Accelerating Simulations of Materials using Graph Neural Networks with Embedded Constitutive Models

Graphs&Data seminar

J. Storm, I. B. C. M. Rocha and F. P. van der Meer

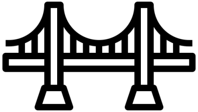
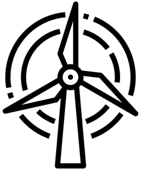
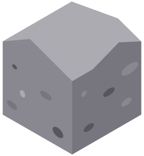
Delft University of Technology

TU Delft

SLIM MLAB LABS

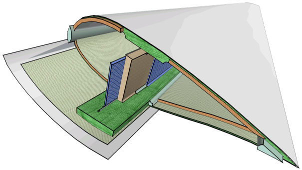
April 3, 2025

Computational models of structures are used everywhere

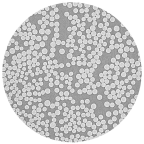
Design-specific complex materials can lead to more efficient structures

Modeling complex material behavior requires a multiscale approach

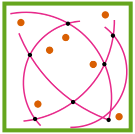


Macroscale

Mesoscale



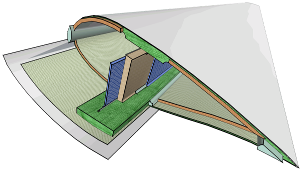
Microscale



Nanoscale

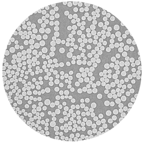
- Each scale depends on a lower scale

Modeling complex material behavior requires a multiscale approach

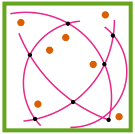


Macroscale

Mesoscale



Microscale

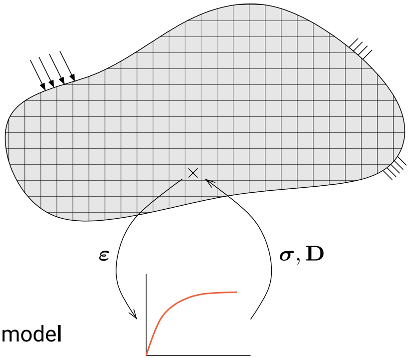


Nanoscale

- Each scale depends on a lower scale
- Ideally, we could optimize lower scale properties for macroscale performance

Single scale

Macro (Ω)



Constitutive model

Finite Element (FE) method

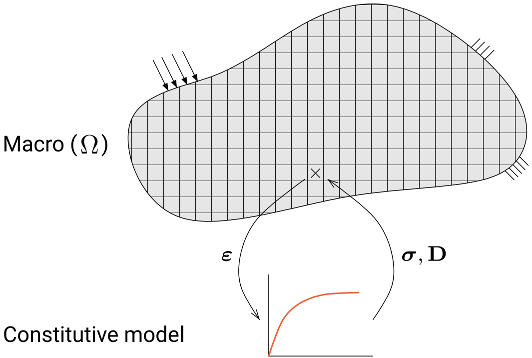
Domain subject to boundary conditions

- Displacements and forces

Find equilibrium

- $\nabla \cdot \boldsymbol{\sigma}^{\Omega} = \mathbf{0}$
- Relate strains ($\boldsymbol{\varepsilon}$) - stresses ($\boldsymbol{\sigma}$)
- Material dependent

Single scale



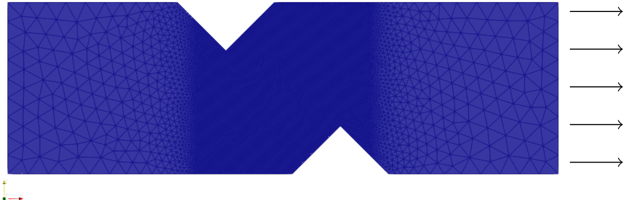
Finite Element (FE) method

Domain subject to boundary conditions

- Displacements and forces

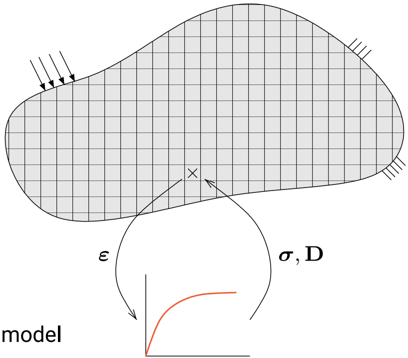
Find equilibrium

- $\nabla \cdot \sigma^{\Omega} = \mathbf{0}$
- Relate strains (ε) - stresses (σ)
- Material dependent



Single scale

Macro (Ω)



