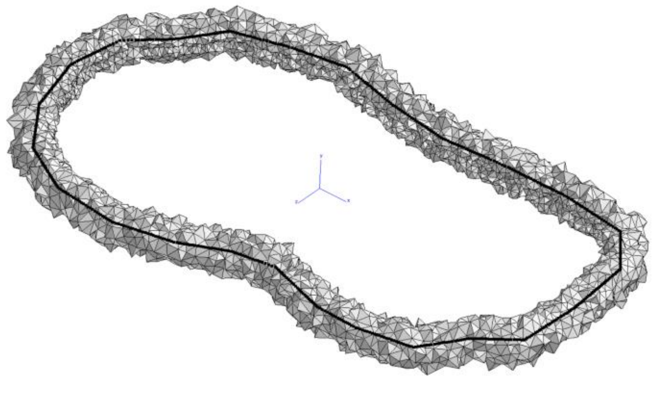


Figure 29: Smaller TLV for defect 1054

Centre de Mise en Forme des Matériaux

Mines Paris - PSL

CNRS UMR 7635

Antipolis, France

# Self-supervised ML of boundary conditions in mechanics of materials

**D. Ryckelynck**

**Th. Daniel, H. Launay, P. Pereira, H. Boukraichi, A. Aublet, S. Le Berre, D. Meshba, M. Bastico, A. Ferhat, P. Belamri**


LJAD, November 20th 2025, LMA, December 2nd 2025, EHF2026

# SCIENTIFIC MACHINE LEARNING

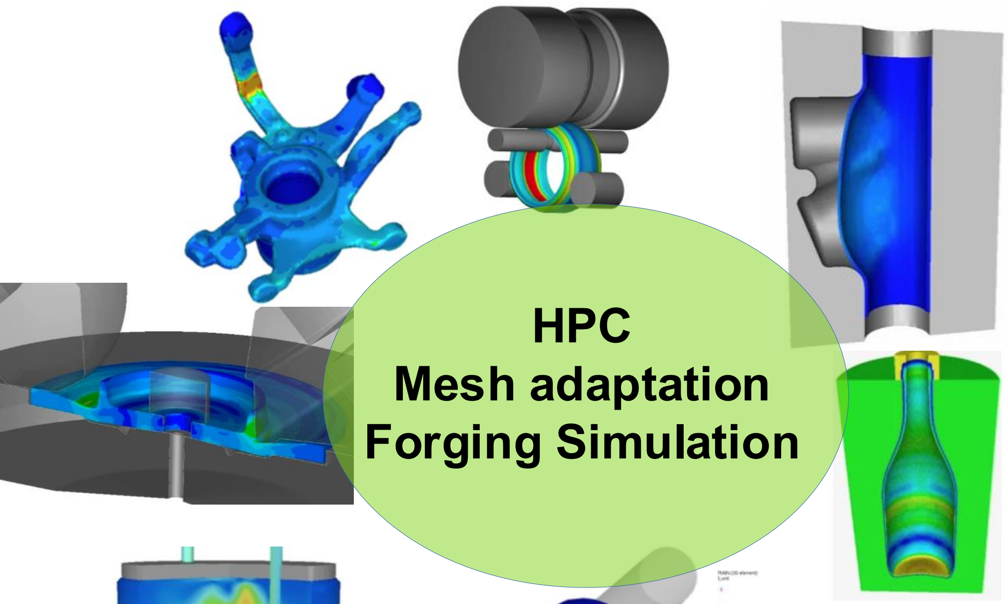
The use of AI, especially generative AI, has become widespread in the daily lives of younger generations.

This comes with the following risks for science and technology:

**R1**: Loss of human expertise: AI can reproduce results or analyses based on pre-existing data, but it does not **understand** the scientific knowledge. AI is performing imitations. An erosion of engineers and researchers' skills who use AI could reduce their ability to innovate or solve complex problems. This risk can be countered by scientific machine learning that do not ignore this risk.
**R2**: Undetected biases and errors: AI models rely on databases that may contain biases, modeling errors, or gaps. Without human expertise to validate and interpret the results, critical errors could go unnoticed, with serious industrial or safety consequences. It is important to continue training experts with strong AI and applied mathematics skills in a **Human-Centered AI framework**.
**R3**: Excessive **standardization**: AI tends to favor solutions optimized for average cases. Care must be taken to build training databases where data variability has been favored.
**R4**: Technological dependence: Blind trust in AI tools could make industry players too dependent on these systems, reducing their autonomy.
**R5**: **Hindrance to knowledge transfer**: science is based on a scientific culture transmitted through teaching and practice. Engineers and researchers must continue to develop their **theoretical and practical skills**, integrating AI as a lever for analysis, not as a turnkey solution.

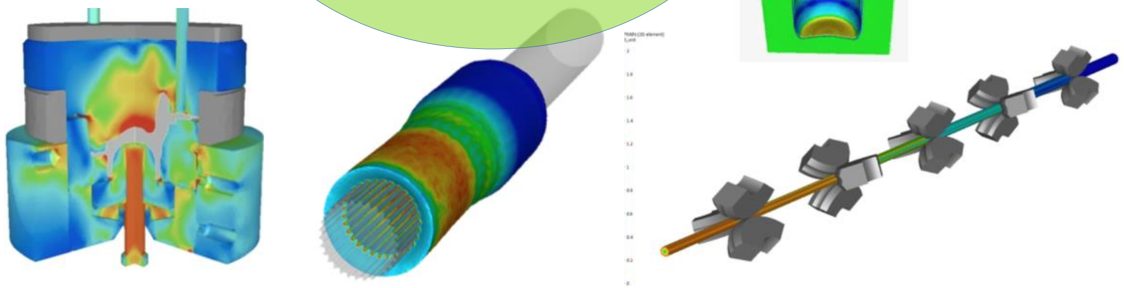
# SIMULATION IN MECHANICS OF MATERIALS AND MANUFACTURING PROCESSES

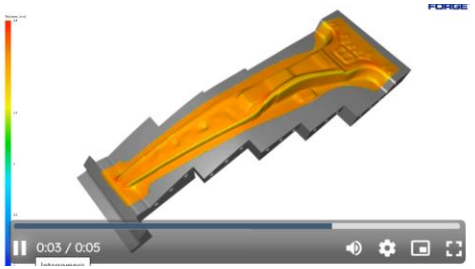
Simulation since the 60s :

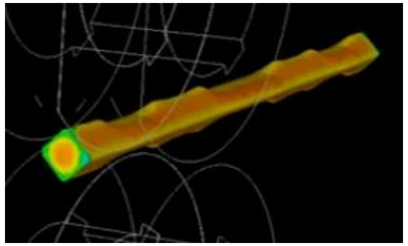


**HPC**
**Mesh adaptation**
**Forging Simulation**

- Reduce testing and empirical phases on real systems
- Get it right the first time
- Provide predictions (with or without defects/damage)
- Modeling and understanding
- Compare experiments and predictions, address inverse problems
- Perform sensitivity analysys
- Optimize designs or processes...





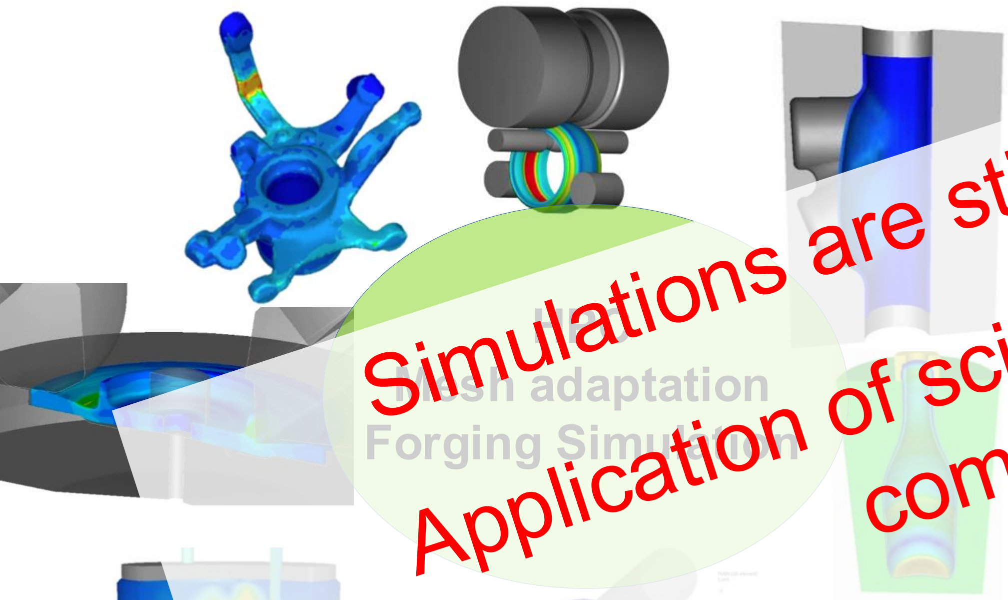

Since the 2000s model order reduction (in Zset, Aster, Cast3M, Ansys, without mesh adapticity).

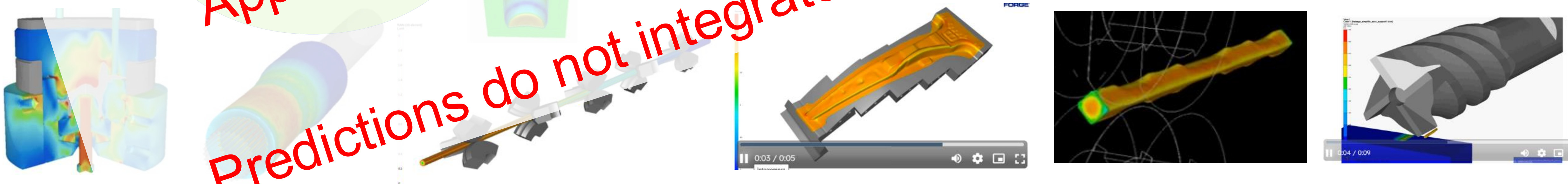
Images Transvalor https://www.transvalor.com/

# SIMULATION IN MECHANICS OF MATERIALS AND MANUFACTURING PROCESSES

Simulation since the 60s :

- Reduce testing and empirical phases on real systems
- Provide predictions (with or without defects/damage)
- Modeling and understanding
- Compare tests and predictions, address inverse problems
- Perform sensitivity analysys
- Get it right the first time
- Optimize designs or processes...




Simulations are still too slow! HPC is not enough!

Application of scientific knowledge is limited by its computational complexity!

Predictions do not integrate new tasks enabled by AI.

