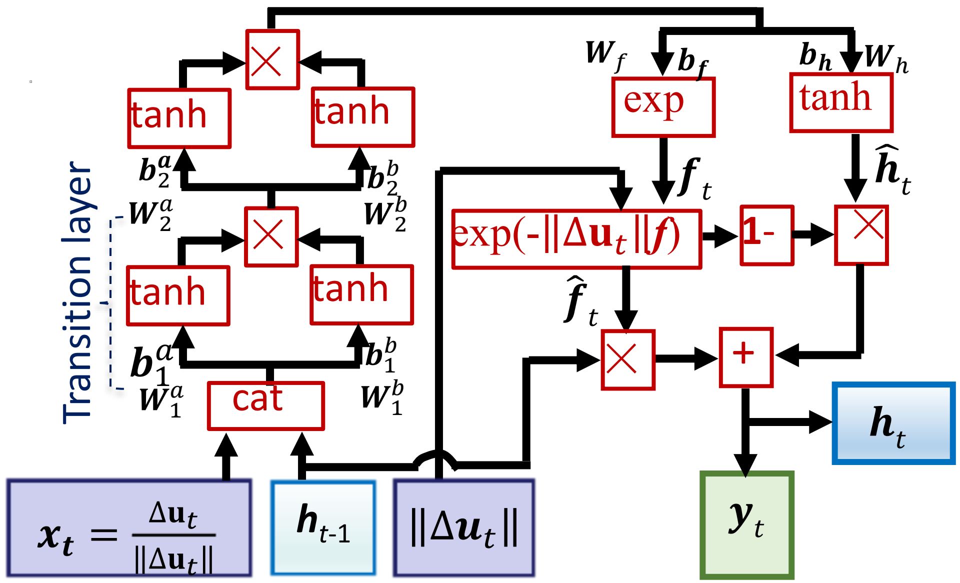


  - Sequence increment $\boldsymbol{\Delta u_t} = \Delta \mathbf{E}_{\mathrm{M}}$ of comparable order of magnitude between training and testing data

- 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

*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.

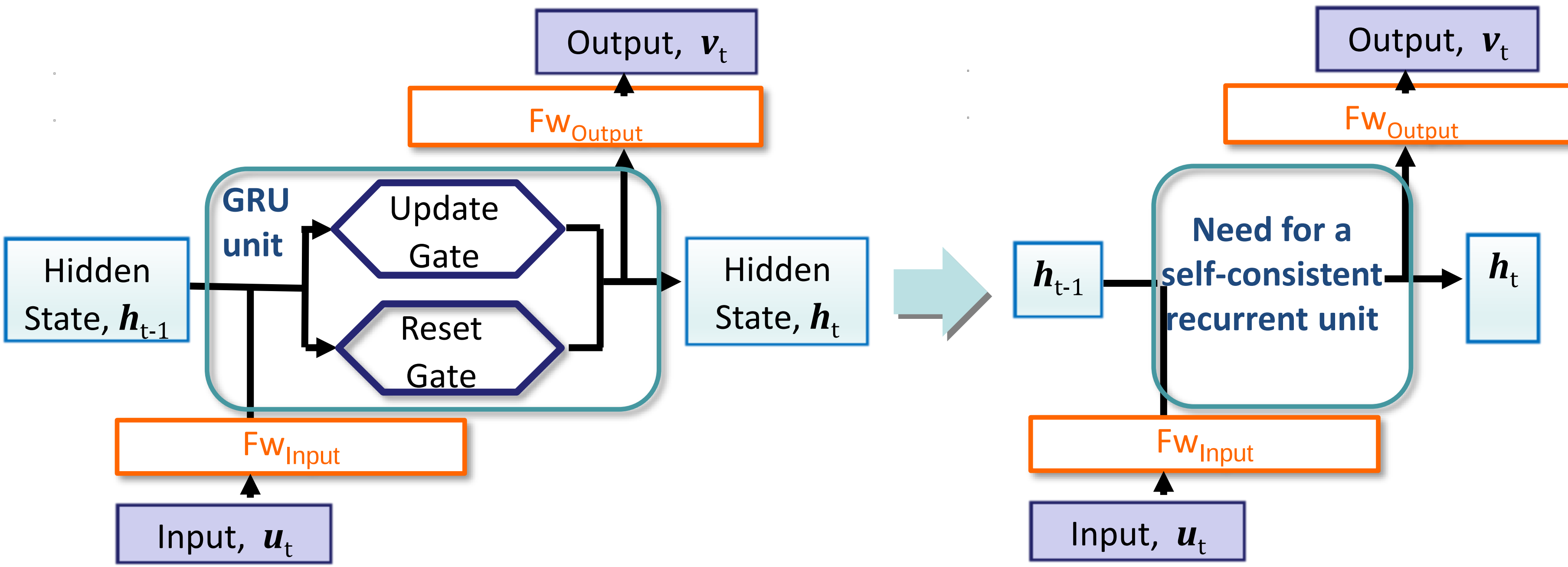
- Self-Consistent Linearized Minimal State Cell (SC-LMSC)*