Constitutive model

Finite Element (FE) method

Domain subject to boundary conditions

- Displacements and forces

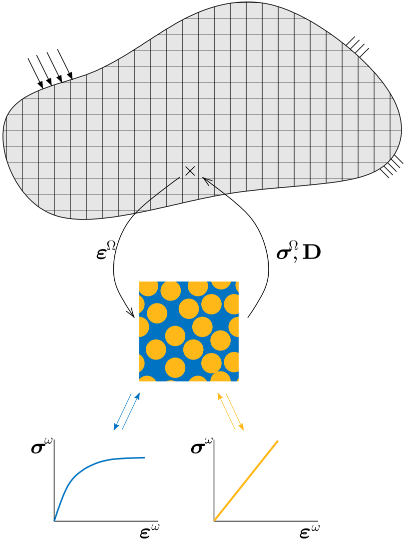
Find equilibrium

- $\nabla \cdot \boldsymbol{\sigma}^{\Omega} = \mathbf{0}$
- Relate strains ($\boldsymbol{\varepsilon}$) - stresses ($\boldsymbol{\sigma}$)
- Material dependent

Multiscale - FE^2

Macro (Ω)

Micro (ω)



Two-scale coupling: FE^2

- Macroscopic strain is microscopic boundary conditions
- Find microscopic equilibrium

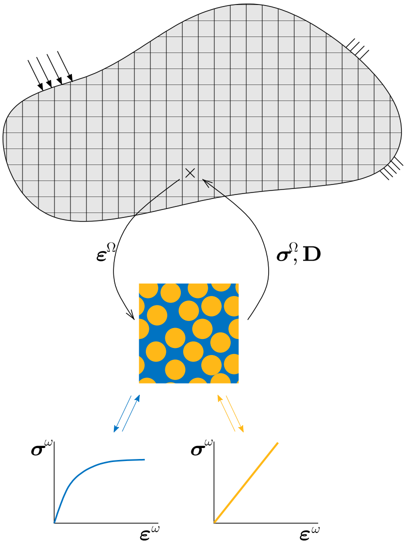
Homogenization

- Obtain microscopic stress field
- Average stress $\rightarrow \boldsymbol{\sigma}^\Omega$

Large promise for increased accuracy

Multiscale - FE^2

Macro (Ω)



Micro (ω)

Two-scale coupling: FE^2

- Macroscopic strain is microscopic boundary conditions
- Find microscopic equilibrium

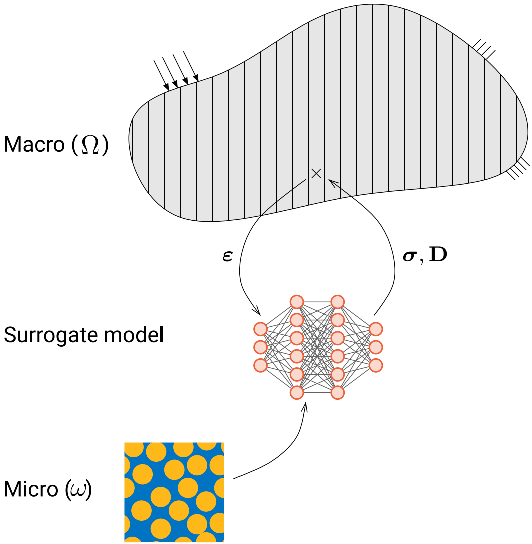
Homogenization

- Obtain microscopic stress field
- Average stress $\rightarrow \sigma^{\Omega}$

Large promise for increased accuracy

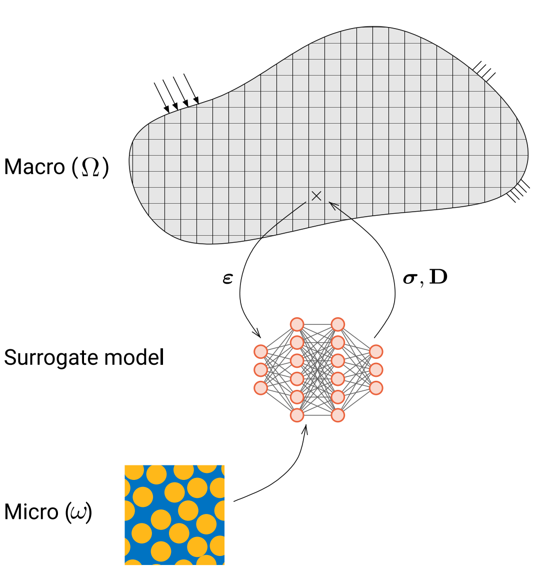
However: computationally expensive

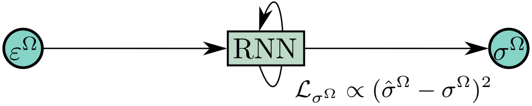
Surrogate modeling



- Data-driven model
- Trained on microscale simulations
- Much faster to evaluate