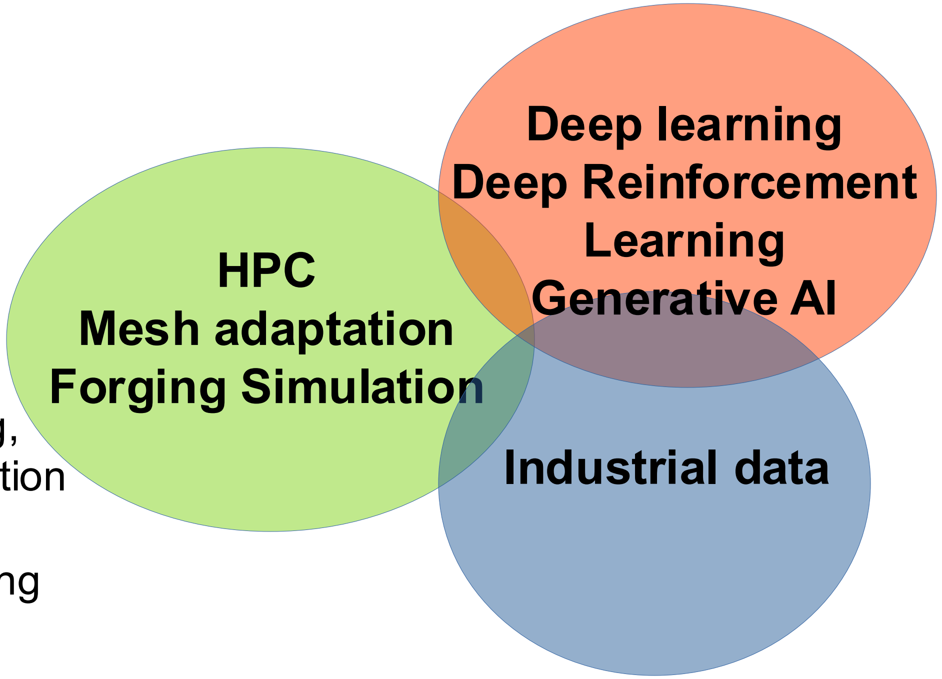
Since the 2000s model order reduction (in Zset, Aster, Cast3M, Ansys, without mesh adapticity).

Images Transvalor https://www.transvalor.com/

# INTEGRATING PHYSICS-BASED MODELING WITH AI

## Fundamentally transforming the use of simulation through data and machine learning

Hybrid AI
Scientific Deep Learning




- Develop **new engineering tasks**
- Develop new **global and local indicators** such as anomaly index.
- ...
- Propose **contributions to XAI**

- **Assist** human for : digital twining, modeling constitutive laws or friction laws, train new collaborators …
- **Automate and speedup** modeling task
- **Optimization**

# IA

## Machine Learning and Deep Learning

**Deep learning**
**Deep Reinforcement Learning**
**Generative AI**

Main categories:

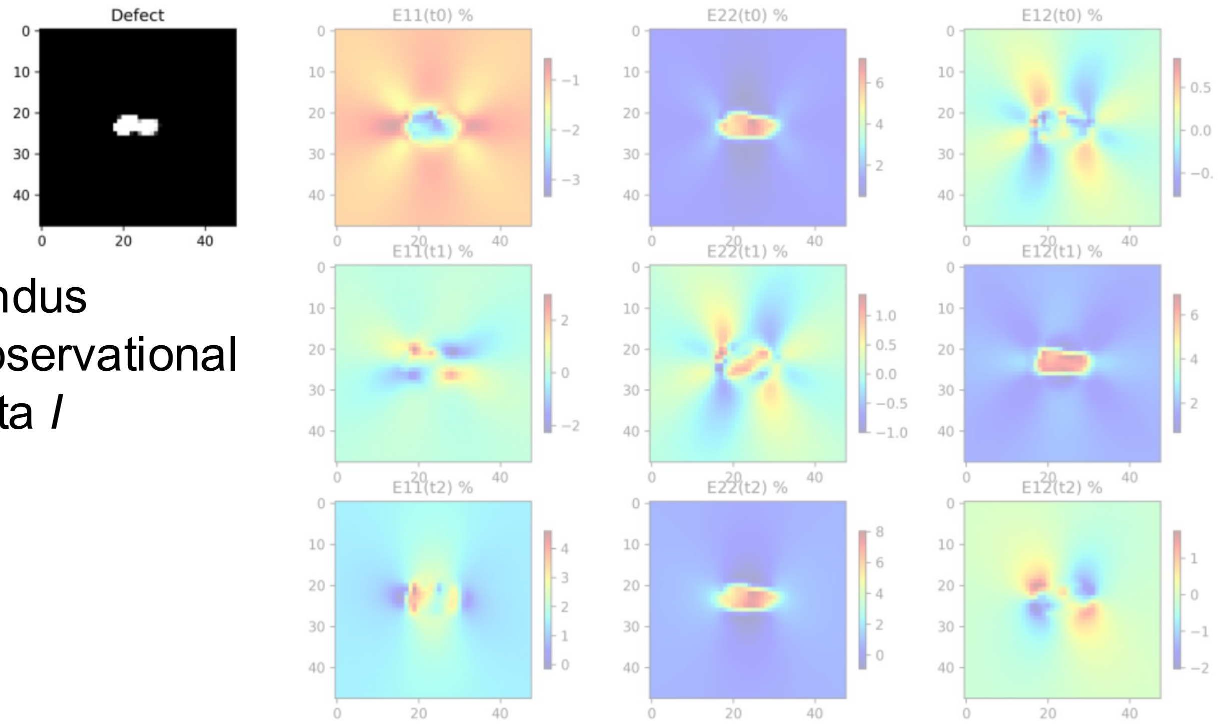
- Supervised machine learning**, Y(X) based on examples** (X,Y)
- Unsupervised machine learning based on X (clustering, OOD...).
- Self-supervised machine learning**, Y(X) = X**, pretrained models, dimensionality reduction...
- Reinforcement learning, optimization, autonomous systems...

Implementation steps:

- Choose an ambient space for data
- Save **datasets**
- **Augment data** in the train dataset
- Choose a neural network **architecture** and a **loss**
- **Train** the neural network
- **Test** the neural network

# Numerical example: void modeling in dilution condition and elasticity

Xindus = Image of a defect (48x48x1 tensor),
Y = 3 strain fields of 3 components related to 3 macroscopic loading




Xindus
Observational data *I*

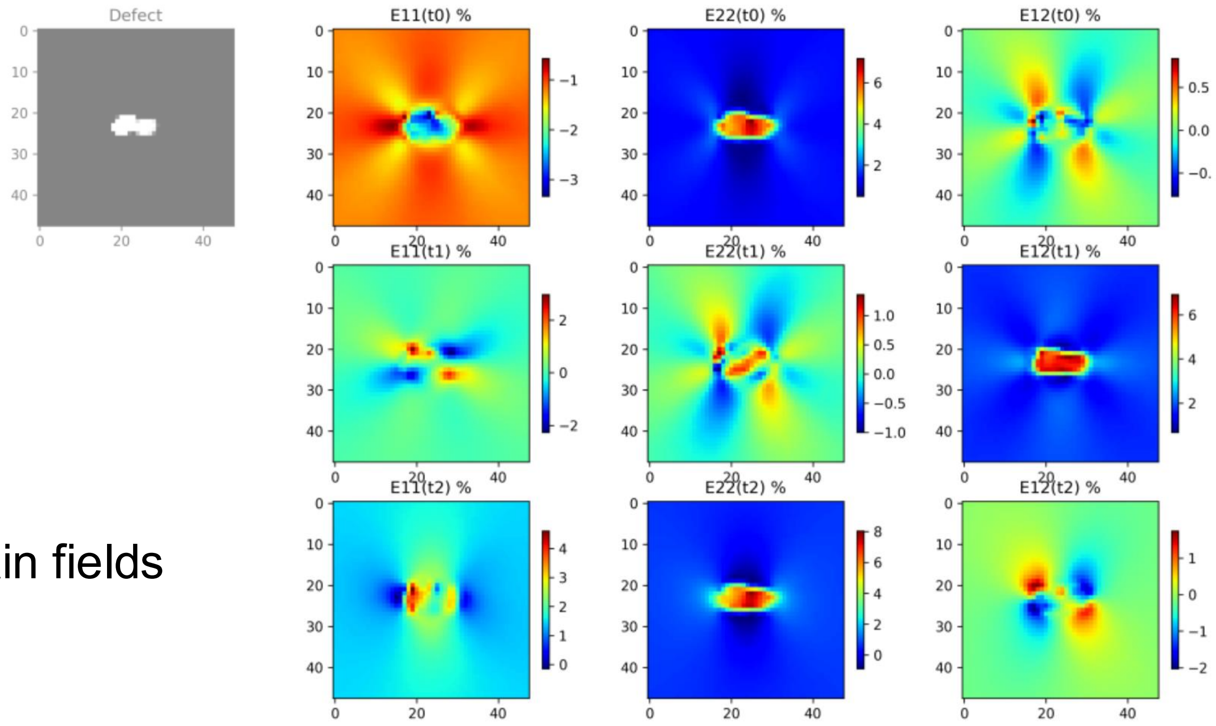

**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_\square$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)})$. Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.

# Numerical example: void modeling in dilution condition and elasticity

Xindus = Image of a defect (48x48x1 tensor),
Y = 3 strain fields of 3 components related to 3 macroscopic loading

Train a digital twin for strain predictions

Y: 9 strain fields



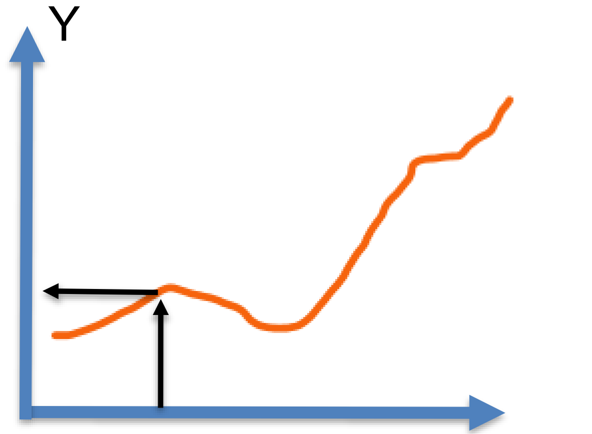
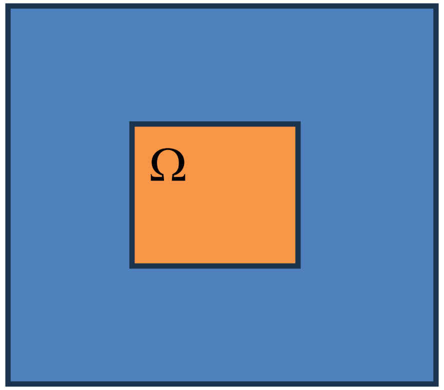




**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_\square$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)}$.
Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.