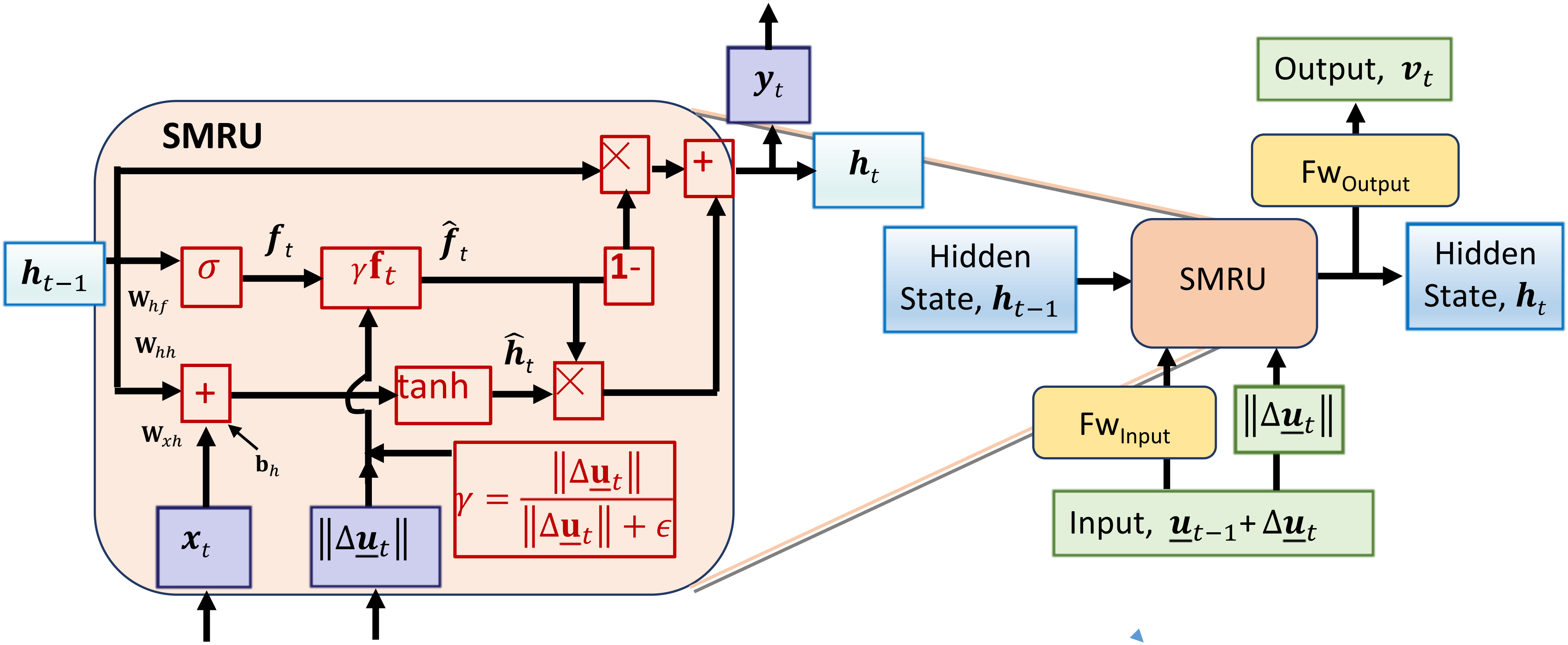
– Ingredients:
  - Incremental form of input variables converted to its direction $\frac{\Delta \boldsymbol{u}_t}{\|\Delta \boldsymbol{u}_t\|}$ and norm $\|\Delta \boldsymbol{u}_t\|$ *
  - Activations layers fed by direction $\frac{\Delta \boldsymbol{u}_t}{\|\Delta \boldsymbol{u}_t\|}$ and previous hidden variable direction $\boldsymbol{h}_t$
  - Double exponential activation function on output $\boldsymbol{O}_t$ of activation layers:
    – $\boldsymbol{f}_t = \exp\left[\boldsymbol{W}_f \boldsymbol{O}_t + \boldsymbol{b}_f\right] > 0$ and ratio $\hat{\boldsymbol{f}}_t = \exp[-\|\Delta \boldsymbol{u}_t\|\ \boldsymbol{f}_t] \in [0, 1]$
    – Hidden variables $\boldsymbol{h}_t$ are an element-wise interpolation (ratio $\hat{\boldsymbol{f}}_t$ dependent on the norm of $\|\Delta \boldsymbol{u}_t\|$) between previous value $\boldsymbol{h}_{t-1}$ and $\hat{\boldsymbol{h}}_t$

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



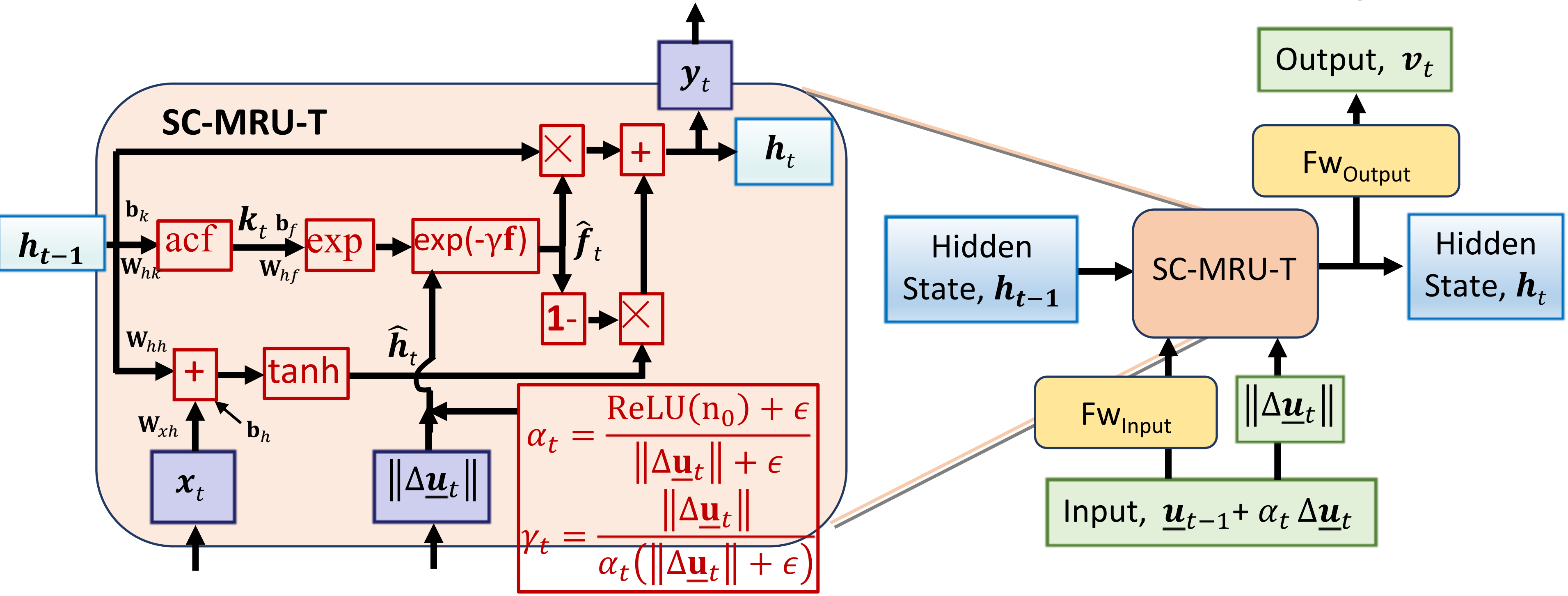
- New cell 1: New simplified recurrent unit: Simplified Minimal Recurrent Unit