Surrogate modeling





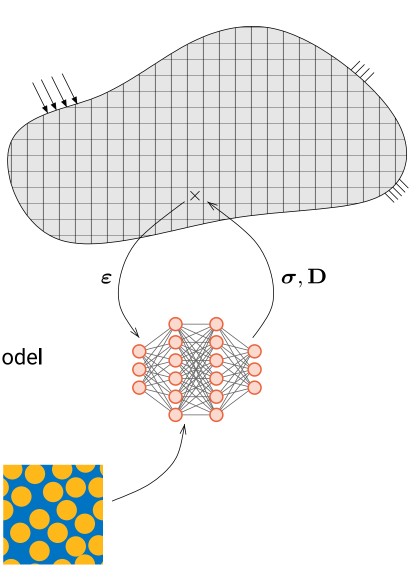
- Data-driven model
- Trained on microscale simulations
- Much faster to evaluate

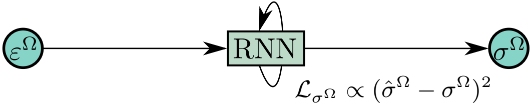
Surrogate modeling

Macro (Ω)

Surrogate model

Micro (ω)



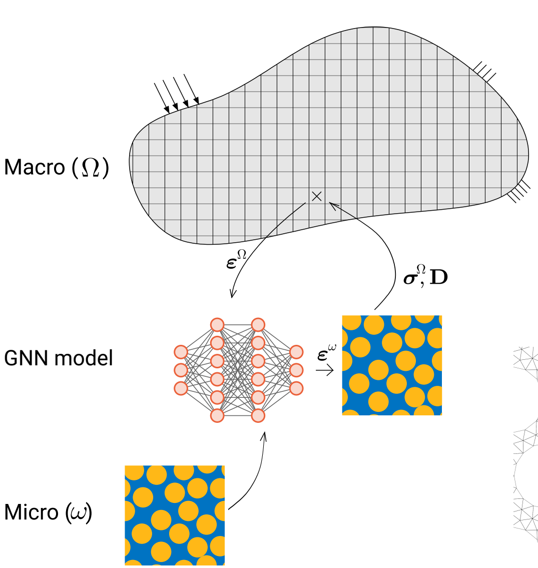


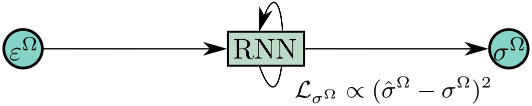
- Data-driven model
- Trained on microscale simulations
- Much faster to evaluate

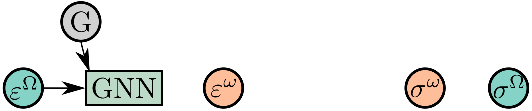
Back to single-scale. Lose:

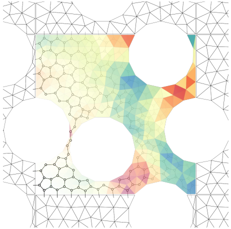
- microscale geometry
- microscale full-field solution
- ability to switch back

Graph neural network (GNN) approach







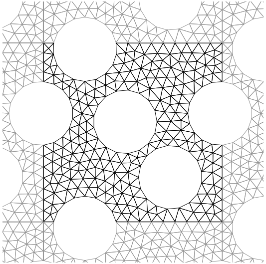


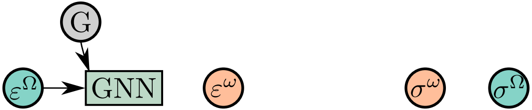
From mesh to graph

Mesh:

- Nodes with forces, displacements and boundary conditions
- Integration points with stresses, strains, and the material model

Mesh $\rightarrow$ Dual graph

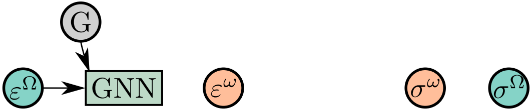




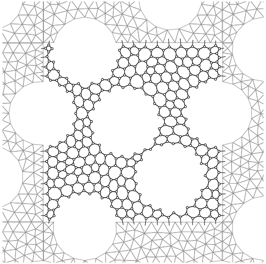
From mesh to graph

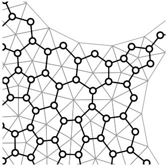
Mesh:

- Nodes with forces, displacements and boundary conditions
- Integration points with stresses, strains, and the material model