# Theory: void modeling in dilution condition and elasticity




Domain for dilution condition

Submodel

$$div(\,\boldsymbol{C}(\boldsymbol{x})\,\varepsilon(\boldsymbol{u})\,) = f \quad \forall x \in \Omega, \qquad \boldsymbol{u}(x) = \boldsymbol{u}_D(x) \quad \forall x \in \partial\Omega$$

Ellipticity

$$\alpha_{\Omega,C}\,\|v\|_{\Omega}^{2} \leq \int_{\Omega} tr\big(\varepsilon(\boldsymbol{v})\,\boldsymbol{C}\,\varepsilon(\boldsymbol{v})\big)\,d\Omega \leq \gamma_{\Omega,C}\,\|v\|_{\Omega}^{2}$$

Coercivity
$\alpha_{\Omega,C} > 0$

Continuity
$\gamma_{\Omega,C} > 0$

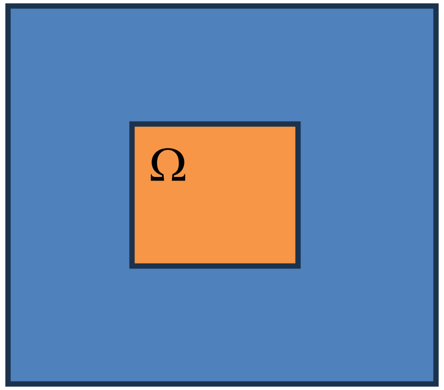
Target application for computer vision:

$$I \rightarrow \Omega(\mathrm{I}), \mathbf{C}(\mathrm{x};\mathrm{I}), \boldsymbol{u}_D(\mathrm{x};\mathrm{I}) \qquad x \in \Omega(I)$$

Learning boundary condition $\boldsymbol{u}_D(\mathrm{x};\mathrm{I})$.

# Theory: void modeling in dilution condition and elasticity

$$div(\, \boldsymbol{C}(\boldsymbol{x})\, \varepsilon(\boldsymbol{u})\,) = f \quad \forall x \in \Omega, \qquad \boldsymbol{u}(x) = \boldsymbol{u}_D(x) \quad \forall x \in \partial\Omega$$




**Domain for dilution condition**

**Submodel**

FE modeling : $\boldsymbol{Kq} = \boldsymbol{F}$

$$\boldsymbol{u}(x;I) = \sum_i \boldsymbol{\varphi}_i(x;I)\; q_i(I) + E\,x + \boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)$$

FE fluctuation
+ macro strain
+ filting of boundary condition
+ periodic coefficients

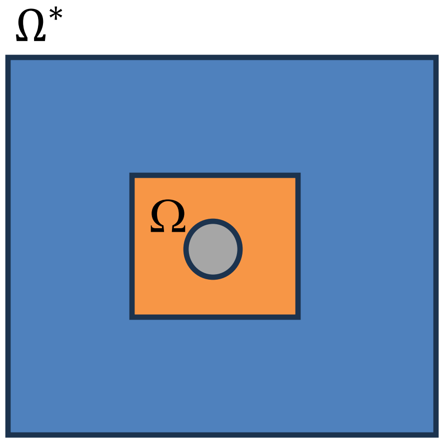
$$K_{ij}(x;I) = \int_{\Omega(\mathrm{I})} tr\Big(\varepsilon(\boldsymbol{\varphi}_i)\; \boldsymbol{C}(\boldsymbol{x};\boldsymbol{I})\; \varepsilon\big(\boldsymbol{\varphi}_j\big)\Big)\; d\Omega$$

$$\tilde{\alpha}\, \|\boldsymbol{q}\|_2{}^2 \leq \boldsymbol{q}^T \boldsymbol{K}\, \boldsymbol{q} \;\leq\; \tilde{\gamma}\, \|\boldsymbol{q}\|_2{}^2$$

$$F_i(I) = -\int_{\Omega(\mathrm{I})} tr\Big(\, \varepsilon(\boldsymbol{\varphi}_i)\; \boldsymbol{C}(\boldsymbol{x};\boldsymbol{I})\; \Big(\, E + \varepsilon\Big(\, \boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)\Big)\Big)\Big)\; d\Omega \;\; = (\boldsymbol{F_o} + \boldsymbol{B}\boldsymbol{u}_D)_{\boldsymbol{i}}$$

# Theory: void modeling in dilution condition and elasticity

$$\boldsymbol{u}(x;I) = \sum_i \boldsymbol{\varphi}_i(x;I)\ q_i(I) + E\,x + \boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)$$




Zone of interest

Application of **Saint-Venant's principle**

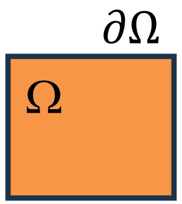
Under dilution conditions, for $\Omega^*$ sufficiently large and far from an area of interest, the solution $\boldsymbol{u}$(x;I) is independent of the lifting $\boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)$. (See asymptotic properties of Green's functions [Blanc et al. 2013]*.)

Definition : $\boldsymbol{u}_D(I)$ is related to the solution under dilution condition restrained to a given sub-domain Ω.

*[X. Blanc et al. 2013]

# Theory: void modeling in dilution condition and elasticity

$$\boldsymbol{u}(x;I) = \sum_i \boldsymbol{\varphi}_i(x;I)\ \ q_i(I) + E\ x + \ \boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)$$




Deep learning predictions:

$$\boldsymbol{u}_{ANN}(I) \text{ on } \partial\Omega,$$
$$\boldsymbol{r}_{ANN}(I)\ over\ \Omega$$

Errors $\quad e_D(I) = \ \|\boldsymbol{u}_D - \ \boldsymbol{u}_{ANN}\ \|_{\partial\Omega}$

$$e_r(I) = \ \|r(\boldsymbol{u}_D) - \ \boldsymbol{r}_{ANN}\ \|_{\Omega}$$

$$e_u(I) = \ \|\boldsymbol{u} - \ \boldsymbol{u}(\boldsymbol{r}_{ANN})\ \|_{\Omega}$$

Right hand side term:

$$\delta\boldsymbol{F} = \ \boldsymbol{F}(I) - \ \boldsymbol{F}_{ANN}\ , F_{ANNi}(I) = -\int_{\Omega(\mathrm{I})} tr\Big(\ \varepsilon(\boldsymbol{\varphi}_i)\ \boldsymbol{C}(\boldsymbol{x};\boldsymbol{I})\ \big(\ E + \varepsilon\big(\ \boldsymbol{r}_{ANN}(I)\big)\ \big)\Big)\ d\Omega$$

# Theory: void modeling in dilution condition and elasticity

$$\boldsymbol{u}(x; I) = \sum_i \boldsymbol{\varphi}_i(x; I)\ \ q_i(I) + E\ x + \boldsymbol{r}\big(\boldsymbol{u}_D(I)\big)$$


**Convergence property**

$$\frac{e_u}{\|\boldsymbol{u}\|_\Omega} \leq \sqrt{\frac{\tilde{\gamma}}{\tilde{\alpha}}} \frac{\|\delta \boldsymbol{F}\|_2}{\|\boldsymbol{F}\|_2} \leq \sqrt{\frac{\tilde{\gamma}}{\tilde{\alpha}}} \frac{\|\boldsymbol{B}\|_2}{\|\boldsymbol{F}\|_2} \|\boldsymbol{u}_D - \boldsymbol{u}_{ANN}\|_{\partial\Omega}$$

The better the prediction $\boldsymbol{u}_{ANN}(I)$ on $\partial\Omega$, the better the submodel prediction.

The smaller the elements in the submodel, the larger $\tilde{\gamma}$ and the upper bound.

# Data generation using a FE model

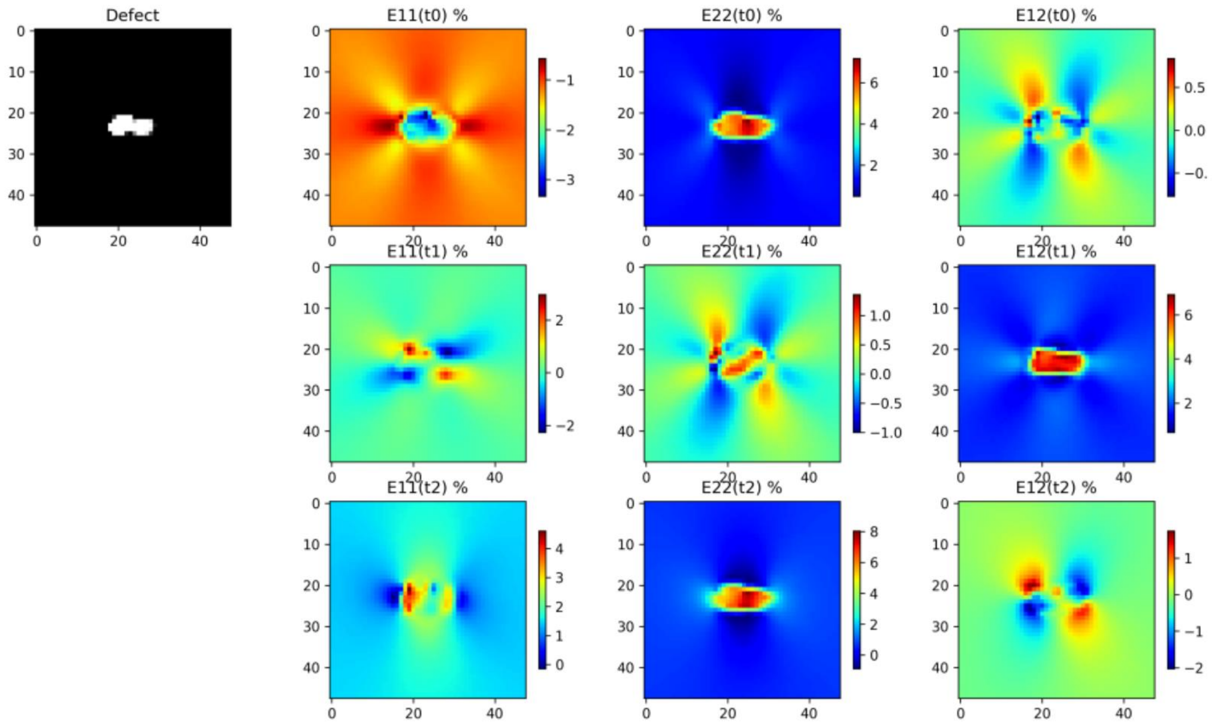
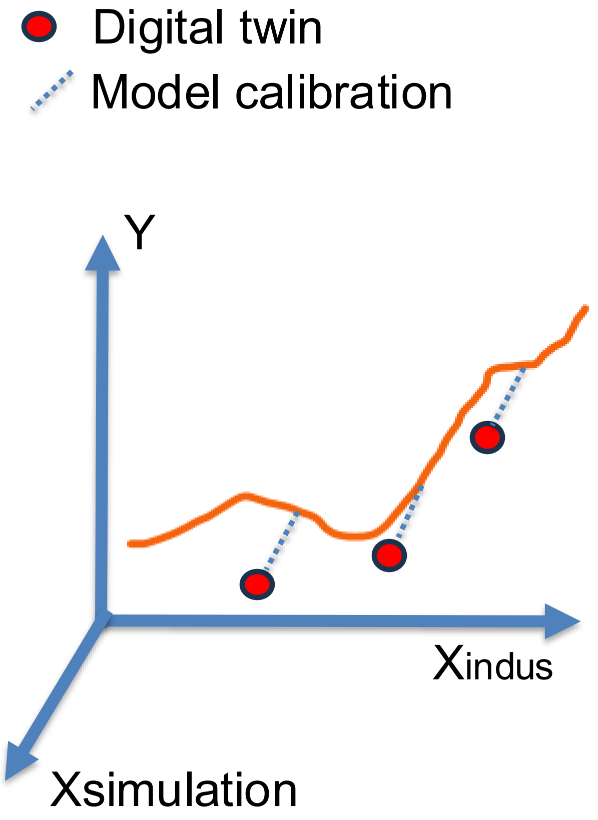



**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_{\square}$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)}$. Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.