- The **total form of input variable** as well as **increment norm** $\|\Delta \underline{u}_t\|$ (like SC-LMCS)
- **Self-consistency weakly enforced**
  - Using norm of $\|\Delta \underline{u}_t\|$ and
  - Data augmentation during training (i.e. subdividing randomly increments in training data)

Self-Consistent Recurrent Neural Network

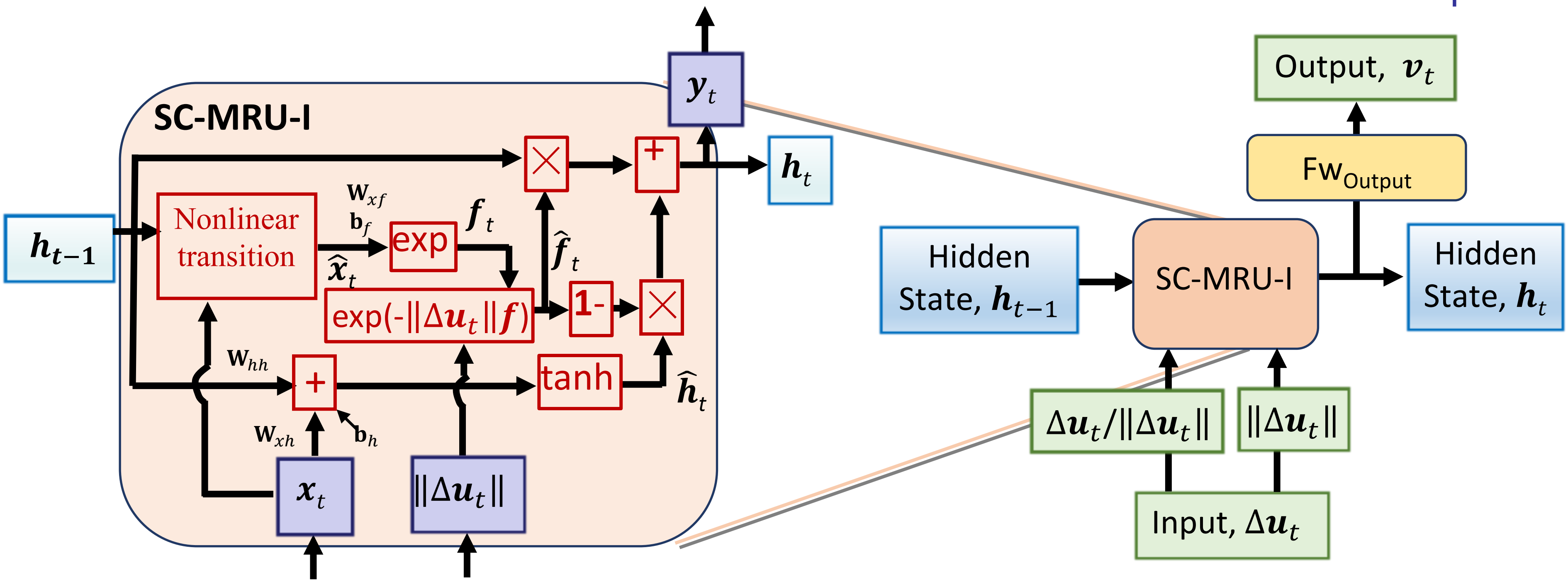
- New cell 2: Self-Consistent Minimal Recurrent Unit with Total form of inputs



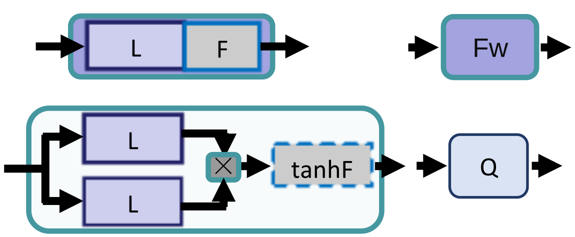
- The **total form of input variable** as well as **increment norm** $\|\Delta\boldsymbol{u}_t\|$ (like SC-LMCS)
  - Use as input $\underline{\boldsymbol{u}}_{t-1} + \alpha_t\,\Delta\underline{\boldsymbol{u}}_t$ (n_0 is a learnable parameter)
  - acf is the same activation function as in $\mathrm{Fw}_{\mathrm{input}}$
- **Self-consistency enforced**
  - Double exponential function $\boldsymbol{f}_t = \mathbf{exp}[\boldsymbol{W}_f\boldsymbol{k}_t + \boldsymbol{b}_f] > \mathbf{0}$ & ratio $\hat{\boldsymbol{f}}_t = \mathbf{exp}[-\gamma(\|\Delta\boldsymbol{u}_t\|)\,\boldsymbol{f}_t] \in [\mathbf{0},\mathbf{1}]$
  - Hidden variables $\boldsymbol{h}_t$ is an element-wise interpolation (ratio $\hat{\boldsymbol{f}}_t$ dependent on the norm of $\|\Delta\boldsymbol{u}_t\|$) between previous value $\boldsymbol{h}_{t-1}$ and $\hat{\boldsymbol{h}}_t$

Self-Consistent Recurrent Neural Network

- New cell 3: Self-Consistent Minimal Recurrent Unit with Incremental form of inputs



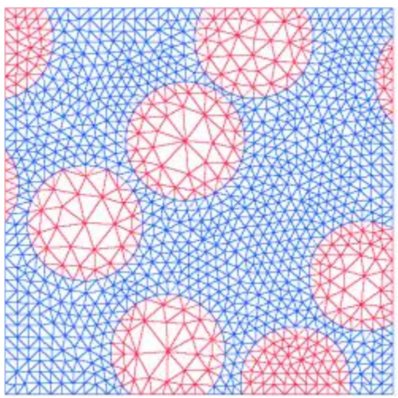
- The **incremental form of input variable** as well as **increment norm** $\|\Delta\boldsymbol{u}_t\|$ (like LMCS)
  - Use as input $\Delta\boldsymbol{u}_t/\|\Delta\boldsymbol{u}_t\|$ and $\|\Delta\boldsymbol{u}_t\|$
  - Non-linear transition blocks:

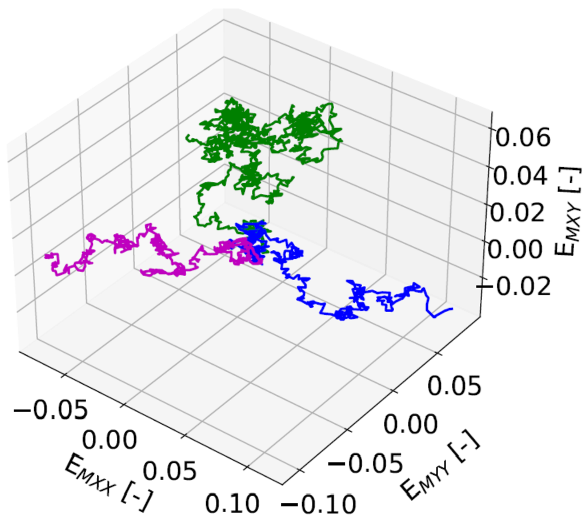


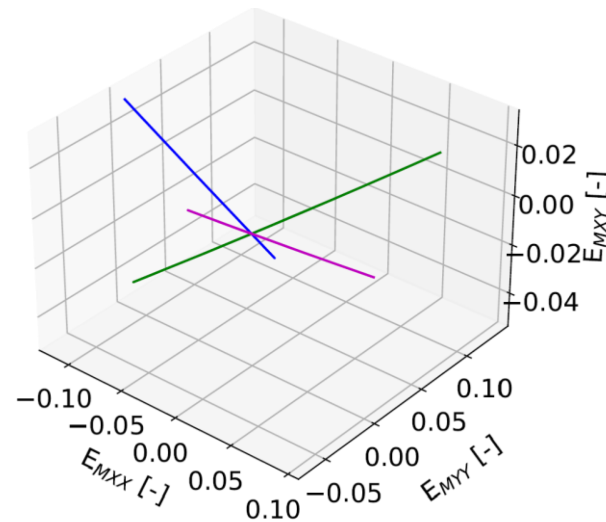
- **Self-consistency enforced**
  - Double exponential function $\boldsymbol{f}_t = \exp[\boldsymbol{W}_{xf}\widehat{\boldsymbol{x}}_t + \boldsymbol{b}_f] > \mathbf{0}$ & ratio $\widehat{\boldsymbol{f}}_t = \exp[-(\|\Delta\boldsymbol{u}_t\|)\,\boldsymbol{f}_t] \in [\mathbf{0}, \mathbf{1}]$
  - Hidden variables $\boldsymbol{h}_t$ is an element-wise interpolation (ratio $\widehat{\boldsymbol{f}}_t$) between previous value $\boldsymbol{h}_{t-1}$ and $\widehat{\boldsymbol{h}}_t$

Training of Recurrent Neural Network

- Training strategy
  - Elasto-plastic composite RVE





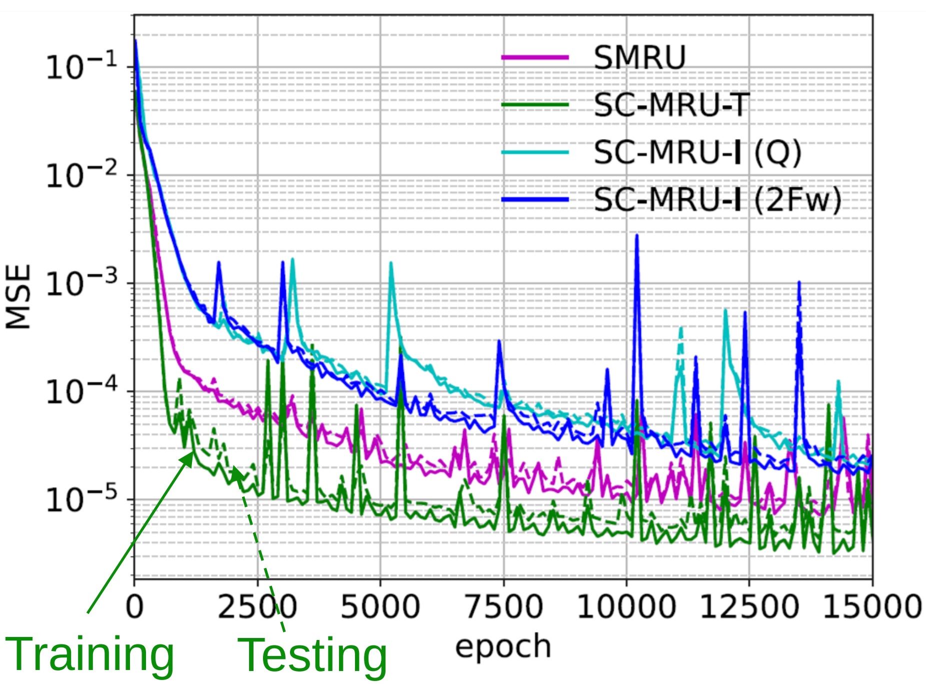


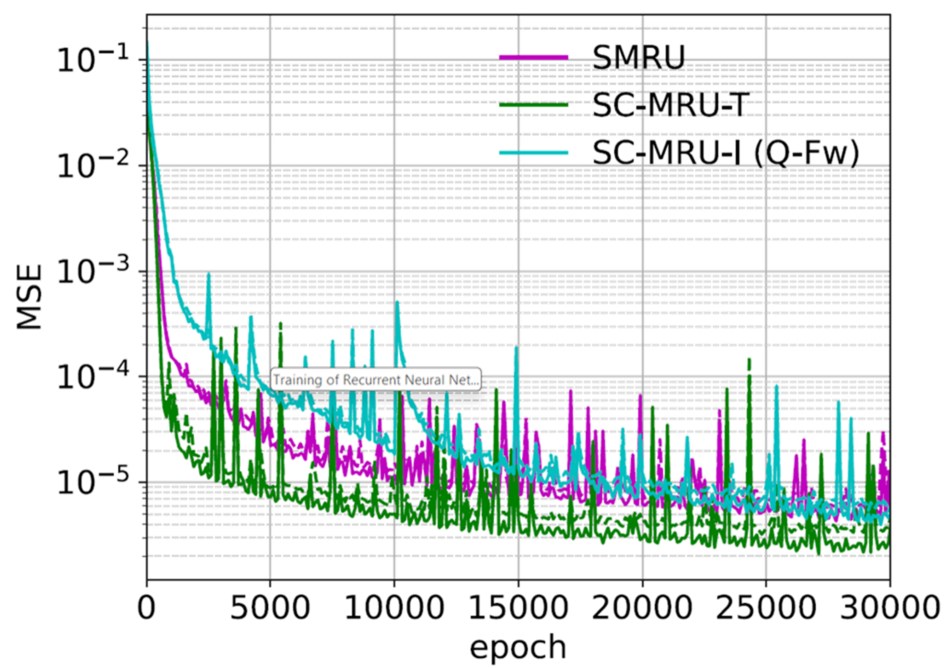
  - Training data
    - Should cover full range of possible loading histories
    - Use random walking strategy
    - Completed with random cyclic loading
    - Bounded by a hypercube of 12% deformation