Mesh $\rightarrow$ Dual graph



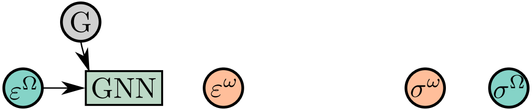


From mesh to graph

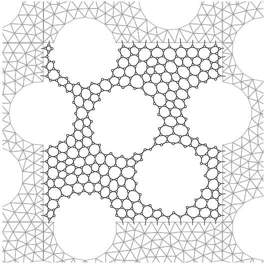
Mesh:

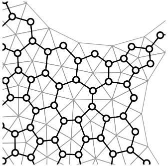
- Nodes with forces, displacements and boundary conditions
- Integration points with stresses, strains, and the material model

No dynamic behavior that spreads over time



Mesh $\rightarrow$ Dual graph





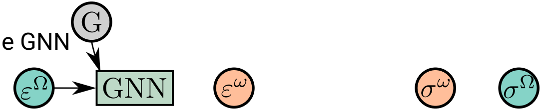
From mesh to graph

Mesh:

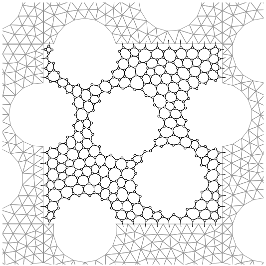
- Nodes with forces, displacements and boundary conditions
- Integration points with stresses, strains, and the material model

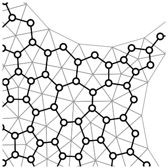
No dynamic behavior that spreads over time

Boundary condition: pass ε^{Ω} directly as features to the GNN



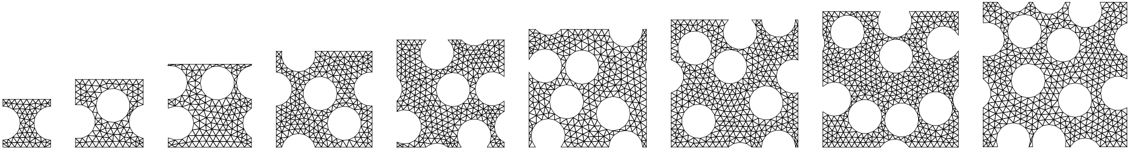
Mesh $\rightarrow$ Dual graph





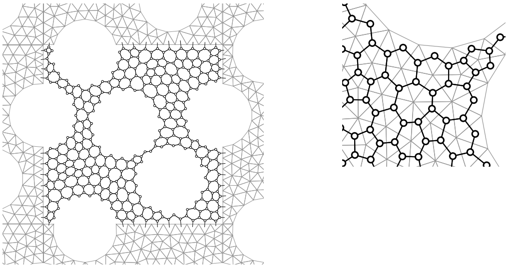
From mesh to graph

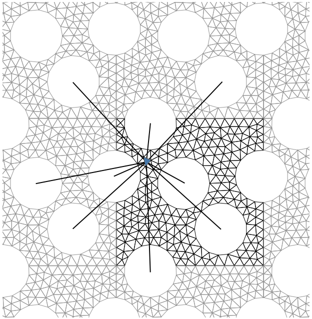
Variable microstructure & element size



G feature: Δx & Δy to voids

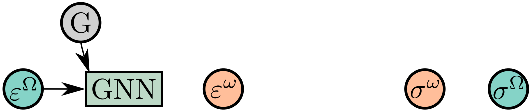
Mesh $\rightarrow$ Dual graph





Inputs and outputs

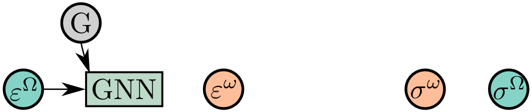
ε^ω, σ^ω and σ^Ω are all relevant

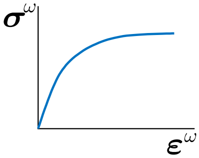


Inputs and outputs

ε^ω, σ^ω and σ^Ω are all relevant

We know the $\varepsilon^\omega \to \sigma^\omega$ relation

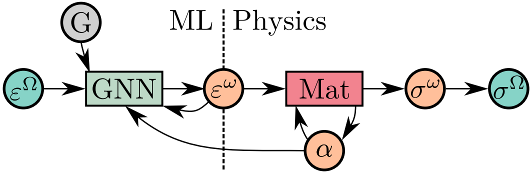




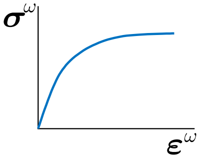
Inputs and outputs

ε^ω, σ^ω and σ^Ω are all relevant

We know the $\varepsilon^\omega \rightarrow \sigma^\omega$ relation



$$\mathcal{L}_{\varepsilon^\omega} \propto (\hat{\varepsilon}^\omega - \varepsilon^\omega)^2$$
$$\mathcal{L}_{\sigma^\omega} \propto (\hat{\sigma}^\omega - \sigma^\omega)^2$$
$$\mathcal{L}_{\sigma^\Omega} \propto (\hat{\sigma}^\Omega - \sigma^\Omega)^2$$