# Data augmentation

Multimodal data augmentation for digital twining assisted by artificial intelligence in mechanics of materials (2022)

HPC Computer-generated Data

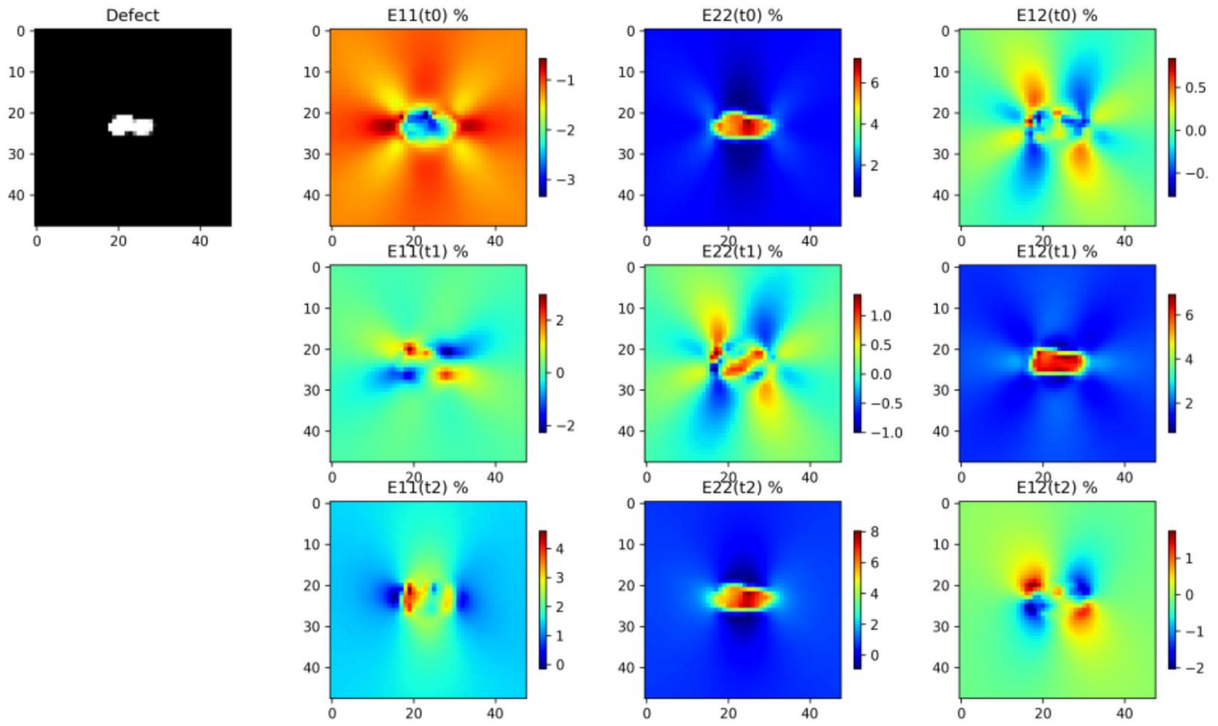
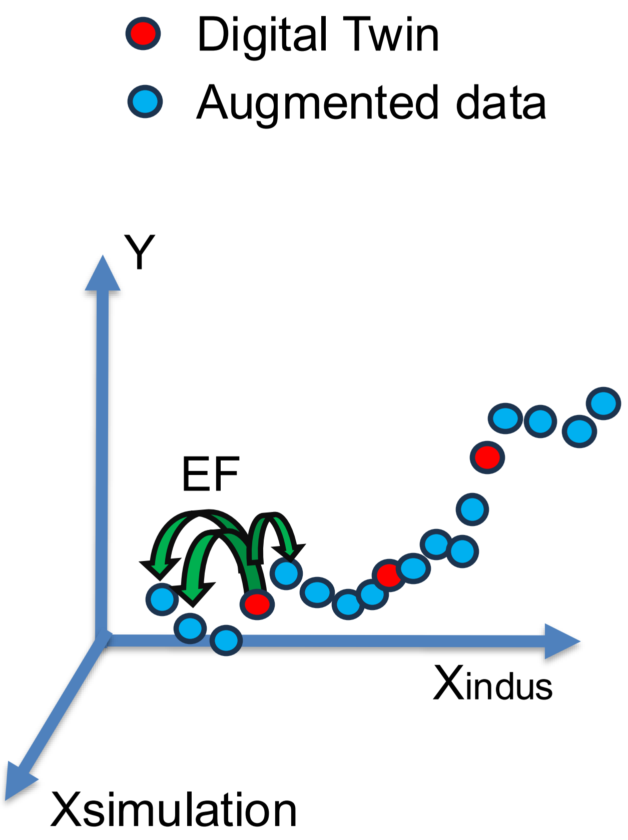
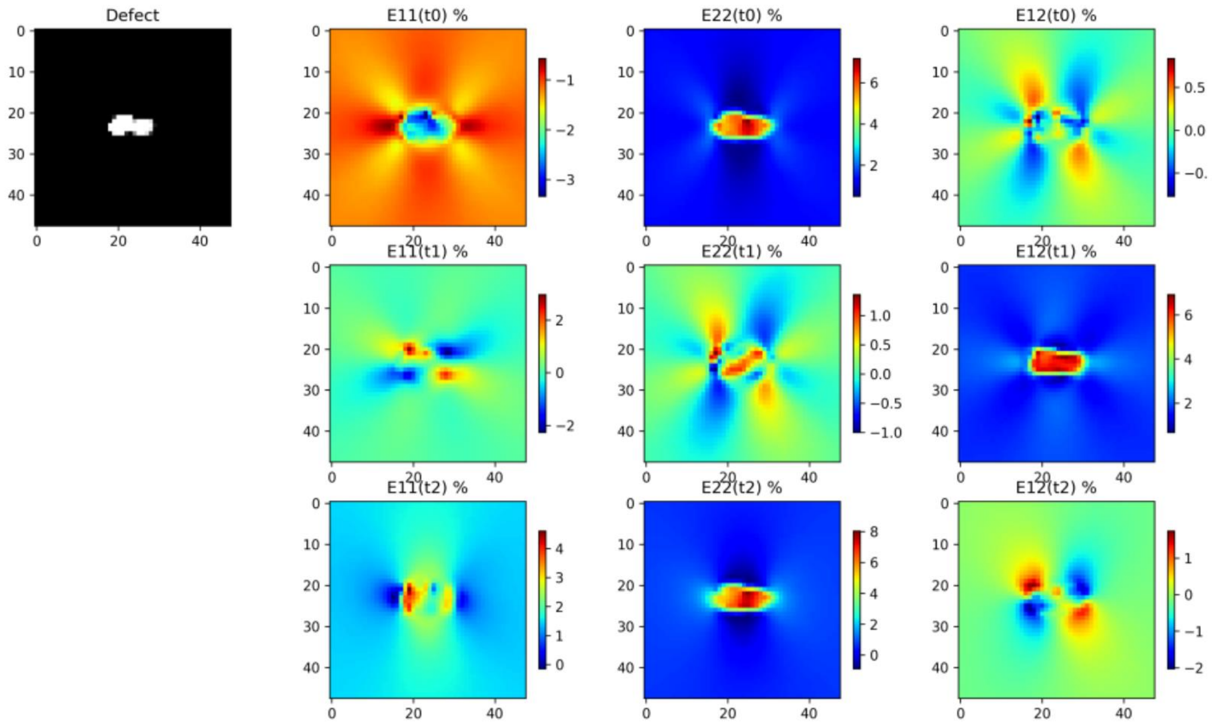


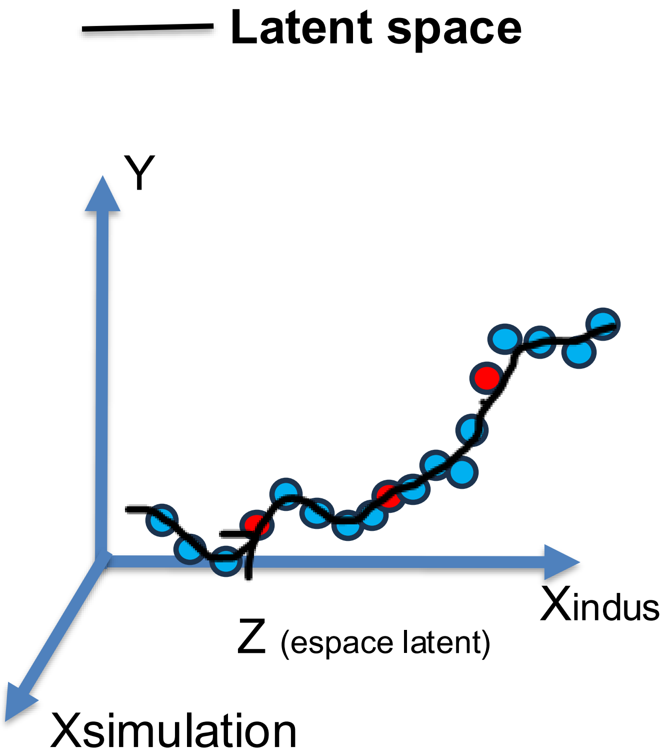

**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_\square$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)}$.
Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.



# Self-supervised machine learning

Multimodal data augmentation for digital twining assisted by artificial intelligence in mechanics of materials (2022)




Selfsupervised Learning

Xmultimodal = ( Xindus, Xsimulation, Y)