Training of Recurrent Neural Network

- Training stage
  - Learnable parameters for 120 hidden variables

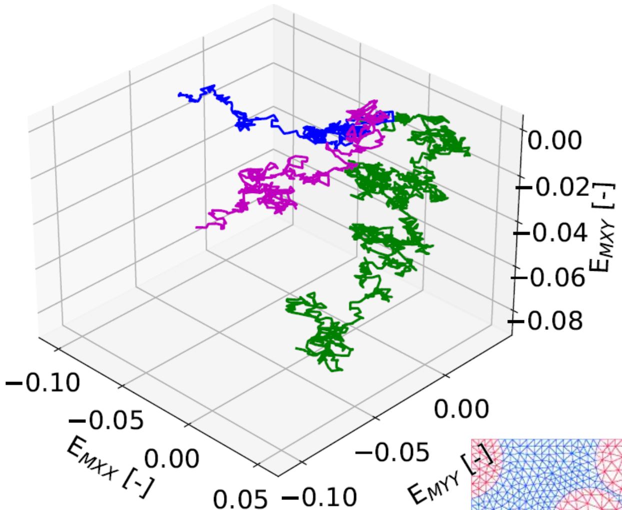
Recurrent unit	SMRU	SC-MRU-T	SC-MRU-I		
Transition block	-	-	Q	Fw-Fw	Q-Fw
Learnable parameters	44 284	58 925	59 644	59 284	74 164



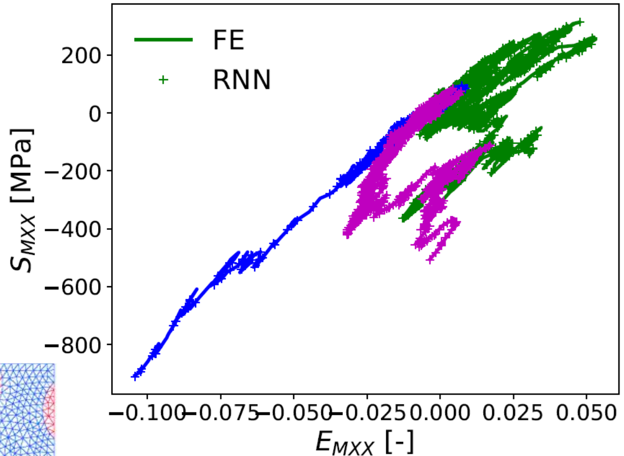


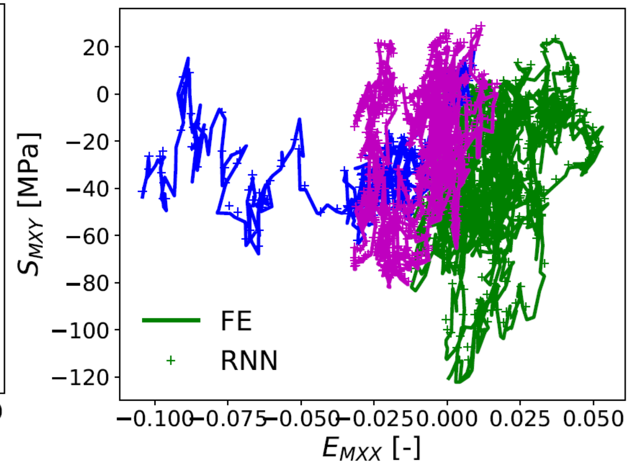
Testing of Recurrent Neural Network

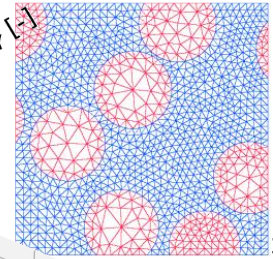
- SC-MRU-T: Testing data with inserted extra-points

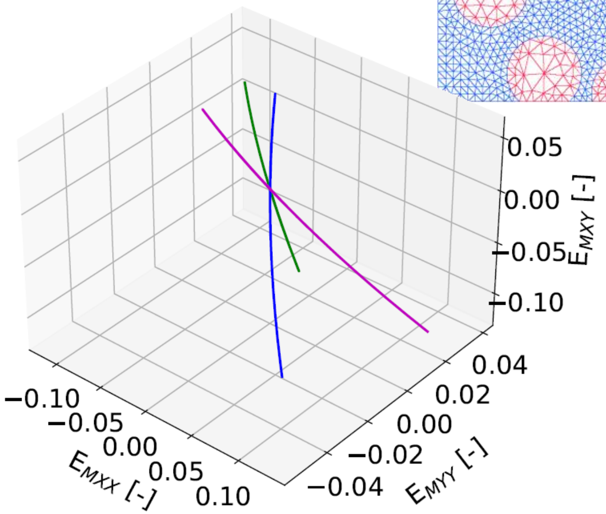


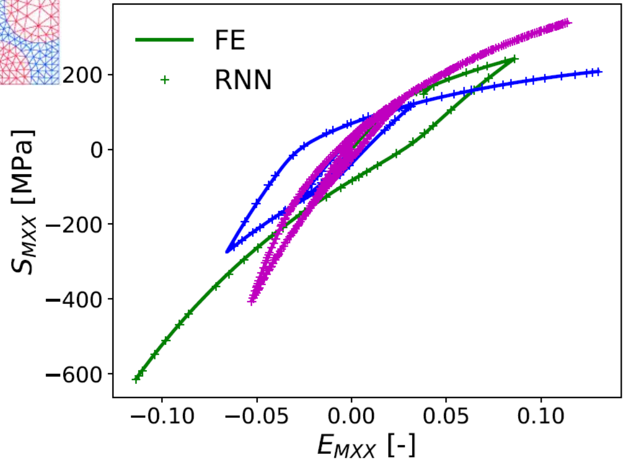
Maximum inserted data points is 5, 20, 60 in each loading step