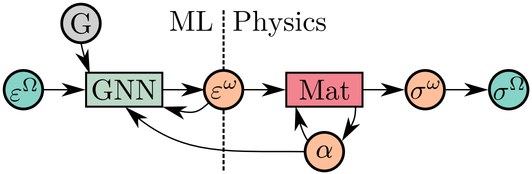


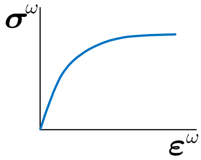
Inputs and outputs

ε^ω, σ^ω and σ^Ω are all relevant

We know the $\varepsilon^\omega \to \sigma^\omega$ relation

- σ^ω fully depend on ε^ω, but including both in the loss function helps
- σ^Ω is an averaged quantity, including it in the loss reduces σ^ω performance

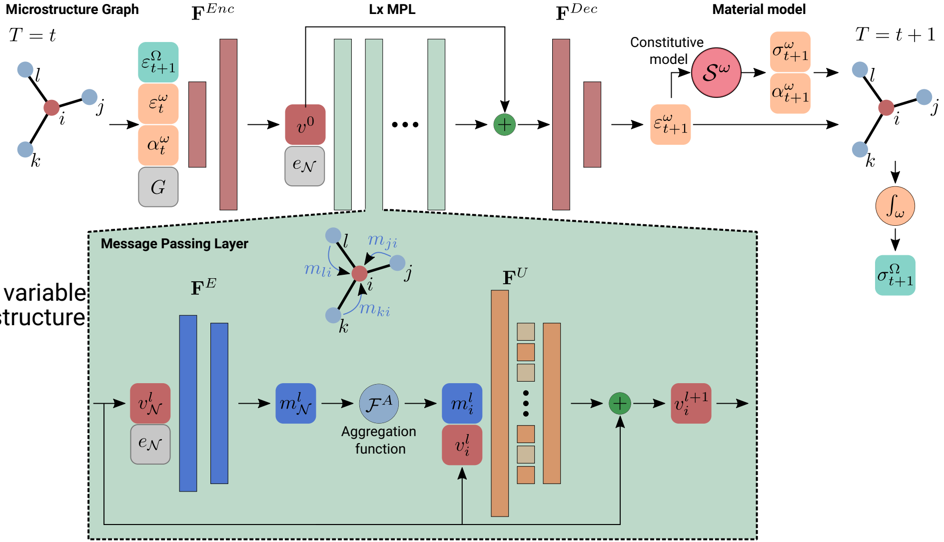




$$\mathcal{L}_{\varepsilon^\omega} \propto (\hat{\varepsilon}^\omega - \varepsilon^\omega)^2$$
$$\mathcal{L}_{\sigma^\omega} \propto (\hat{\sigma}^\omega - \sigma^\omega)^2$$
$$\mathcal{L}_{\sigma^\Omega} \propto (\hat{\sigma}^\Omega - \sigma^\Omega)^2$$

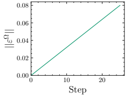
Network architecture

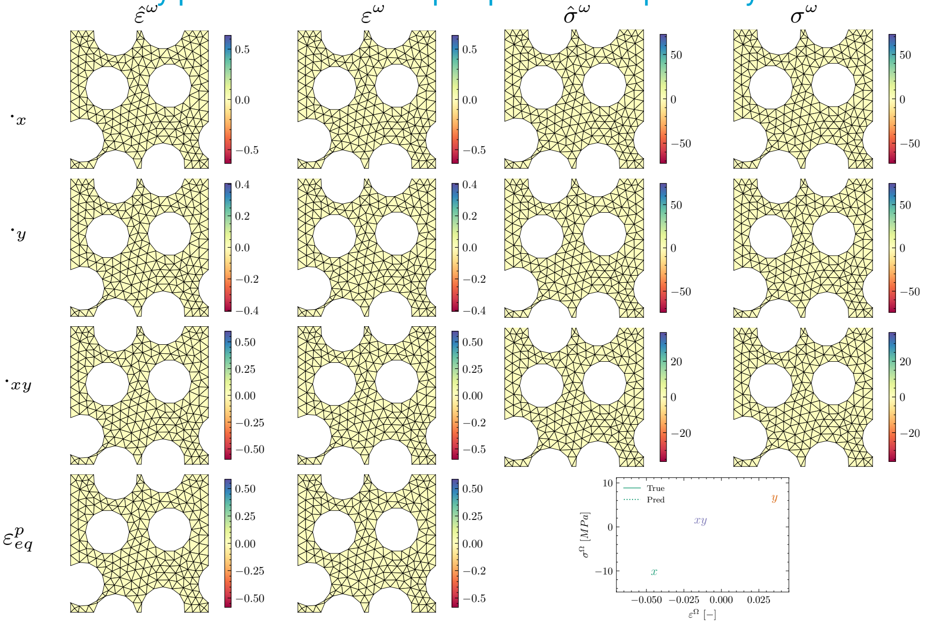
Encode → Process → Decode → Material model