Z reduced coordinate

# Down stream task: Cross modal prediction

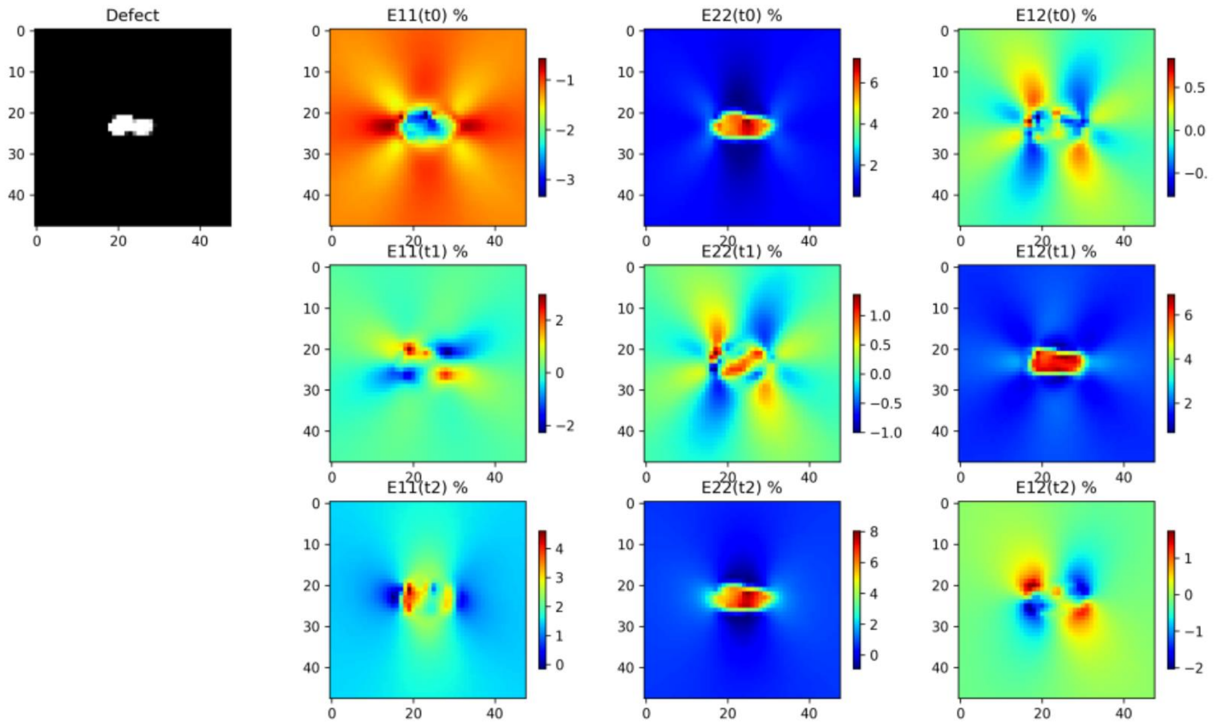
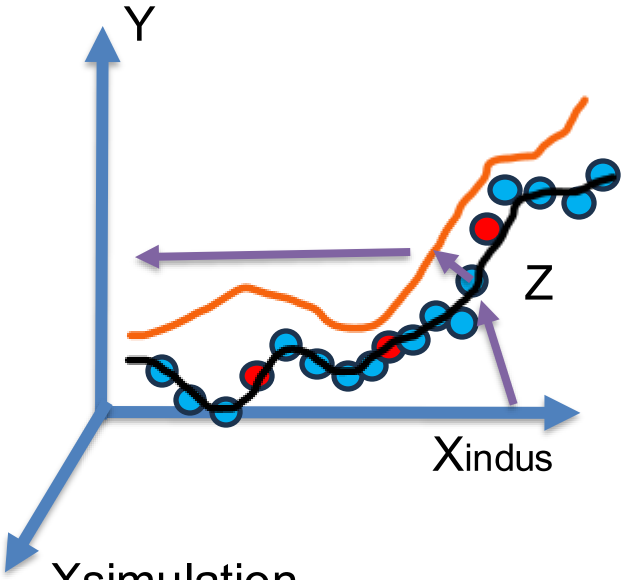



**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_\square$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)}$
Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.

Metamodeling

**Downstream task**
Regression Y(X).



# Down stream task: Cross modal prediction

Metamodeling

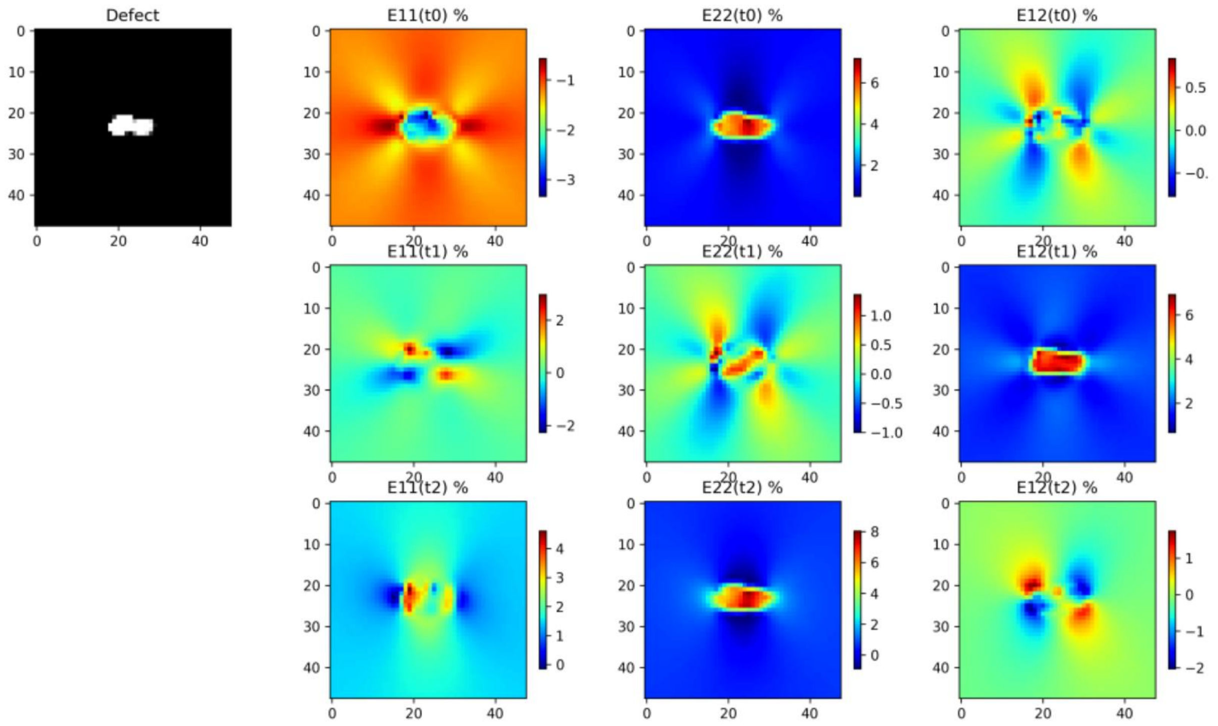
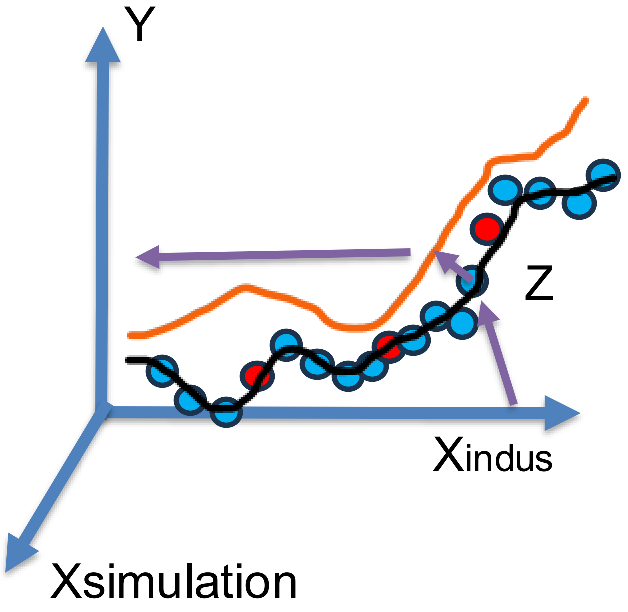



**Figure 6.** Strain components of the LROM for digital twin $i = 1$, on top left $\boldsymbol{\mu}_{\square}$, on the right $\boldsymbol{\varepsilon}(\mathbf{V}^{(1)}$.
Here, $t0,\ t1,\ t2$ stand for the indices of the vectors that span the LROM.

**Downstream task**
Regression Y(X).

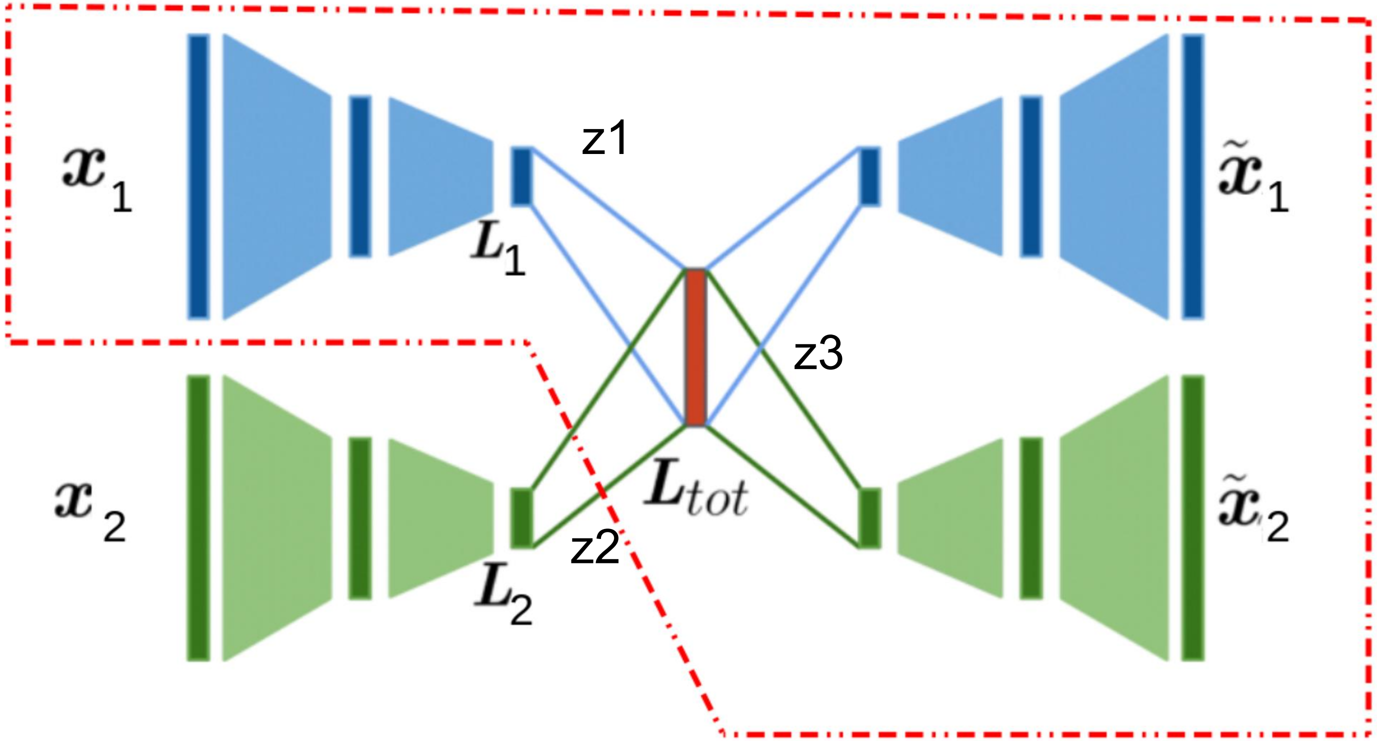


# Model architecture for self-supervised machine learning on multimodal data

Dual encoder architecture

Material Image

Strain fields

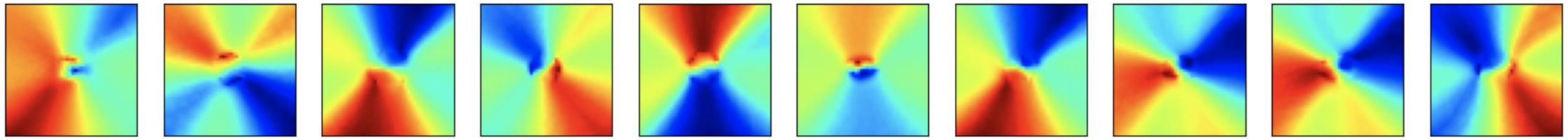


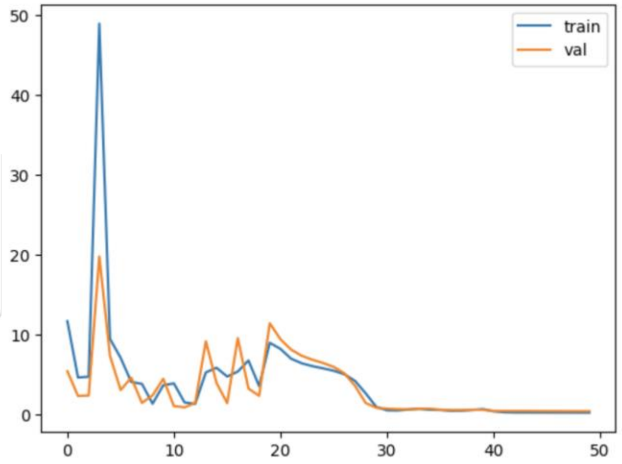

z3 = (z1+z2)/2
Ltot = z1-z2


[Deep multimodal autoencoder for crack criticality assessment](#)
Hugo Launay , David Ryckelynck , Laurent Lacourt , Jacques Besson , Arnaud Mondon et al.
*International Journal for Numerical Methods in Engineering*, 2022, 123 (6), pp.1456-1480. ⟨10.1002/nme.6905⟩