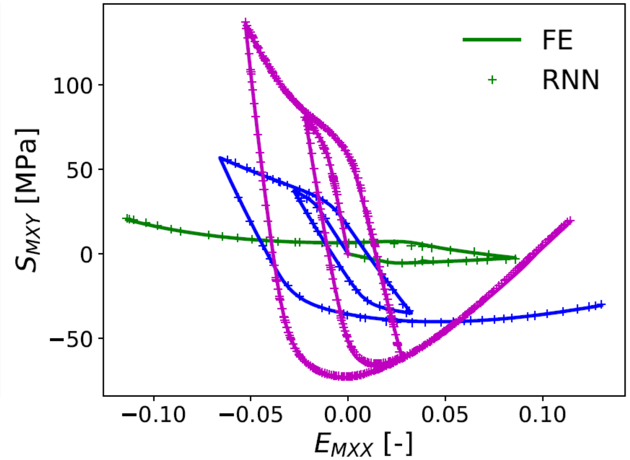




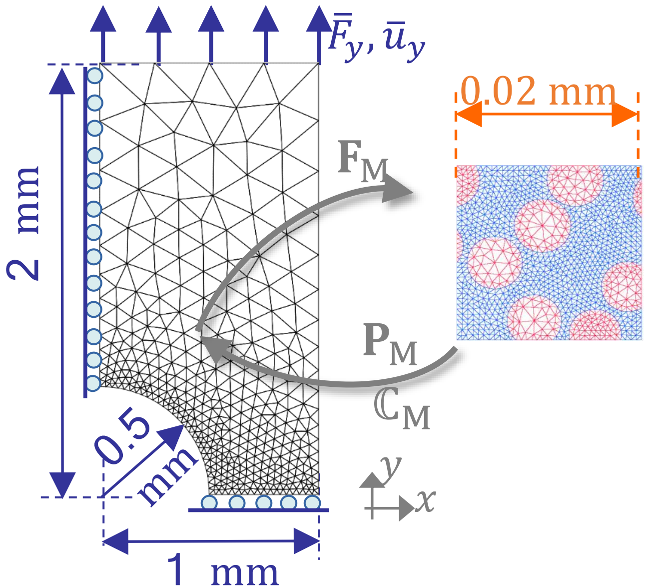


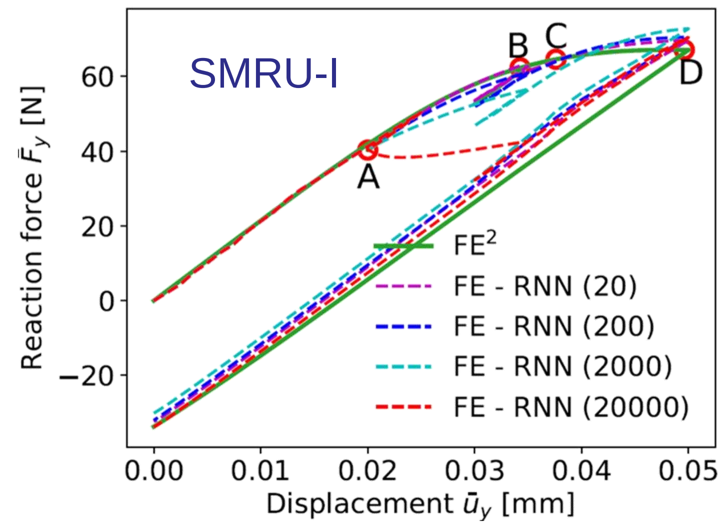


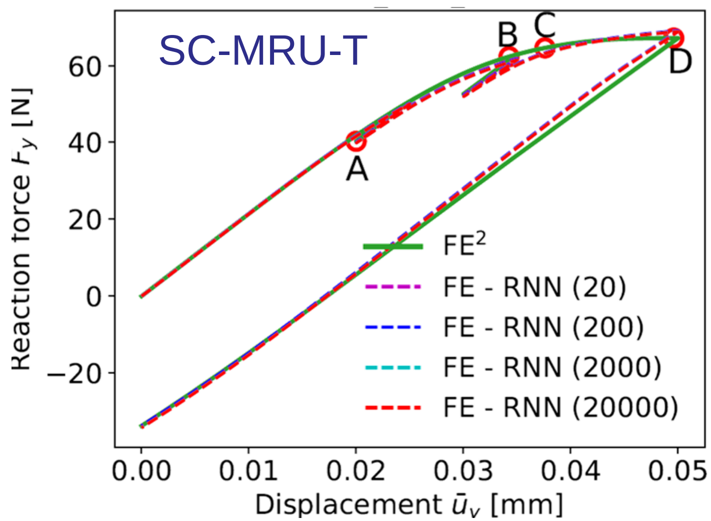


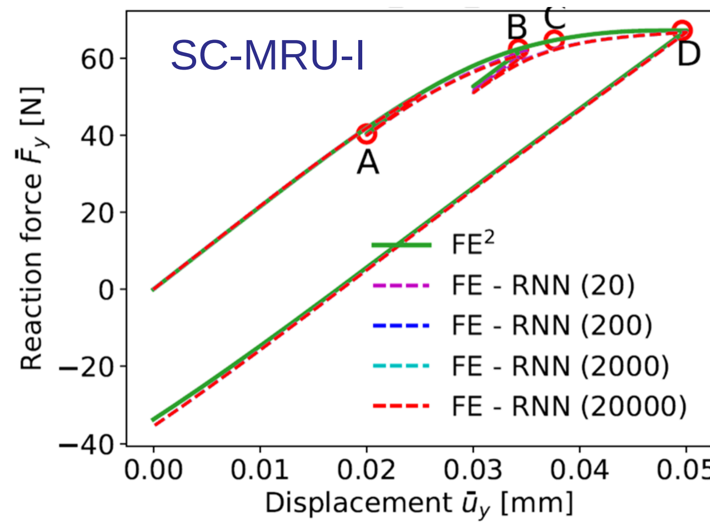


- FE2 vs. FE-RNN: Change in the increment size (between points A&B)



