Simulating material behavior across the scales with hybrid machine learning

Iuri Rocha

with work from

Marina Maia, Joep Storm, Nora Kovacs

TU Delft

SLIM MLAB LABS

Our motivation: high-performance and complex materials

We rely on high-performance materials for many applications, and that reliance will only increase

- Maximum efficiency with minimum material use, high tailorability
- Urgent challenges in climate adaptation, durable infrastructure, sustainable structures

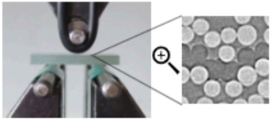
Our motivation: high-performance and complex materials

We rely on high-performance materials for many applications, and that reliance will only increase

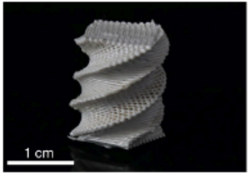
- Maximum efficiency with minimum material use, high tailorability
- Urgent challenges in climate adaptation, durable infrastructure, sustainable structures

At the same time, whole new families of materials are emerging

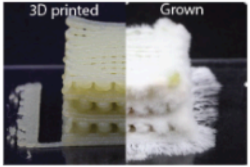
- 3D printing with molecular-scale control, engineered living materials



[Rocha *et al.*(2019), *EJMA*]



[Gantenbein *et al.*(2018), *Nature*]



[Gantenbein *et al.*(2023), *Nature Materials*]

Our motivation: high-performance and complex materials

We rely on high-performance materials for many applications, and that reliance will only increase

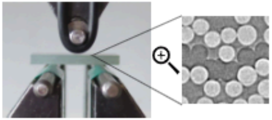
- Maximum efficiency with minimum material use, high tailorability
- Urgent challenges in climate adaptation, durable infrastructure, sustainable structures

At the same time, whole new families of materials are emerging

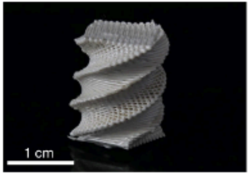
- 3D printing with molecular-scale control, engineered living materials

How can we better understand current materials and design new ones?

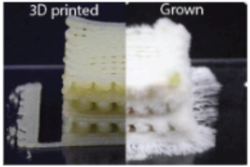
- A purely experimental approach is in any case a no go



[Rocha *et al.*(2019), *EJMA*]



[Gantenbein *et al.*(2018), *Nature*]



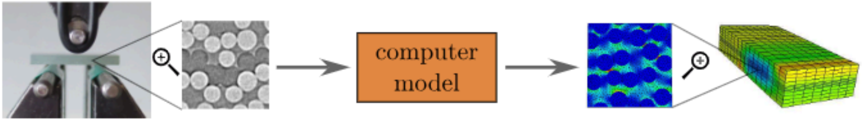
[Gantenbein *et al.*(2023), *Nature Materials*]

Materials and computer models – avenues for progress

The answer: combining a handful of experiments with extensive virtual testing

Multiscale modeling

- Model materials at very small and very large scales at the same time
- More knowledge $\Rightarrow$ smaller safety factors $\Rightarrow$ more efficient designs

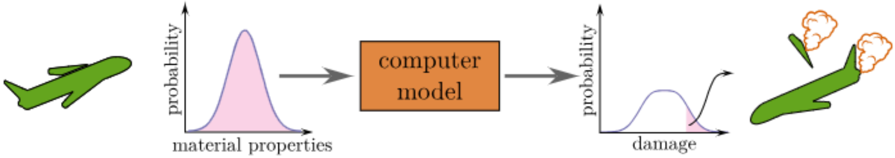


Materials and computer models – avenues for progress

The answer: combining a handful of experiments with extensive virtual testing

Uncertainty Quantification

- Material properties and some service loads are inherently uncertain
- Better understanding of their impact on performance $\Rightarrow$ more efficient designs

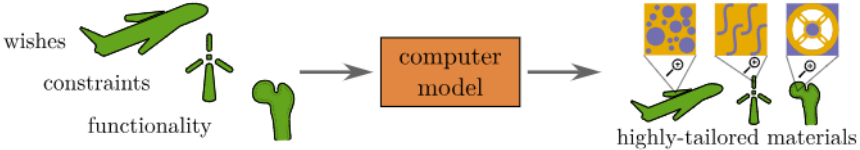


Materials and computer models – avenues for progress

The answer: combining a handful of experiments with extensive virtual testing

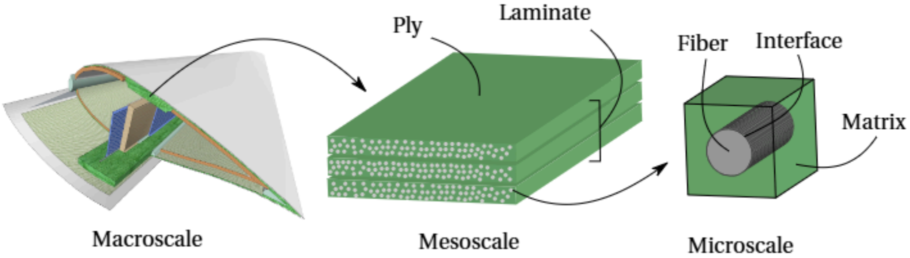
Design optimization

- Design variables can be tweaked at different scales (e.g. microstructures, micro properties)
- Finding the optimum among millions of possible combinations $\Rightarrow$ more efficient designs



Example — Aging of fiber-reinforced polymers

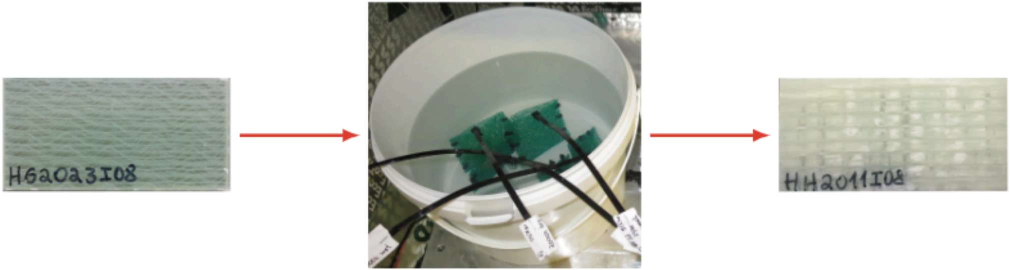
Make wind turbine blades last longer by better understanding the materials used:



Example — Aging of fiber-reinforced polymers

Make wind turbine blades last longer by better understanding the materials used:

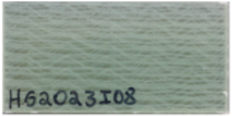
- Observation 1: material goes in a bucket with hot water, comes out looking ugly



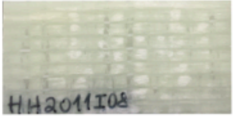
Example — Aging of fiber-reinforced polymers

Make wind turbine blades last longer by better understanding the materials used:

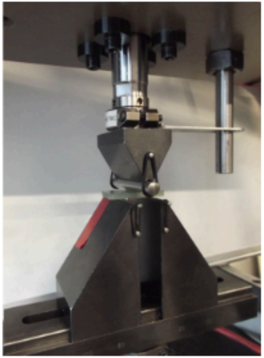
- Observation 1: material goes in a bucket with hot water, comes out looking ugly
- Observation 2: if we test the ugly material it is now much weaker



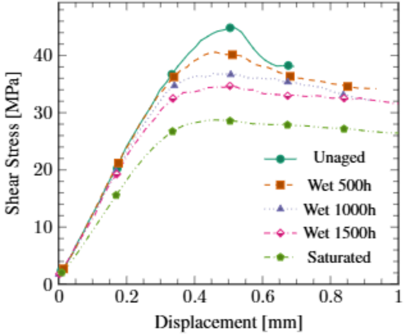
aging



testing



processing

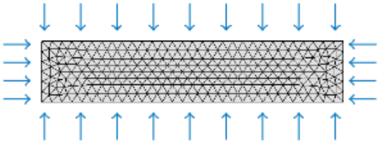


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

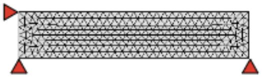
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathrm{b} = 0$$