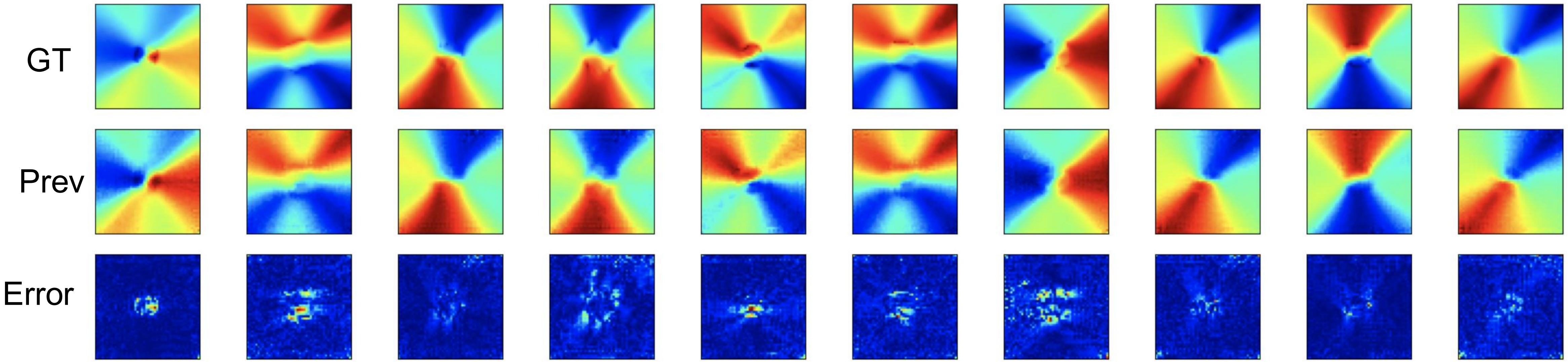
# Mechanical autoencoder for strain fields

Train dataset : 3000 strain fields



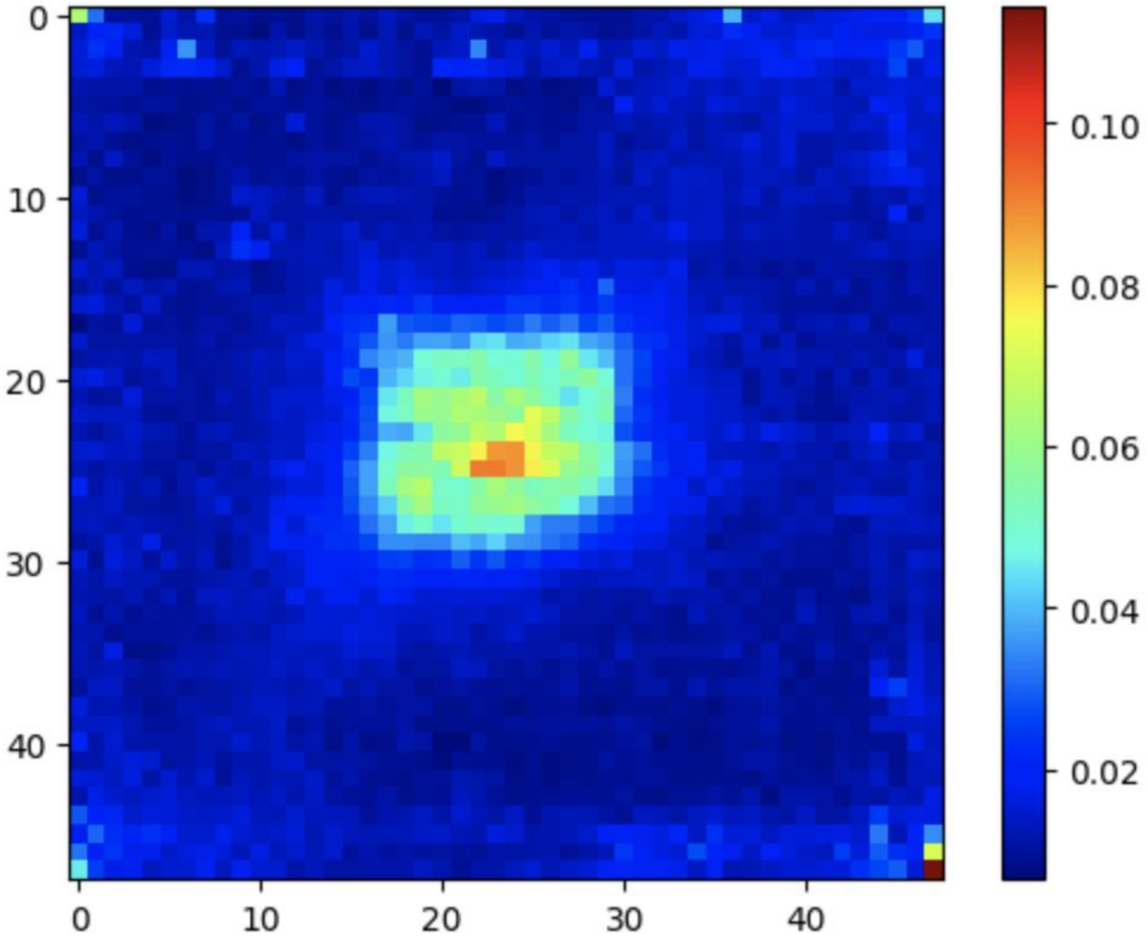



Test



# Mechanical autoencoder for strain fields

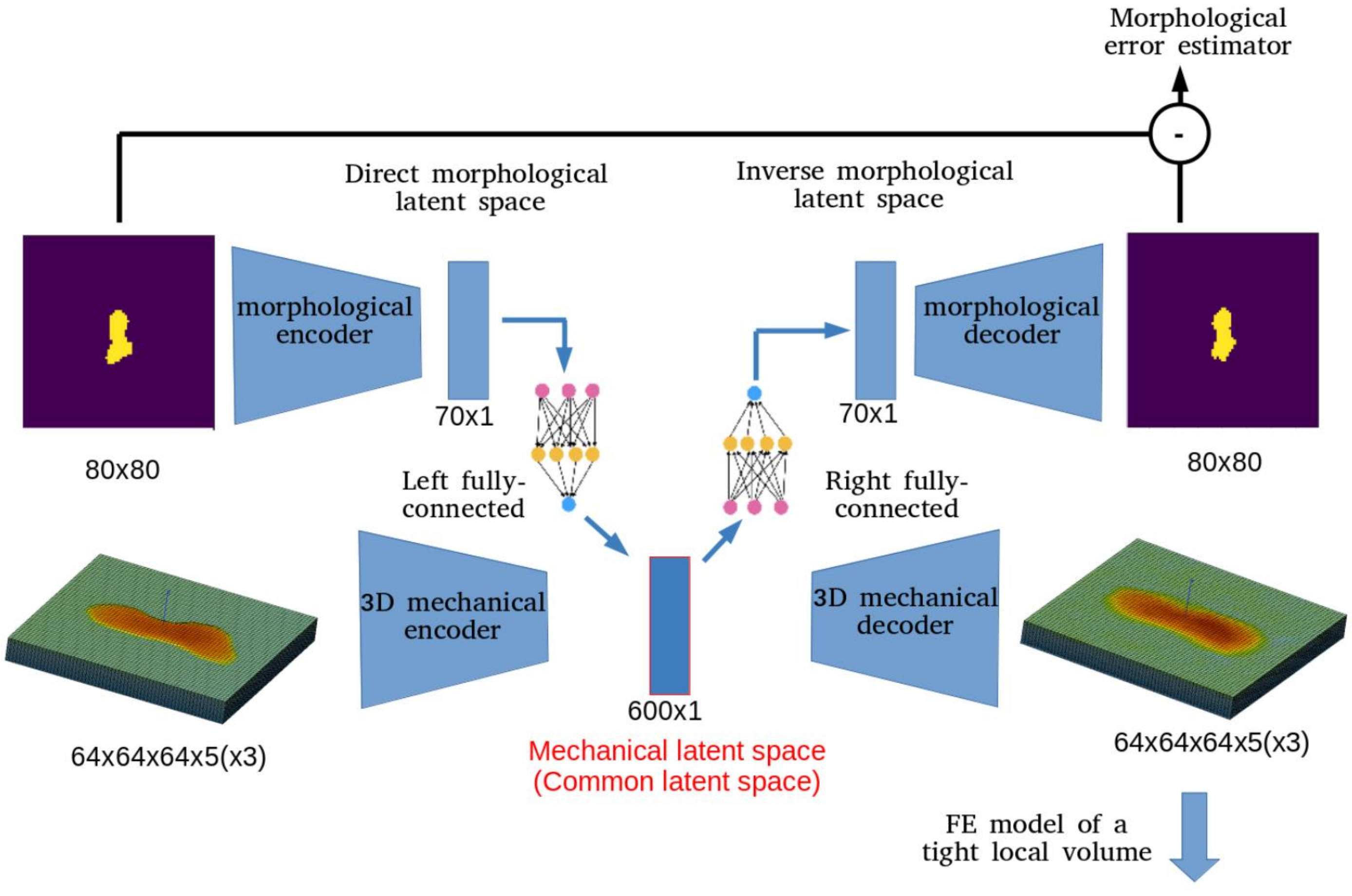
Map of local mean errors

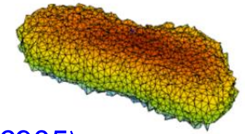



Fully consistent with Saint-Venant's principle and FE submodeling.

# Multimodal autoencoder for FE submodeling

# Learning boundary condition for PDE



Ω : Tight Local Volume (TLV)

[**Deep multimodal autoencoder for crack criticality assessment**](#)
Hugo Launay , David Ryckelynck , Laurent Lacourt , Jacques Besson , Arnaud Mondon et al.
*International Journal for Numerical Methods in Engineering*, 2022, 123 (6), pp.1456-1480. ⟨10.1002/nme.6905⟩

Figure 6: Modified MM Capture d'écran d error indicator to generate displacement field on TLV.