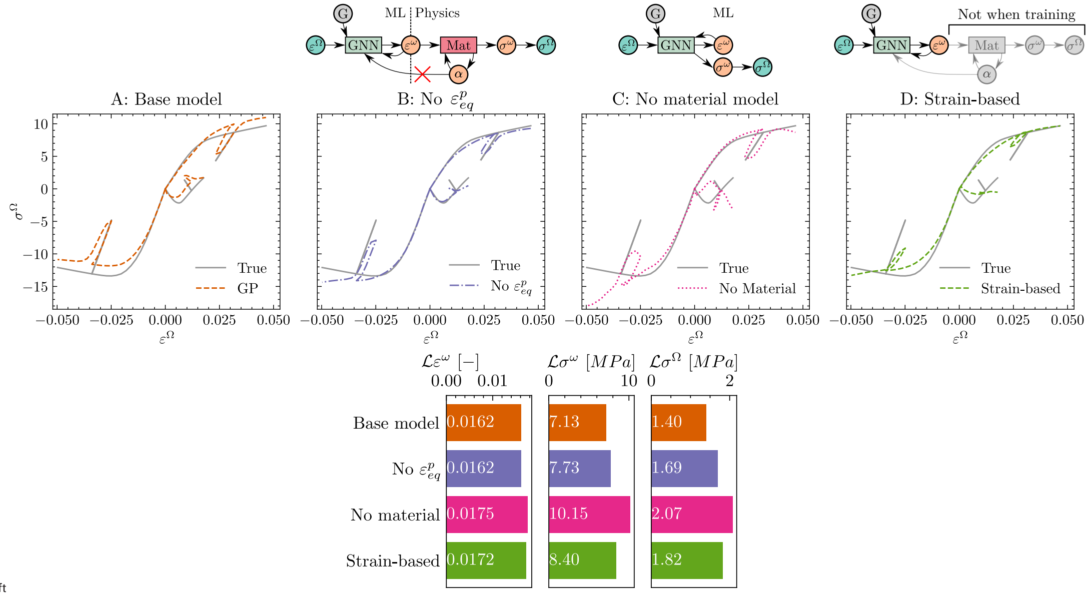
α : Internal variable
G : Graph structure