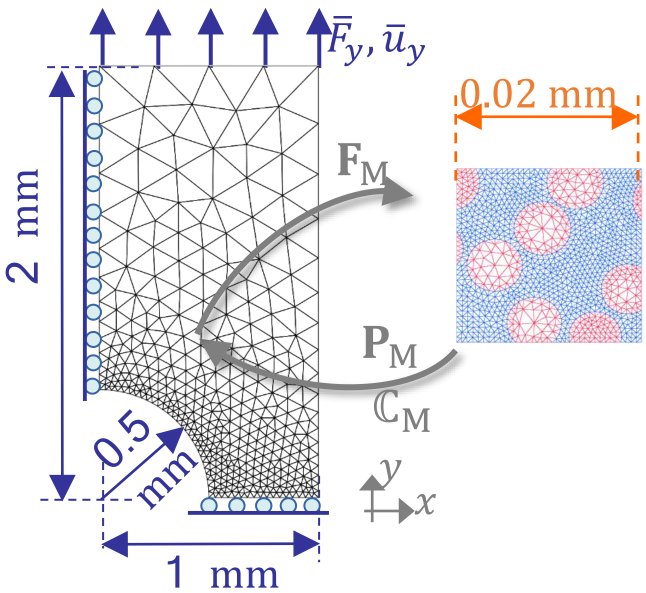






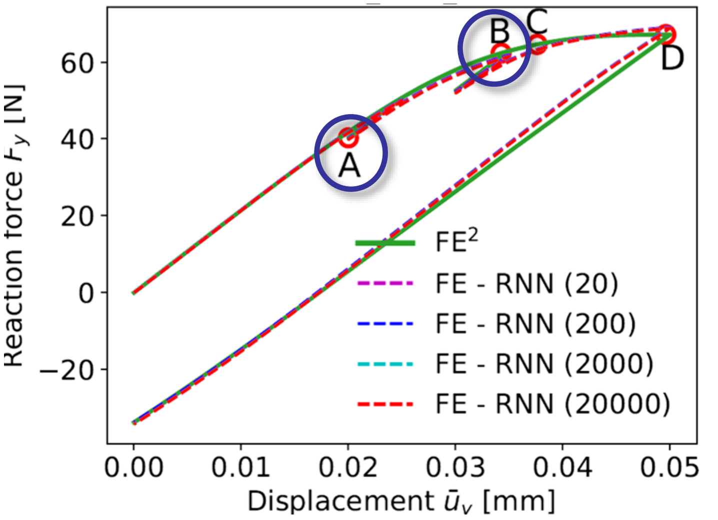
Multi-scale simulations with Recurrent Neural Network

- FE2 vs. FE-RNN: Cost comparison

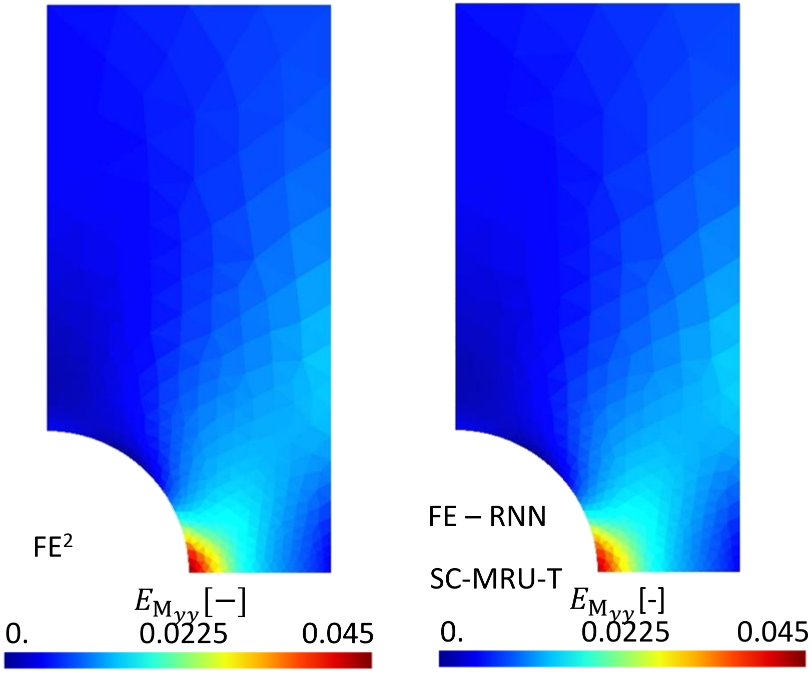


Off-line	FE^2	SMRU	SC-MRU-T	SC-MRU-I
Data generation	-	23500 h-cpu		
Training	-	< 10 h-cpu		
On-line	**FE^2**	**SMRU**	**SC-MRU-T**	**SC-MRU-I**
Simulation	18000 h-cpu	0.27 h-cpu	0.38 h-cpu	0.28 h-cpu