# Multimodal autoencoder for FE submodeling

## Digital Twin

2D crack in 3D volume

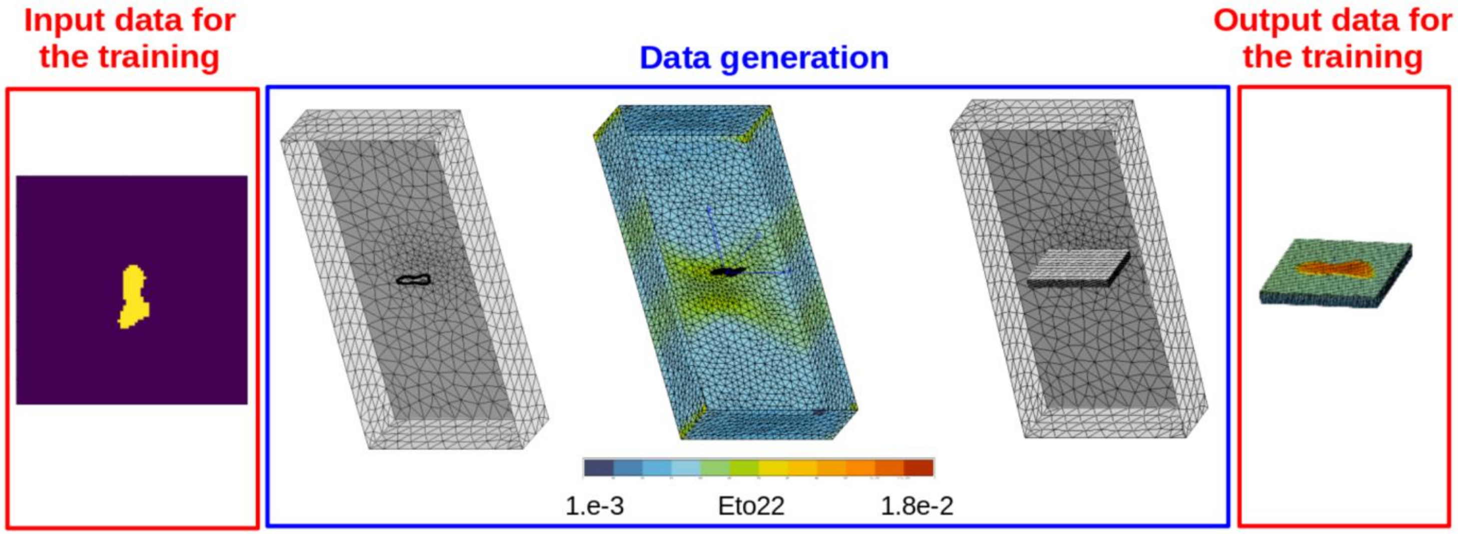



Figure 4: (a) 2D image representing the crack, (b) insertion of the 2D crack in the 3D mesh, (c) $\underset{\sim}{E}_{22}$ field, (d) position of the encoding mesh in the full structure and (e) the encoding mesh with the transferred information.

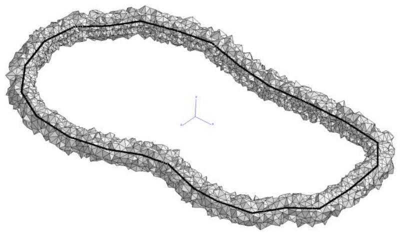
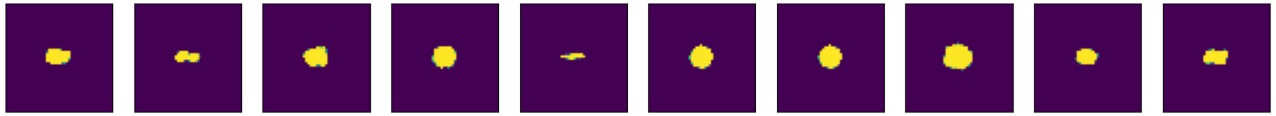