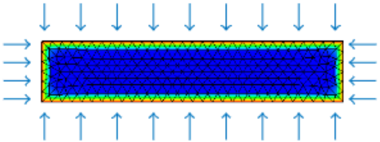


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

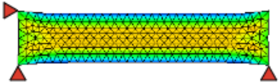
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$$

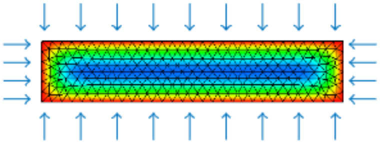


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

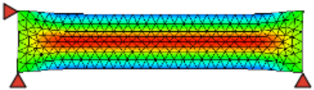
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathrm{b} = 0$$

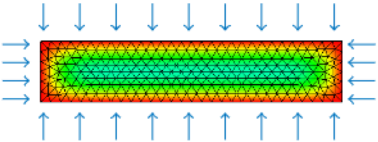


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

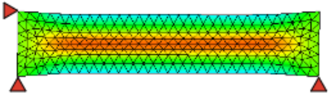
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$$

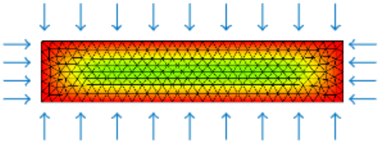


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

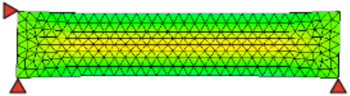
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathrm{b} = 0$$

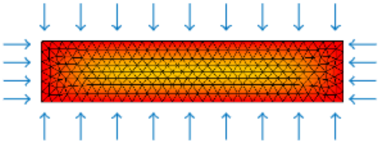


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

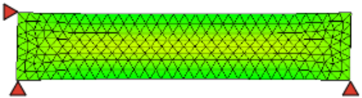
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$$

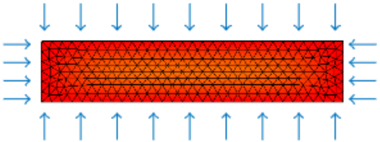


Example — Aging of fiber-reinforced polymers

How can we model this? Let us try a finite element model with water and mechanics:

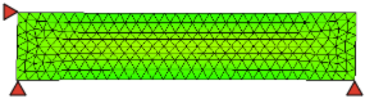
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$



Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathrm{b} = 0$$



Example — Aging of fiber-reinforced polymers

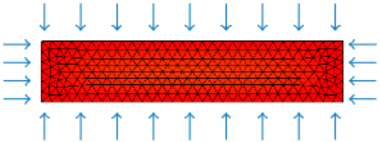
How can we model this? Let us try a finite element model with water and mechanics:

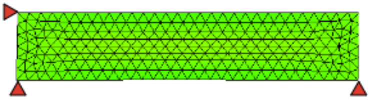
Water diffusion:

$$\frac{\partial c}{\partial t} - D\nabla \cdot (\nabla c) = 0$$

Stress equilibrium:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}$$

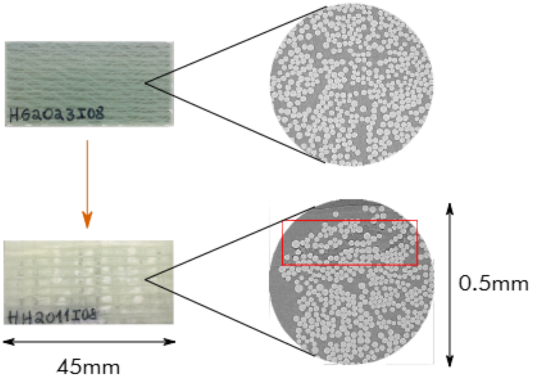




Okay, cool... but that still does not explain why the material degrades.

Example — Aging of fiber-reinforced polymers

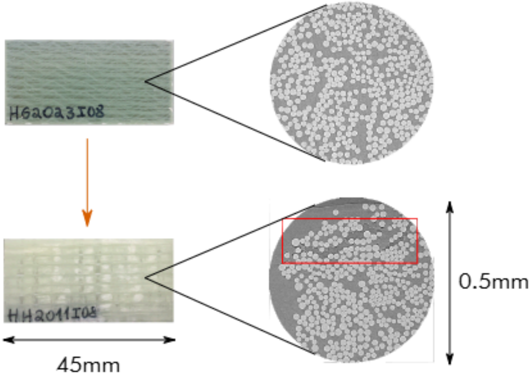
It turns out hygrothermal aging is a multiscale phenomenon:

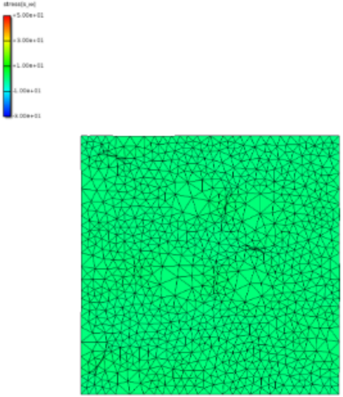


Example — Aging of fiber-reinforced polymers

It turns out hygrothermal aging is a multiscale phenomenon:

- With FEM, an arbitrarily complex microstructure can be modeled!
- With a few extra tricks, we make sure that both geometry and deformation are periodic

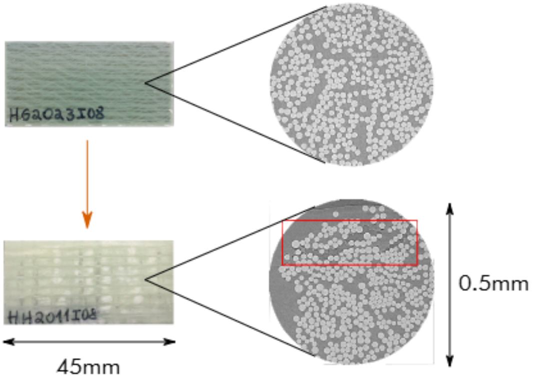


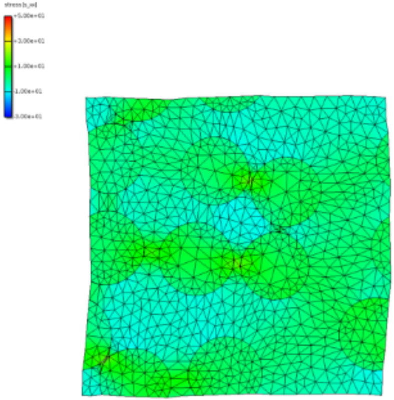


Example — Aging of fiber-reinforced polymers

It turns out hygrothermal aging is a multiscale phenomenon:

- With FEM, an arbitrarily complex microstructure can be modeled!
- With a few extra tricks, we make sure that both geometry and deformation are periodic

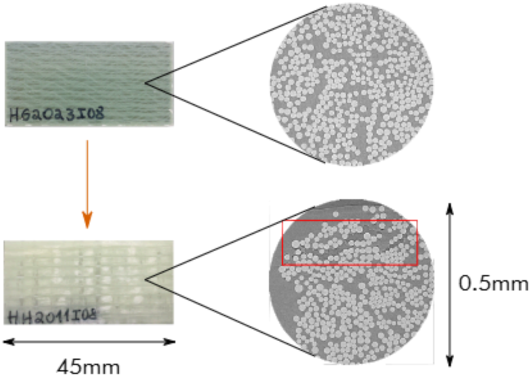


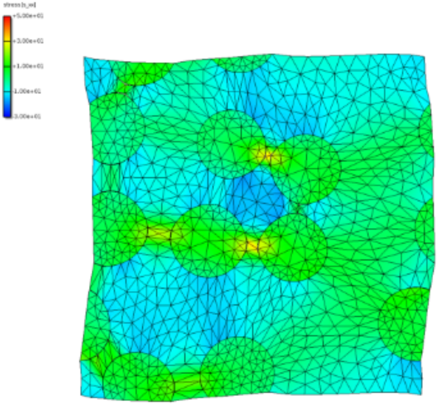


Example — Aging of fiber-reinforced polymers

It turns out hygrothermal aging is a multiscale phenomenon:

- With FEM, an arbitrarily complex microstructure can be modeled!
- With a few extra tricks, we make sure that both geometry and deformation are periodic

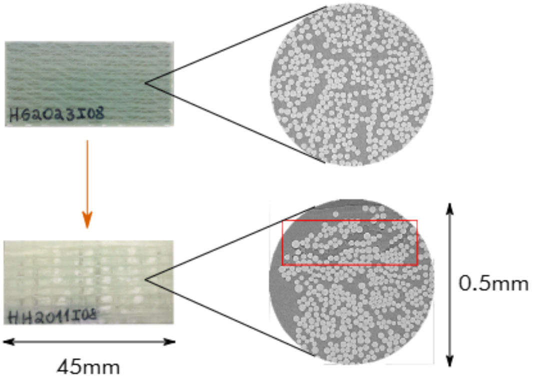


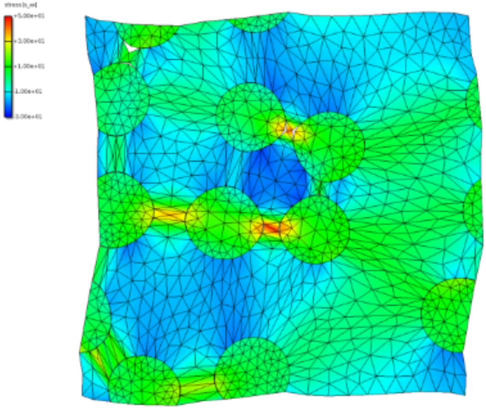


Example — Aging of fiber-reinforced polymers

It turns out hygrothermal aging is a multiscale phenomenon:

- With FEM, an arbitrarily complex microstructure can be modeled!
- With a few extra tricks, we make sure that both geometry and deformation are periodic

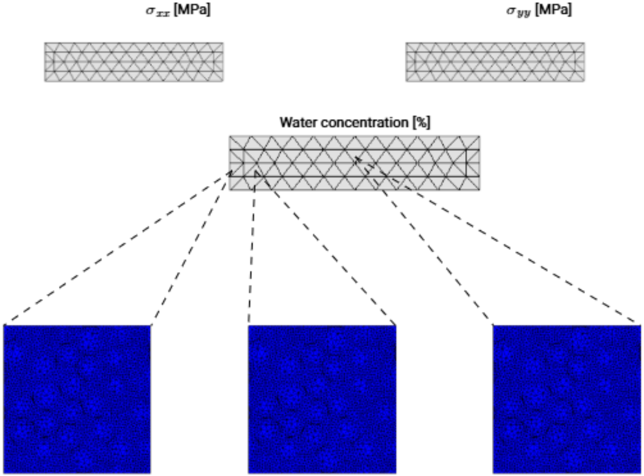




Example — Aging of fiber-reinforced polymers

Now for our final trick, we combine these two scales into a single model:

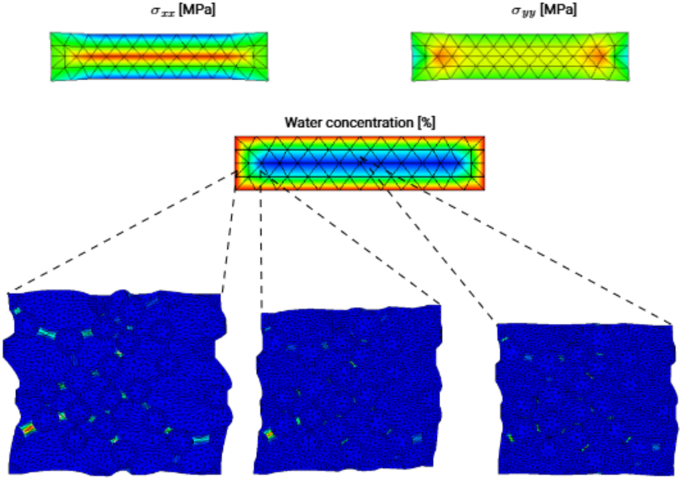
- FEM at both scales, so a lot of FEM models to run...



Example — Aging of fiber-reinforced polymers

Now for our final trick, we combine these two scales into a single model:

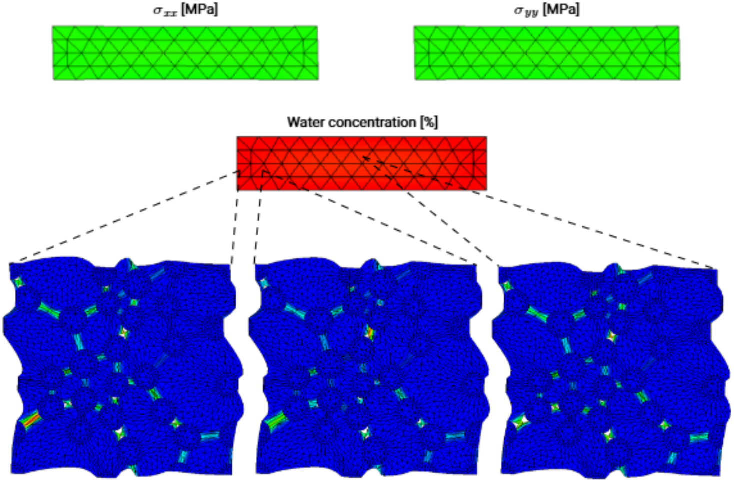
- FEM at both scales, so a lot of FEM models to run...



Example — Aging of fiber-reinforced polymers

Now for our final trick, we combine these two scales into a single model:

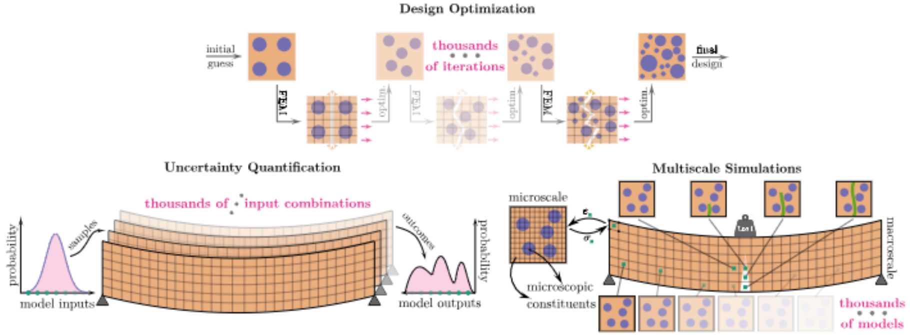
- FEM at both scales, so a lot of FEM models to run...