We can accurately predict all microscopic quantities in plasticity



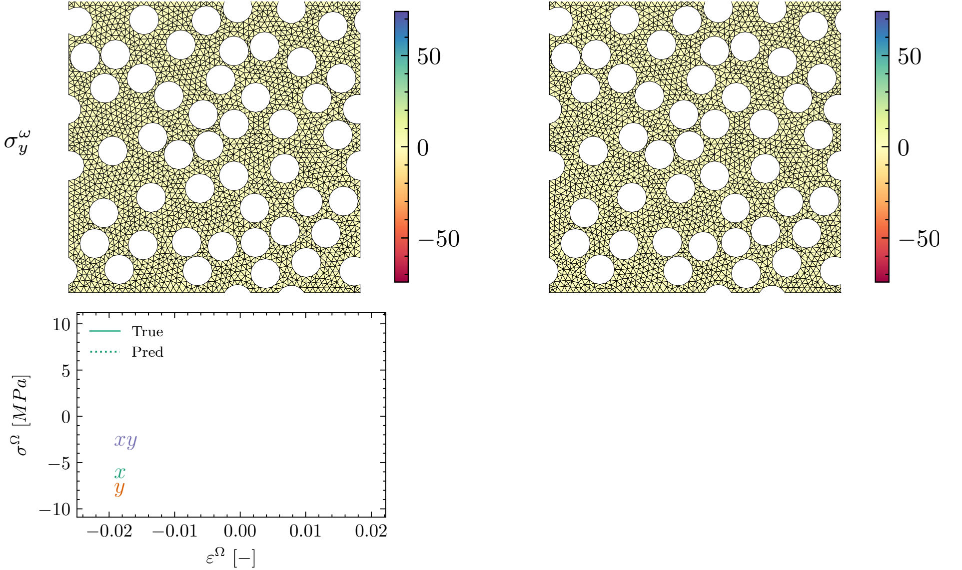


The material model is beneficial for prediction accuracy



Model can predict larger microstructures without modifications

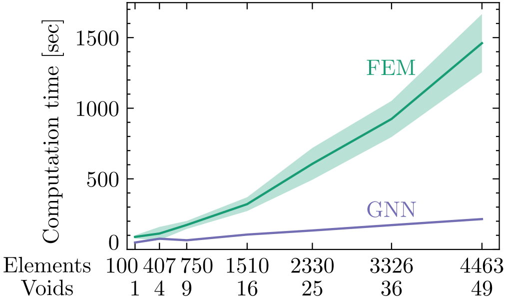
Testing a 49-void microstructure (training samples contain 1 to 9 voids)



GNN can accalerate multiscale simulations

Different coding language and hardware

- Only compare relative scaling, not absolute values



Conclusion

Graph Neural Network surrogate for multiscale simulations

TU Delft

Conclusion

Graph Neural Network surrogate for multiscale simulations

- Carefully choose how to define the graph and process the information
- Can train on small cheap microstructures and extrapolate to larger ones
- Embedding physical material model increases prediction accuracy
- We retain all microscopic quantities

Conclusion

Graph Neural Network surrogate for multiscale simulations

- Carefully choose how to define the graph and process the information
- Can train on small cheap microstructures and extrapolate to larger ones
- Embedding physical material model increases prediction accuracy
- We retain all microscopic quantities

tinyurl.com/MicroGNN

[Storm, Rocha, and van der Meer (2024), *CMAME*, 427:**117001**]

Conclusion

Graph Neural Network surrogate for multiscale simulations

- Carefully choose how to define the graph and process the information
- Can train on small cheap microstructures and extrapolate to larger ones
- Embedding physical material model increases prediction accuracy
- We retain all microscopic quantities

Thank you for your attention

tinyurl.com/MicroGNN

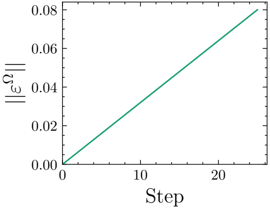
[Storm, Rocha, and van der Meer (2024), *CMAME*, 427:**117001**]

Different training data generation strategies

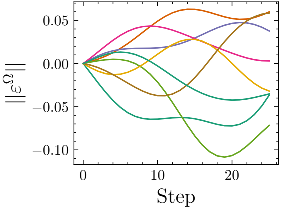
We need artificial load paths to generate training data

Three cases for the load magnitude

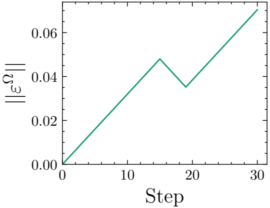
Monotonic loading



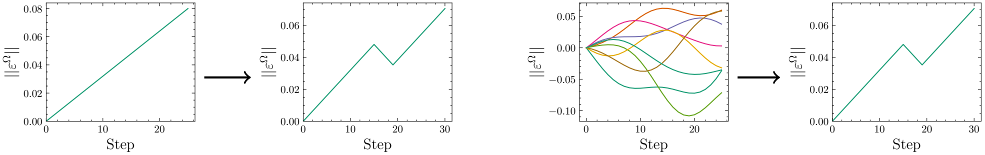
GP loading

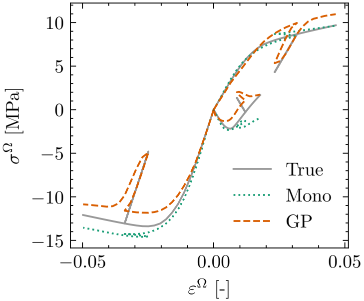


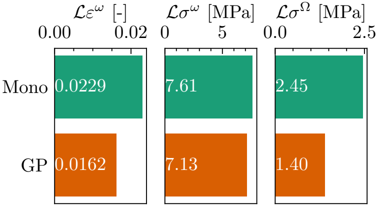
Unloading



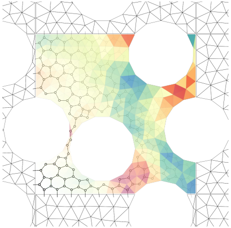
Gaussian process (GP) based training dataset improves generalization

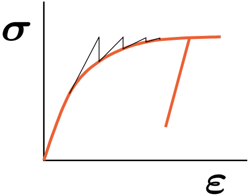


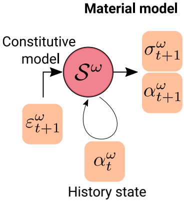




Embedding the material model is expensive





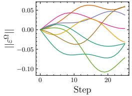


Minimization problem: Newton-Raphson solve

- For every element in mesh
- Inside training loop $\rightarrow$ backpropagate error back