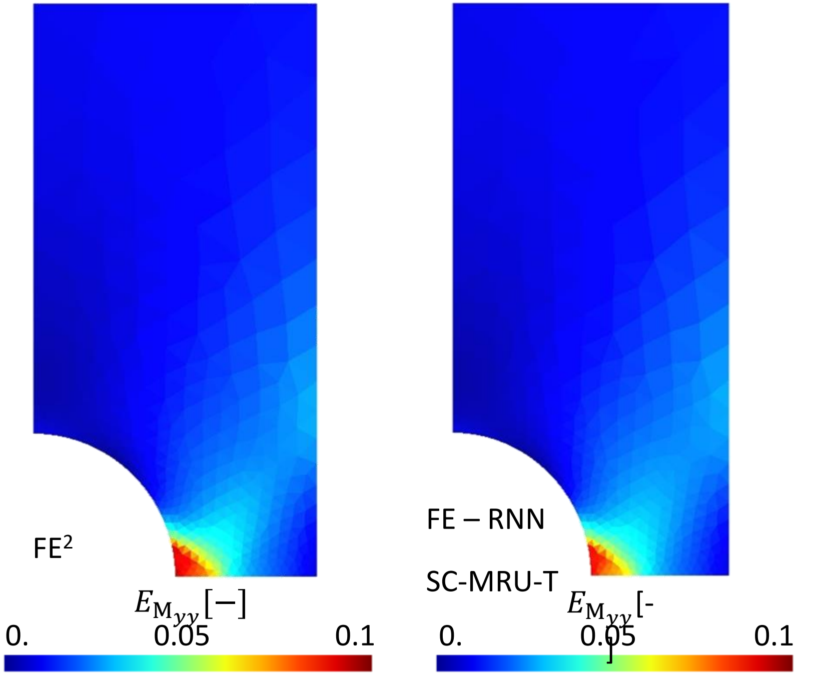
- FE2 vs. FE-RNN: Fields distribution



Point A

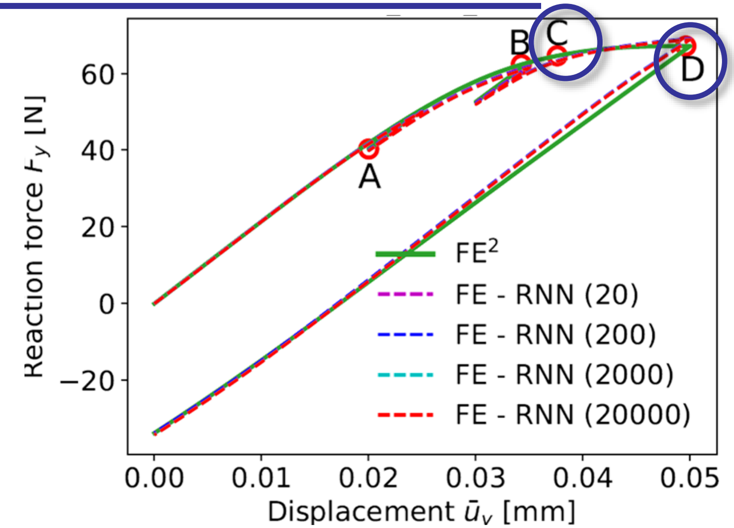


Point B

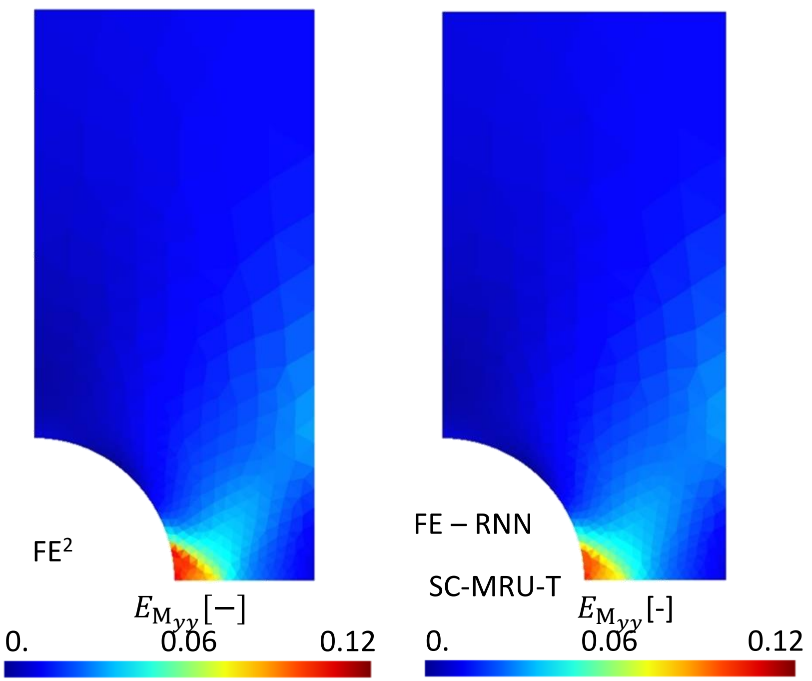


Multi-scale simulations with Recurrent Neural Network

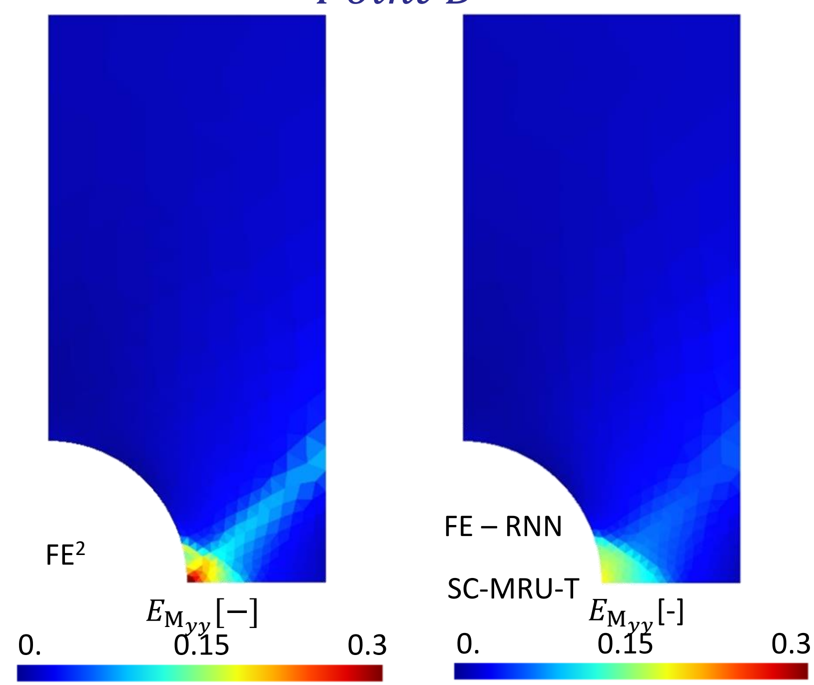
- FE2 vs. FE-RNN: Fields distribution



Point C



Point D



References

- Publication
  - L. Wu, L. Noels, Self-consistency Reinforced minimal Gated Recurrent Unit for surrogate modeling of history-dependent non-linear problems: Application to history-dependent homogenized response of heterogeneous materials, *Computer Methods in Applied Mechanics and Engineering* 424 (2024) 116881, doi: https://doi.org/10.1016/j.cma.2024.116881
- Data and code on
  - Repository: https://gitlab.uliege.be/didearot/didearotPublic/publicationsData/2024_scmru
  - Doi: http://dx.doi.org/10.5281/zenodo.10551272