Good models do not come cheap

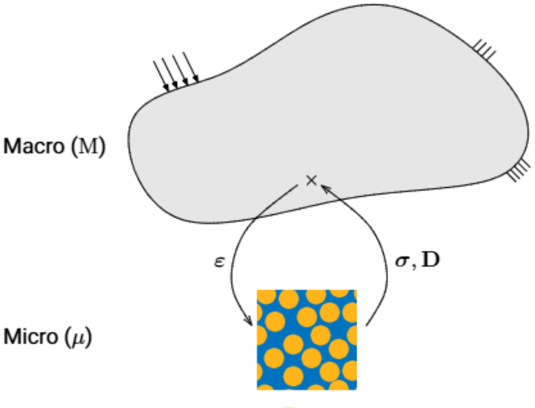
We need a way to run these models much faster:

- Highly detailed models can take months to run and we need to run a lot of them



For now let us just focus on multiscale modeling

Compute stress σ as the volume average in a deformed Representative Volume Element (RVE)



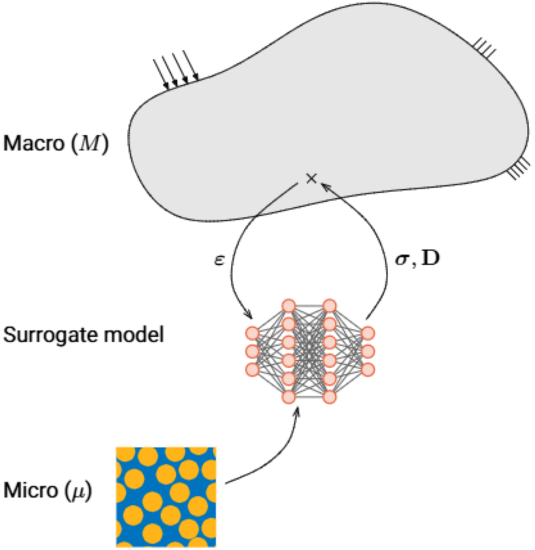
The FE^2 method:

- Conceptually simple: FEM inside FEM
- Captures complex material behavior

Downside:

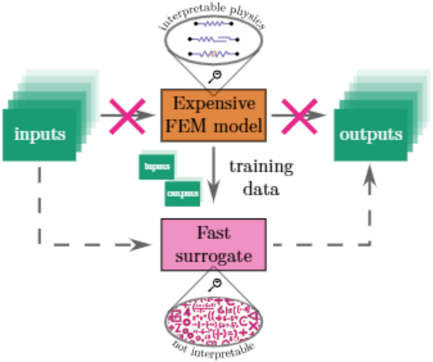
- Extreme computational cost

We need multiscale models to run faster



A popular solution:

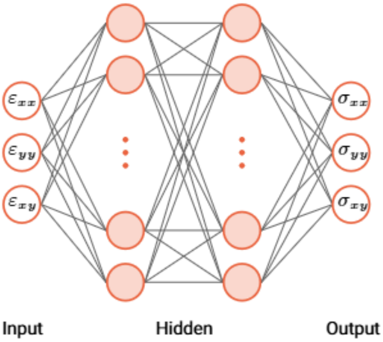
- Run the micro FEM a few times
- Use the results to train a ML model
- Use the trained model at the macroscale



Neural network surrogate models for multiscale FEM

Neural networks have become extremely popular for this task:

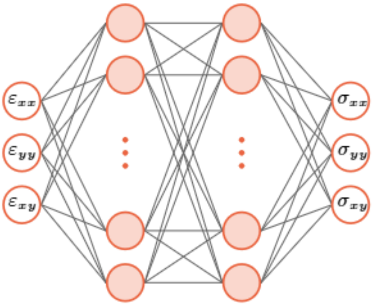
- They are universal approximators, i.e can describe any continuous mathematical function
- They are extremely fast, usually $10000\times \sim 100000\times$ faster than FEM

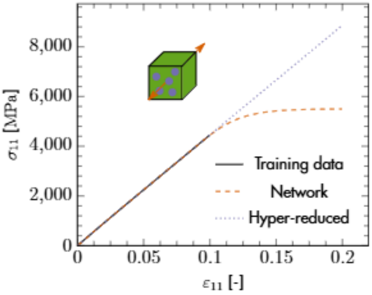


Neural network surrogate models for multiscale FEM

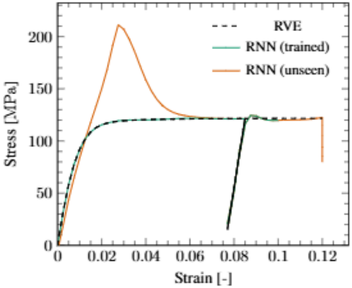
Neural networks have become extremely popular for this task:

- They are universal approximators, i.e can describe any continuous mathematical function
- They are extremely fast, usually $10000\times \sim 100000\times$ faster than FEM
- Yet...





[Rocha *et al.*(2020), *EJMSOL*]

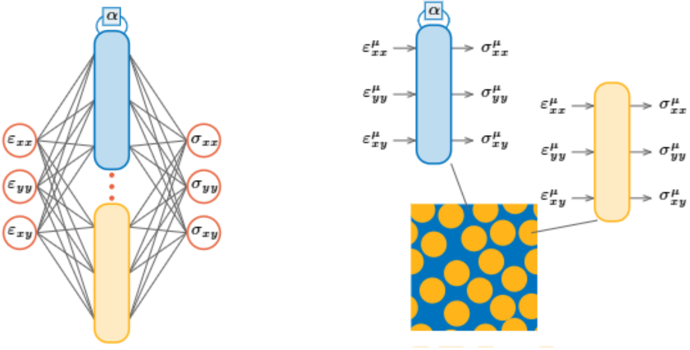


[Rocha *et al.*(2023), *Mech Mater*]

Our approach: Physically Recurrent Neural Networks

Our solution is to go for a hybrid machine learning model:

- Substitute some of the artificial neurons for cells with real physical models
- Leave the rest of the model intact and train it in the same way as normal ML models

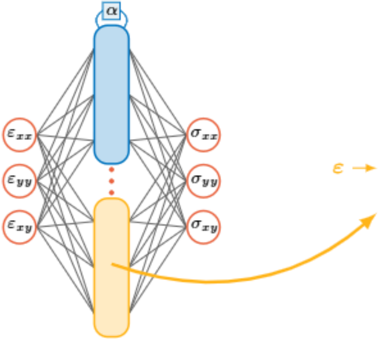


[Maia *et al.*(2023), *CMAME*]

Our approach: Physically Recurrent Neural Networks

Our solution is to go for a hybrid machine learning model:

- Substitute some of the artificial neurons for cells with real physical models
- Leave the rest of the model intact and train it in the same way as normal ML models
- How are the models added? By using material model code we had been using for decades



Linear elasticity

$\varepsilon \rightarrow$

```
def elastic_material(self,strain,ipoint):
    stiff = self.stiff_at_point(ipoint)
    stress = stiff @ strain

    return stiff, stress
```

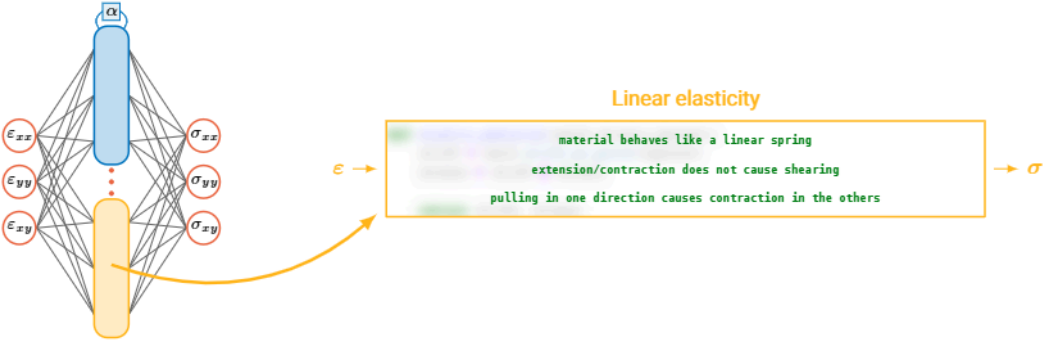
$\rightarrow \sigma$

[Maia *et al.*(2023), *CMAME*]

Our approach: Physically Recurrent Neural Networks

Our solution is to go for a hybrid machine learning model:

- Substitute some of the artificial neurons for cells with real physical models
- Leave the rest of the model intact and train it in the same way as normal ML models
- How are the models added? By using material model code we had been using for decades

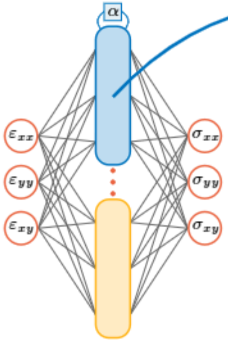


[Maia *et al.*(2023), *CMAME*]