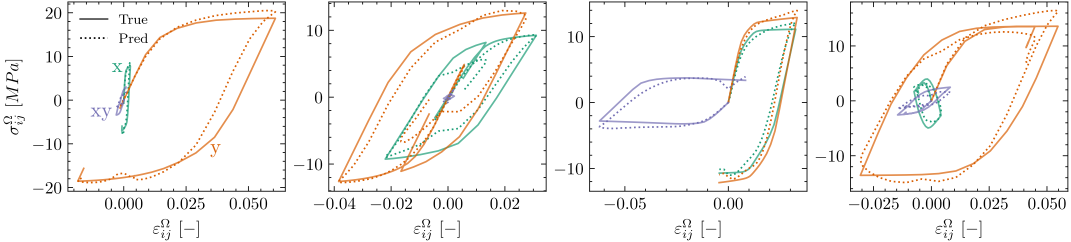
Internal history tracking $\rightarrow$ timesteps depend on all previous ones.

Extrapolating in timesteps is stable

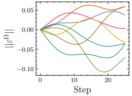


Trained for 25 timesteps → tested for 50

Autoregressive model: errors can accumulate

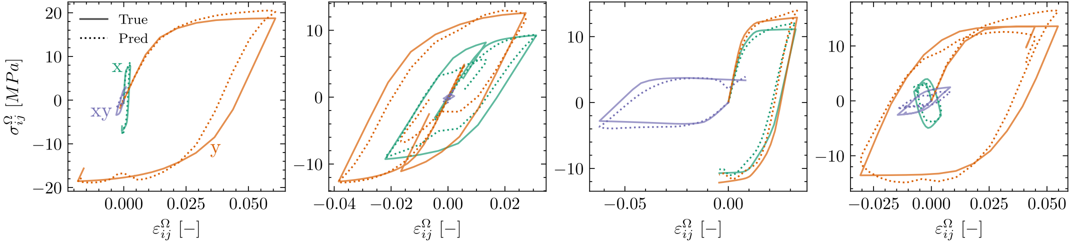


Extrapolating in timesteps is stable

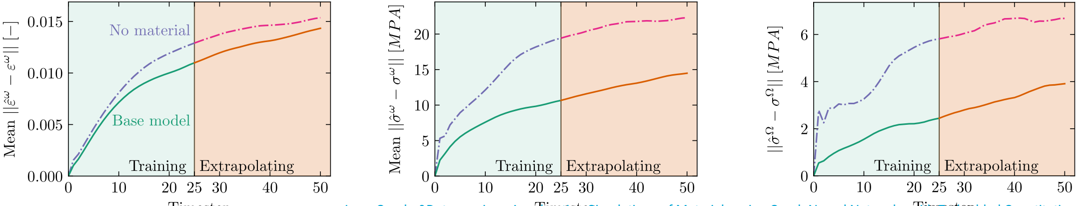


Trained for 25 timesteps → tested for 50

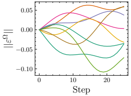
Autoregressive model: errors can accumulate but they do not



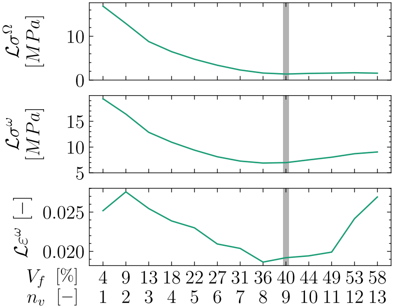
Average error increase per timestep:

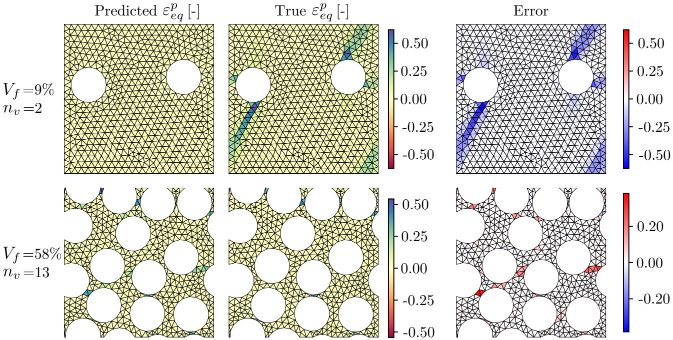


Unseen material settings



All training data uses a void volume fraction V_f of 40%, what happens when using other values?





→ curate training dataset according to needs

→ material parameters can be changed in the material model