Our approach: Physically Recurrent Neural Networks

Our solution is to go for a hybrid machine learning model:

- Substitute some of the artificial neurons for cells with real physical models
- Leave the rest of the model intact and train it in the same way as normal ML models
- How are the models added? By using material model code we had been using for decades



Von Mises (J_2) plasticity

$\varepsilon, \alpha_{\text{old}} \rightarrow$

```
def j2_material(self,strain,ipoint):
    eps_elastic = self._y[ipoint].elastic_strain(strain)

    trial_stress = elastic_material(eps_elastic,ipoint)

    trial_f = self._y[ipoint].yield_trial(trial_stress)

    if trial_f < -self._tol:
        return trial_stress
    else:
        sol = root_scalar(self._y[ipoint].yield_function, method='newton')

        stress = self._y[ipoint].updated_stress()
        stiff  = self._y[ipoint].tangent_stiffness()

        return stress
```

$\rightarrow \sigma, \alpha_{\text{new}}$

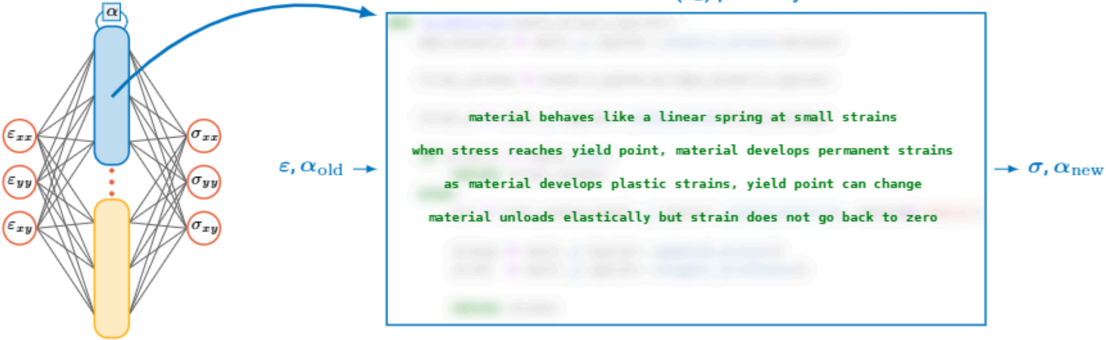
[Maia *et al.*(2023), *CMAME*]

Our approach: Physically Recurrent Neural Networks

Our solution is to go for a hybrid machine learning model:

- Substitute some of the artificial neurons for cells with real physical models
- Leave the rest of the model intact and train it in the same way as normal ML models
- How are the models added? By using material model code we had been using for decades

Von Mises (J_2) plasticity



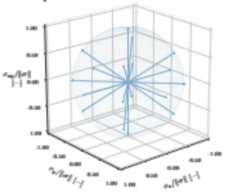
[Maia *et al.*(2023), *CMAME*]

PRNNs versus conventional ML models

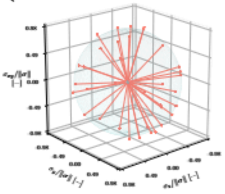
Training and testing with the same type of data:

PRNN (18 monotonic curves) RNN (576 curves with unloading)

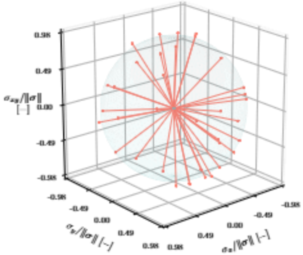
Train

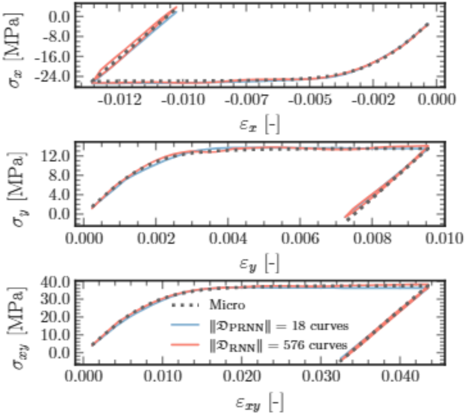


or



Test





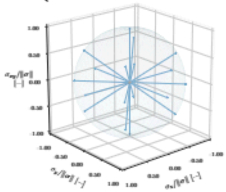
[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

PRNNs versus conventional ML models

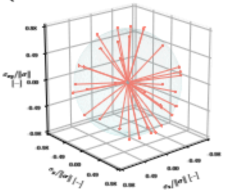
Testing with more challenging paths:

PRNN (18 monotonic curves) RNN (864 curves with unloading)

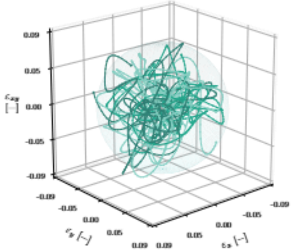
Train

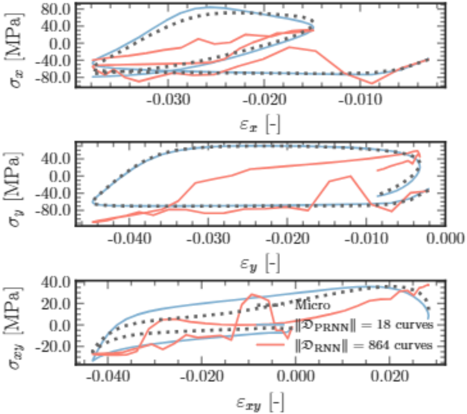


or



Test





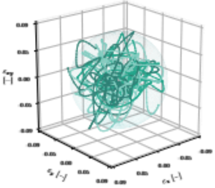
[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

PRNNs versus conventional ML models

Okay, how about training with the challenging paths?

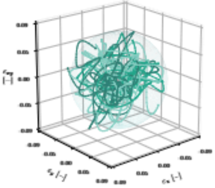
Train

PRNN (36 curves)

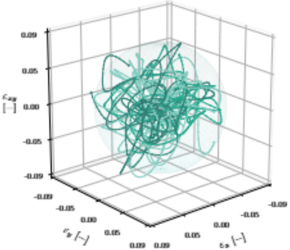


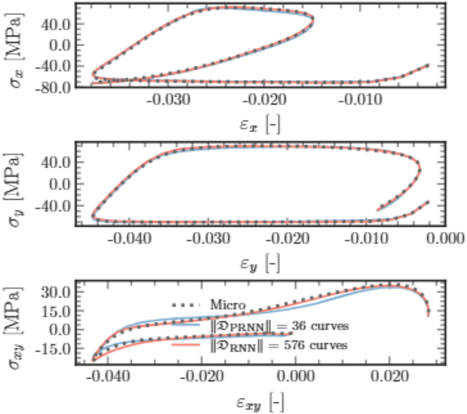
or

RNN (576 curves)



Test





[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

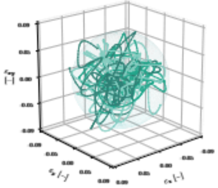
PRNNs versus conventional ML models

Great, but now we can do the simple ones again... right? No!

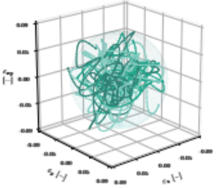
PRNN (36 curves)

RNN (576 curves)

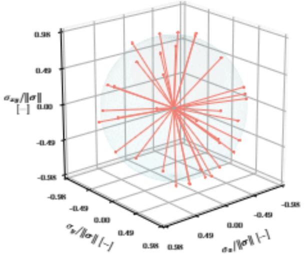
Train

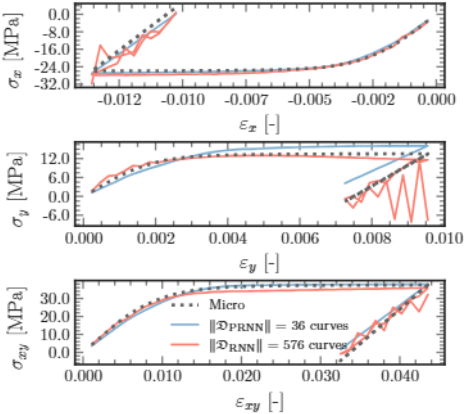


or



Test



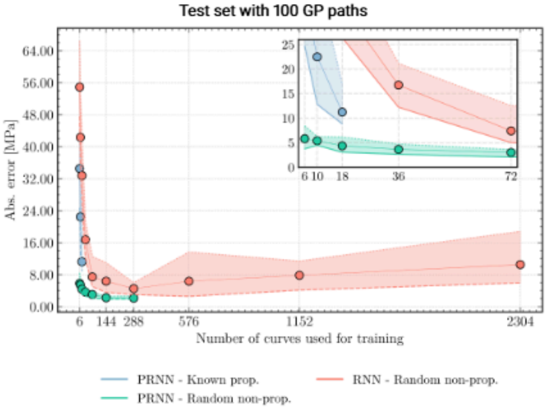


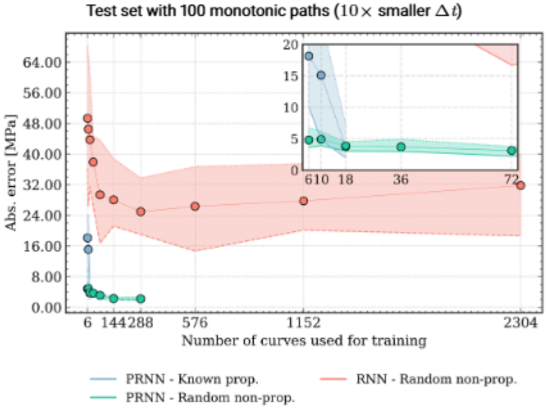
[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

PRNNs versus conventional ML models

Learning curves with 10 realizations per dataset size:

- PRNNs remain accurate even when trained with very small datasets



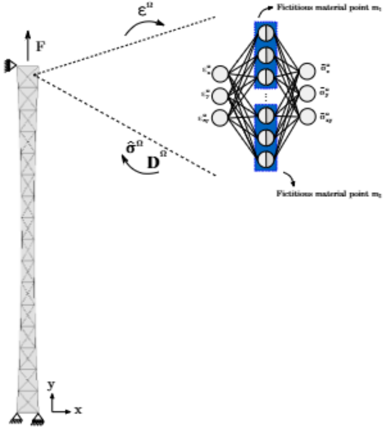


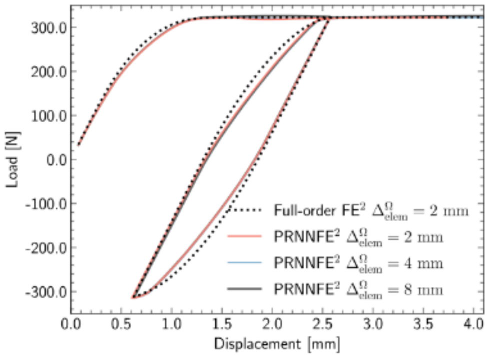
[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

PRNNs for FE^2

Trying out the model for FE^2, tapered bar with different mesh densities:

- Speed-ups against FE^2 of $20000\times \sim 30000\times$
- Trained without unloading, still predicts it accurately





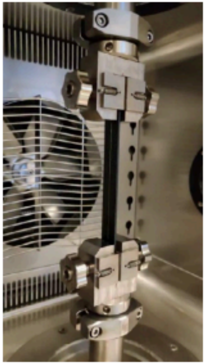
[Maia, Rocha, Kerfriden, van der Meer (2023), **CMAME**]

What we are now able to do: Reproducing real experiments

We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

Uniaxial tension experiment



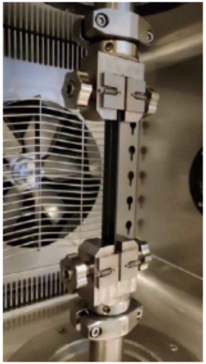
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

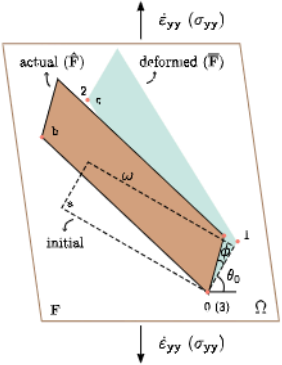
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

Uniaxial tension experiment



Specialized arc-length



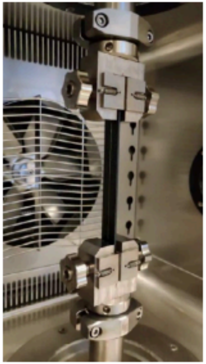
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

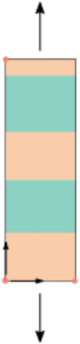
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

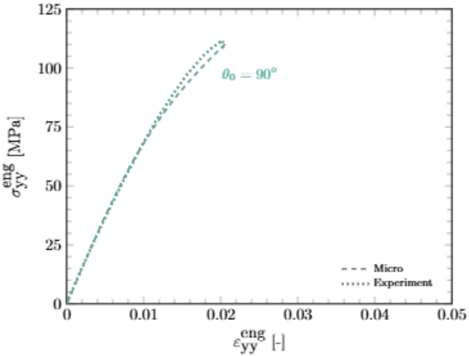
Uniaxial tension experiment



Off-axis angle



Engineering stress-strain



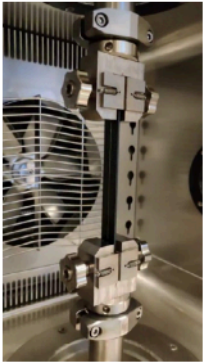
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

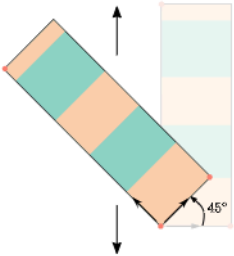
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

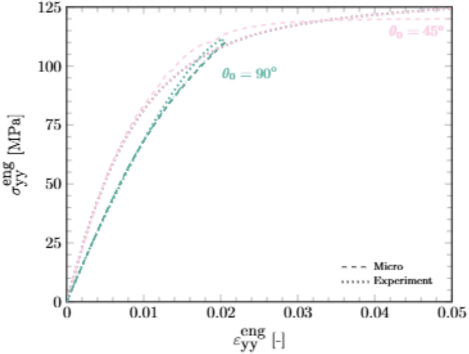
Uniaxial tension experiment



Off-axis angle



Engineering stress-strain



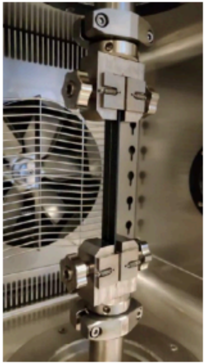
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

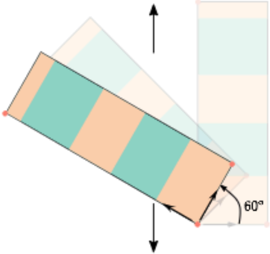
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

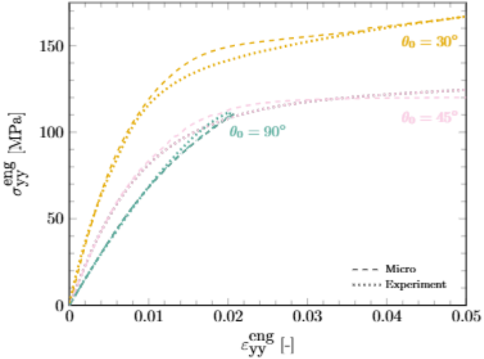
Uniaxial tension experiment



Off-axis angle



Engineering stress-strain



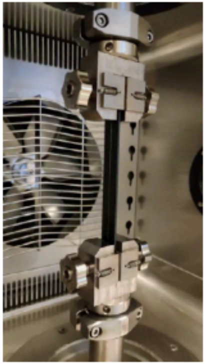
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

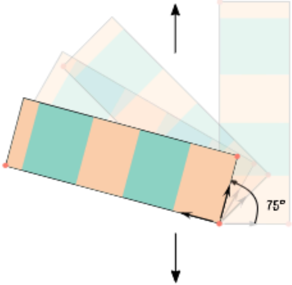
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

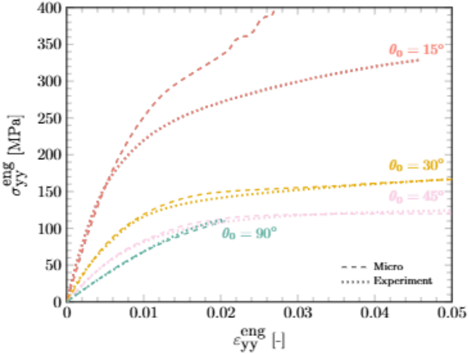
Uniaxial tension experiment



Off-axis angle



Engineering stress-strain



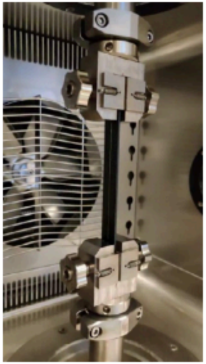
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

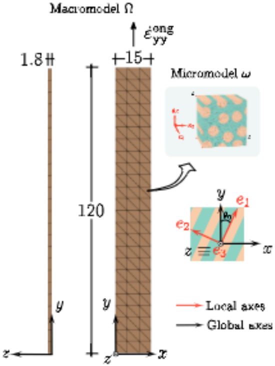
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

Uniaxial tension experiment



Multiscale model



Engineering stress-strain

too expensive
to simulate

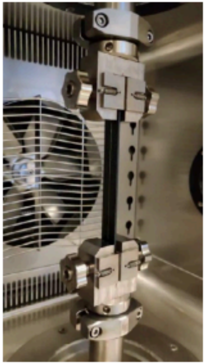
[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

What we are now able to do: Reproducing real experiments

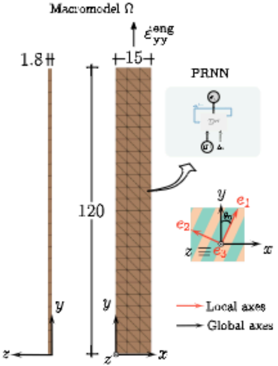
We also used it to confirm some experimental results that had been puzzling us:

- Uniaxial tension on off-axis carbon-PEEK coupons

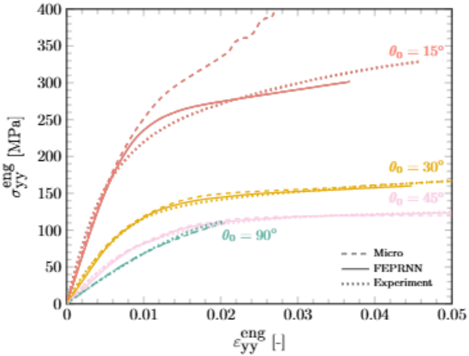
Uniaxial tension experiment



Multiscale model



Engineering stress-strain

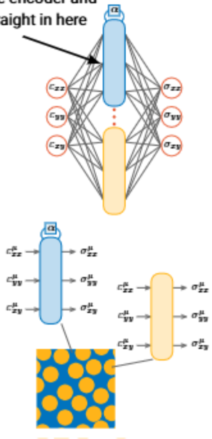


[Maia, Rocha, van der Meer (2025), **arXiv:2501:10193**]

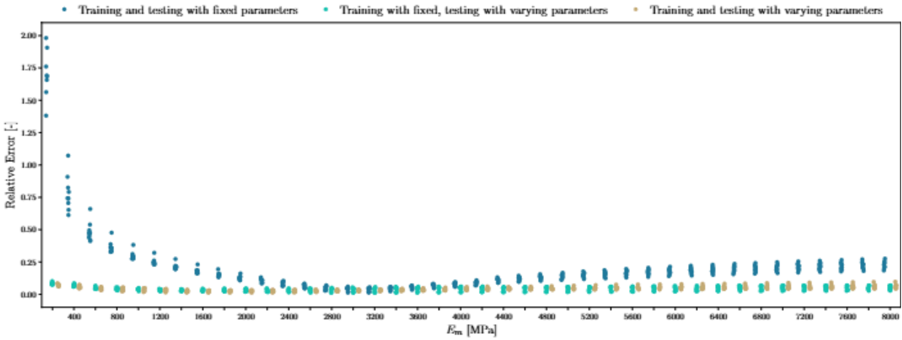
Microscale Uncertainty Quantification

The network inherits material properties. Can we also extrapolate for those?

- Apparently yes! Train with a single value, extrapolate quite far



Matrix Young's modulus E_m



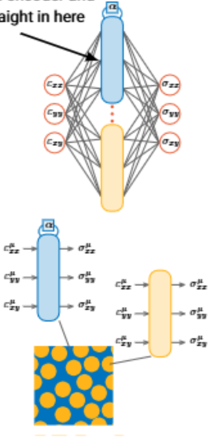
[Kovacs, Rocha, Furtado, Camanho, van der Meer (2025), **arXiv:2504.11625**]

Microscale Uncertainty Quantification

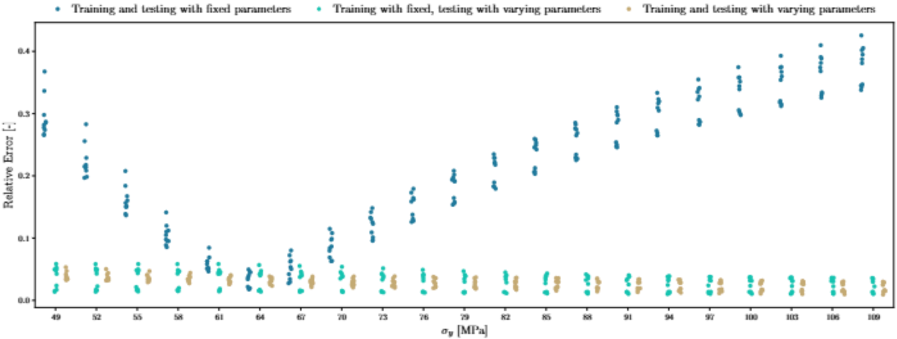
The network inherits material properties. Can we also extrapolate for those?

- Apparently yes! Train with a single value, extrapolate quite far

some features
skip the encoder and
go straight in here



Matrix yield stress σ_y

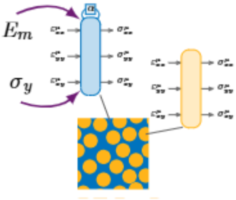


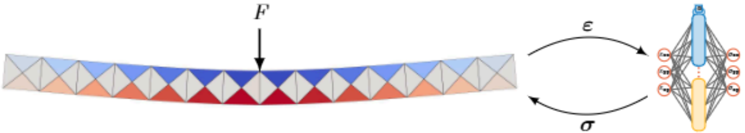
[Kovacs, Rocha, Furtado, Camanho, van der Meer (2025), **arXiv:2504.11625**]

Multiscale Uncertainty Quantification

Three-point bending, coarse macro mesh:

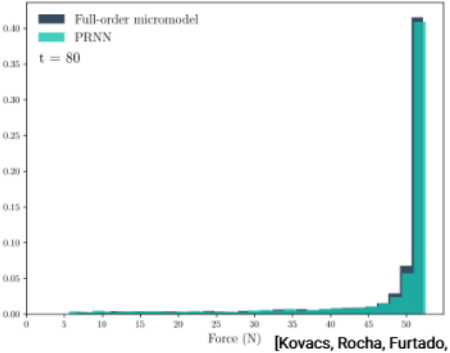
- Monte Carlo forward UQ, $N = 5000$ samples
- PRNN trained with a single (E_m, σ_y) pair
- Good agreement with full-order FE^2

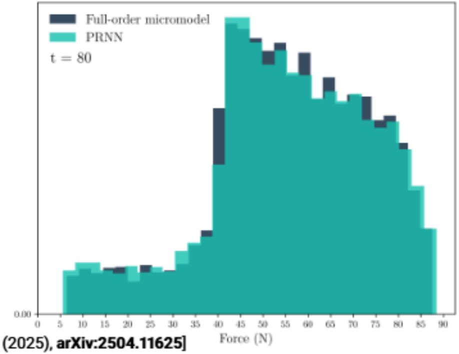




$$p(F) = \int p(F|E_m)p(E_m)\,\mathrm{d}E_m$$

$$p(F) = \int p(F|E_m, \sigma_y)p(E_m, \sigma_y)\,\mathrm{d}E_m\mathrm{d}\sigma_y$$



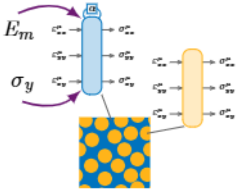


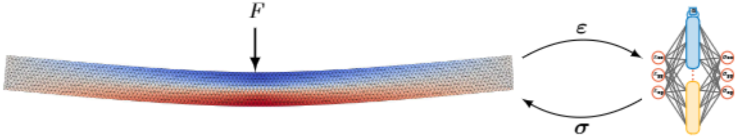
[Kovacs, Rocha, Furtado, Camanho, van der Meer (2025), **arXiv:2504.11625**]

What we are now able to do: Multiscale UQ

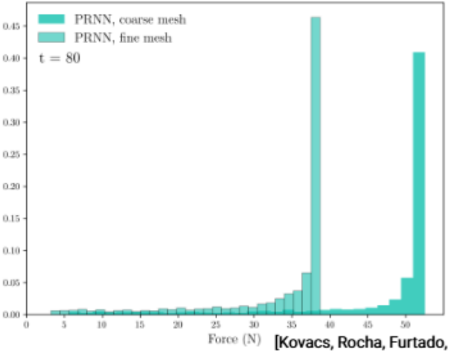
Three-point bending, fine macro mesh:

- Monte Carlo forward UQ, $N = 5000$ samples
- PRNN trained with a single (E_m, σ_y) pair
- Sequential FE^2 UQ would now take ≈50 years to run

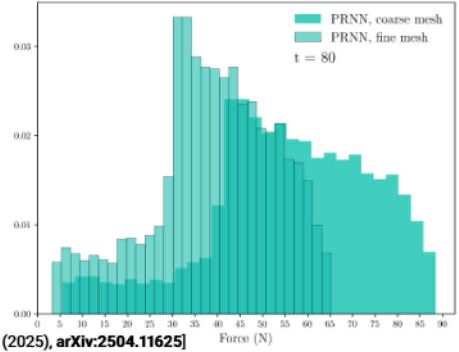




$$p(F) = \int p(F|E_m)p(E_m)\,\mathrm{d}E_m$$



$$p(F) = \int p(F|E_m,\sigma_y)p(E_m,\sigma_y)\,\mathrm{d}E_m\mathrm{d}\sigma_y$$

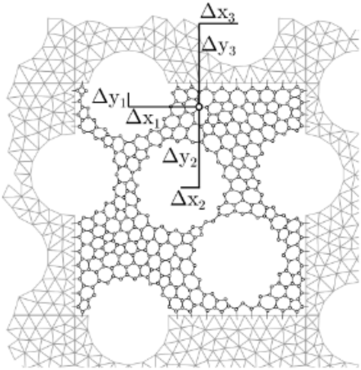


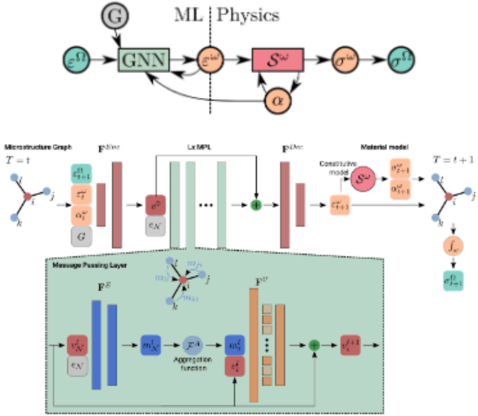
[Kovacs, Rocha, Furtado, Camanho, van der Meer (2025), **arXiv:2504.11625**]

Can we get complete microscopic fields?

All previous models discard the microscale model completely, which means information is lost

- We also tried a GNN approach with embedded material models
- Skips FEM but still predicts the complete microscopic response



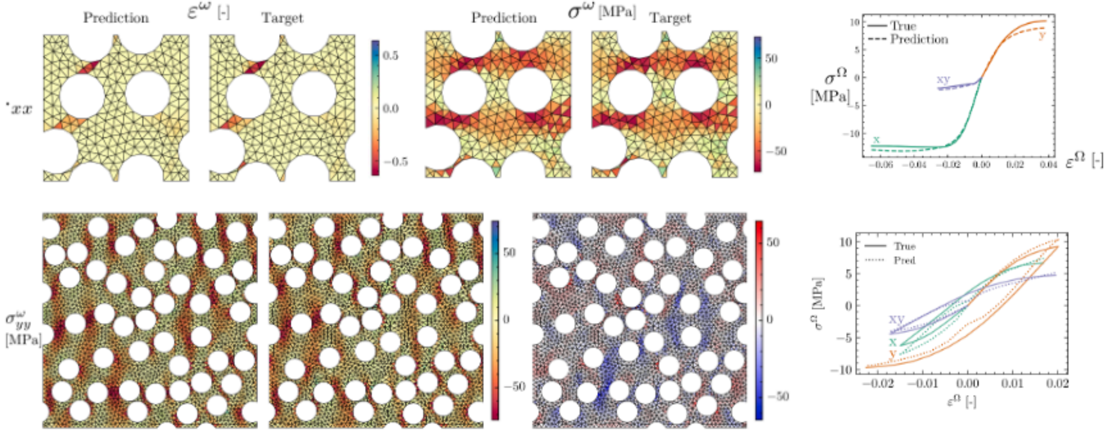


[Storm, Rocha, van der Meer (2024), **CMAME**]

A GNN for multiscale solid mechanics

We basically have an encode-process-decode-material approach:

- GNN computes strains (ε), material models compute stresses (σ)
- We train on models with up to 9 circular features, transfer to up to 64 features without retraining

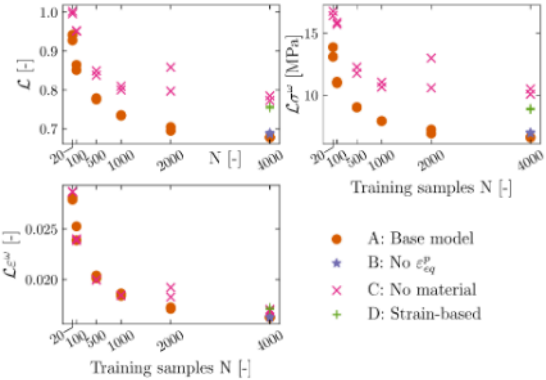


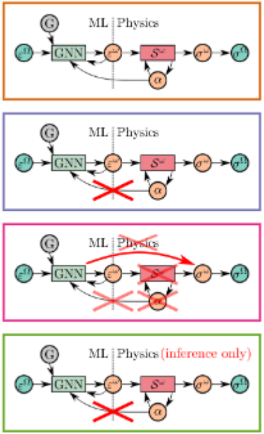
[Storm, Rocha, van der Meer (2024), **CMAME**]

A GNN for multiscale solid mechanics

The role of the material model

- Learning curves with/without material model and with material model just at inference time
- Once again baked-in physics benefits learning





[Storm, Rocha, van der Meer (2024), **CMAME**]

Thank you!