Figure 29: Smaller TLV for defect 1054

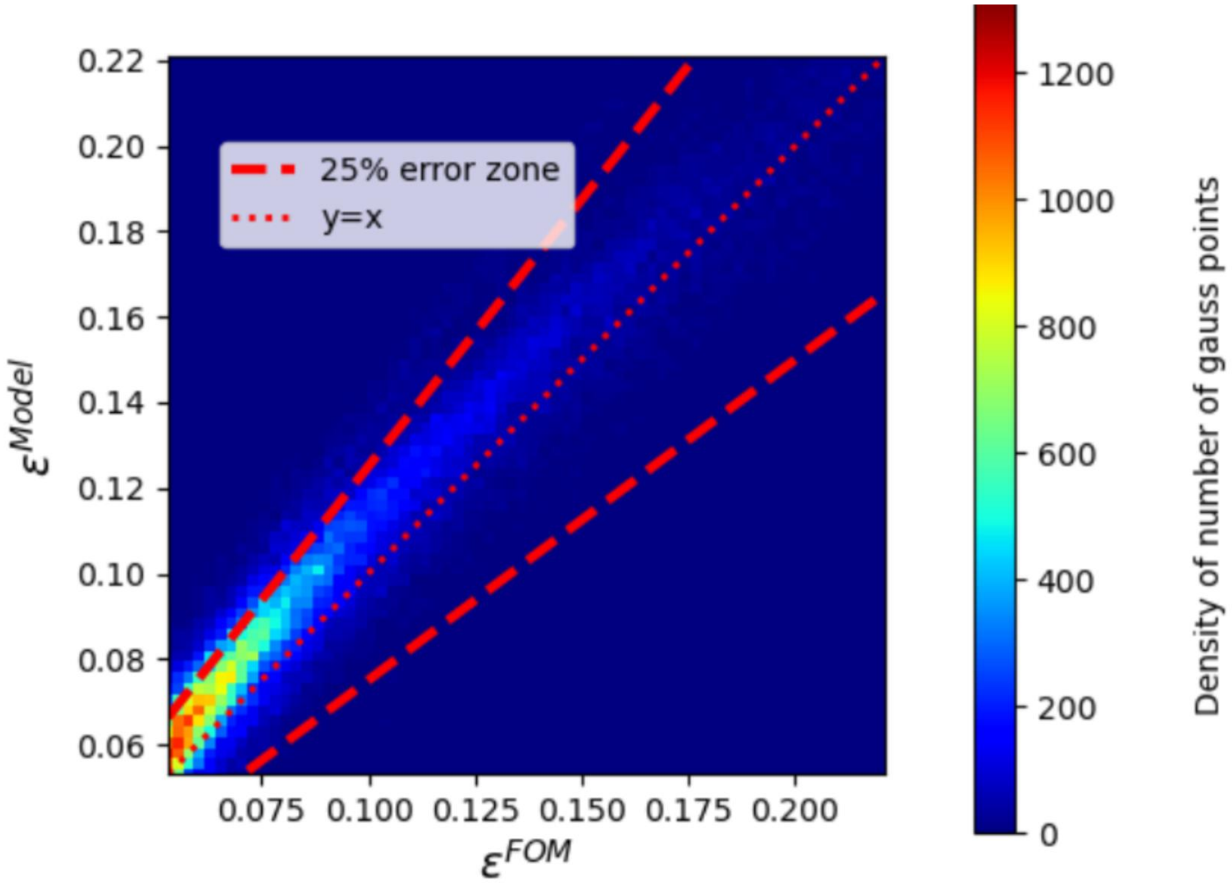
Tight Local Volume

Train dataset



# Multimodal autoencoder for FE submodeling

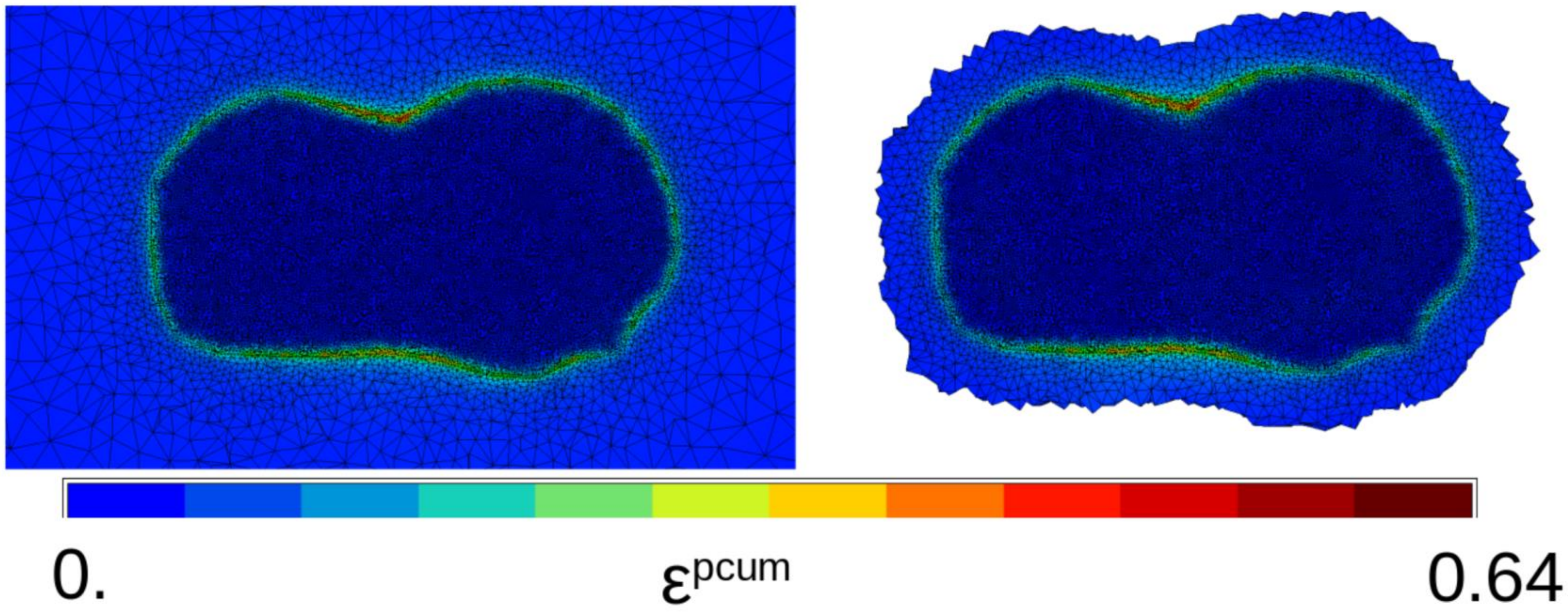
## A posteriori error

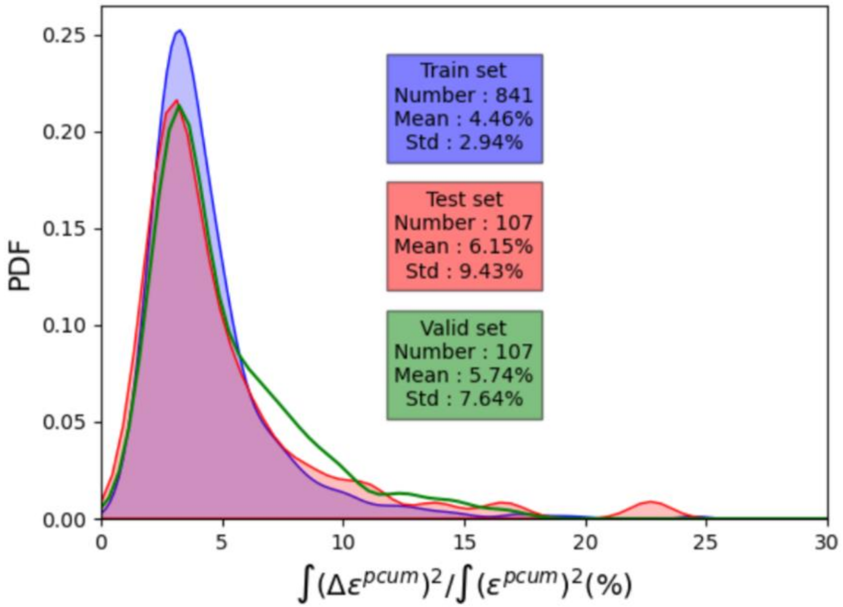


# Multimodal autoencoder for FE submodeling

## Generalisation error

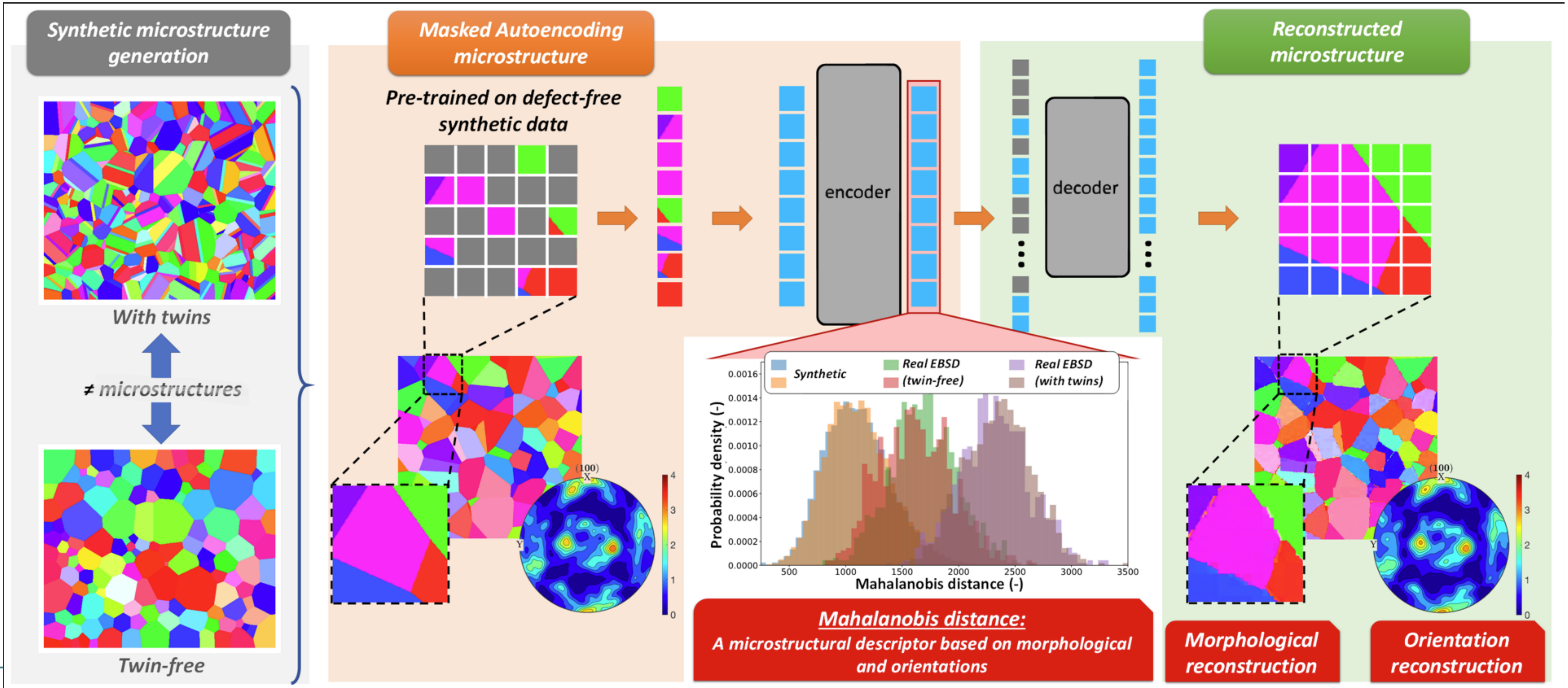
Plasticity, isotropic hardening, speedup = 10



# Anomaly detection using autoencoder

# In crystal plasticity



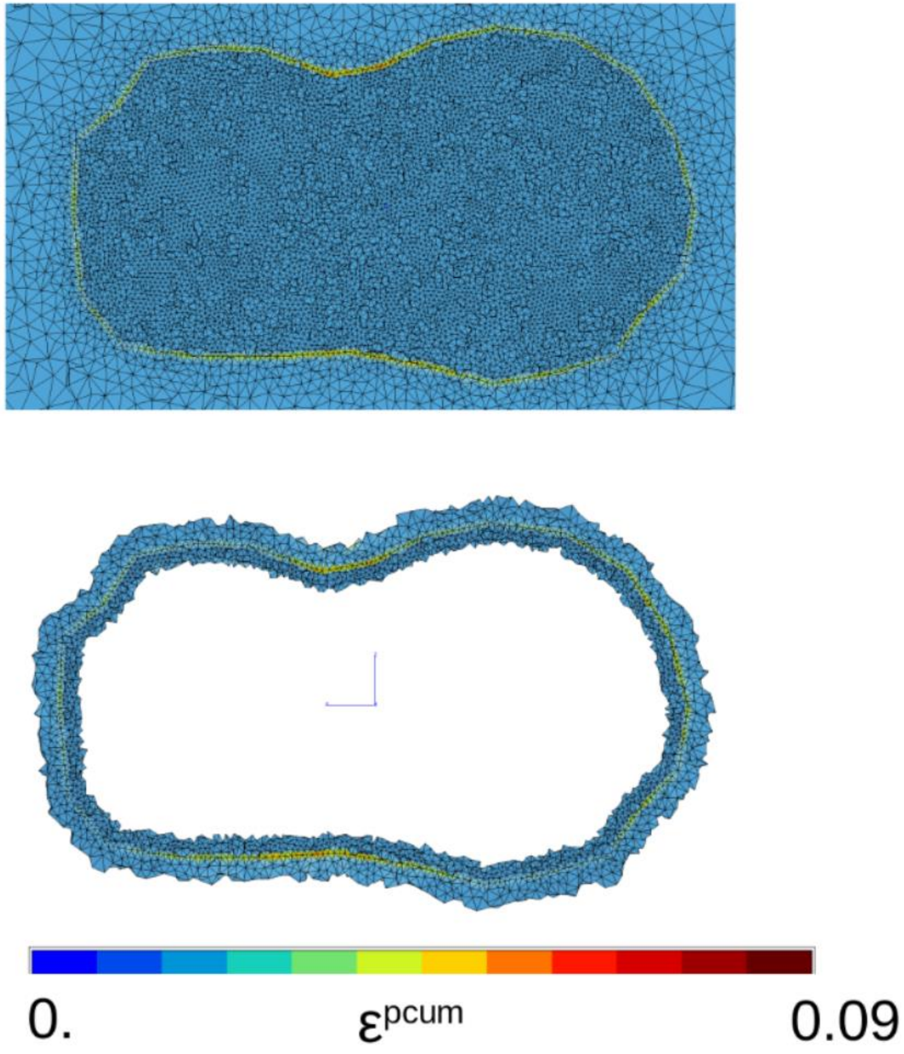


**Quaternion-Based Vision-Transformer for Polycrystalline EBSD Scans Pre-Trained on Large-Scale Synthetic Data**

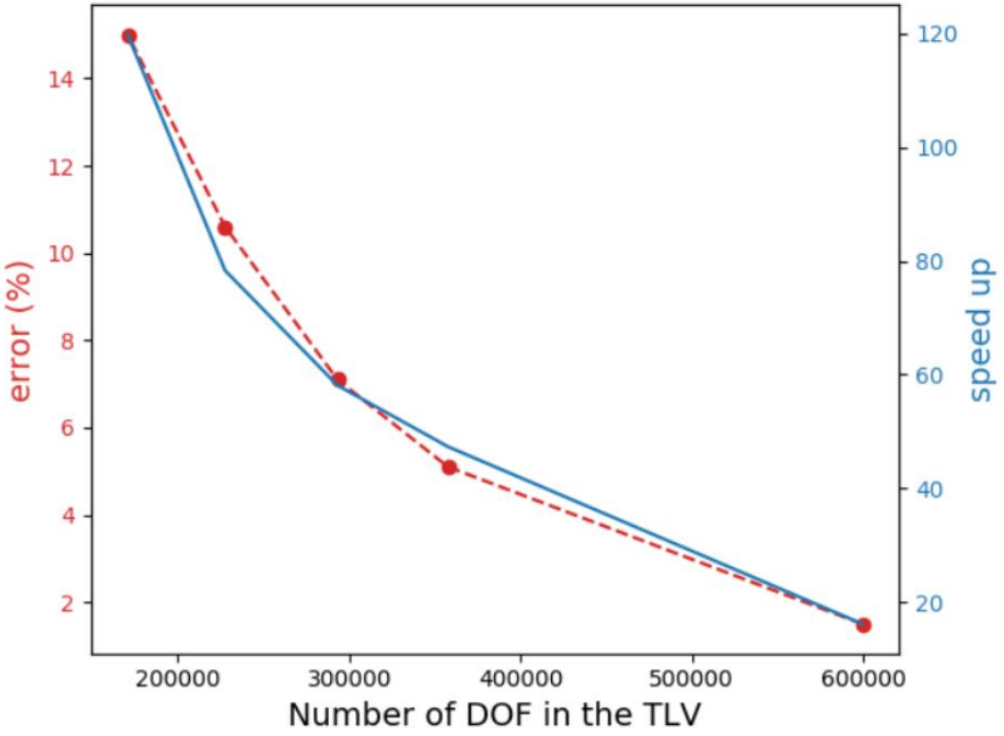
Pierre Belamri , Henry Proudhon , Damien Texier , David Ryckelynck

*Materials & Design*, 2025, 258, pp.114599. ⟨10.1016/j.matdes.2025.114599⟩

# Conclusion




**Toward XAI using FE submodels**




(b)

# Numerical practice

Requirement : Internet access and Google Drive

Create a working directory on your Google Drive.
Download Load_PyTorch_Convolutional_Autoencoder_Solution.ipynb in this directory.
Open this file with Google Collaboratory
Run Load_PyTorch_Convolutional_Autoencoder_Solution.ipynb

Download PyTorch_Convolutional_Autoencoder_Solution.ipynd
Open

Work to do :
Compute the reconstruction error on the boundary of a squarre of dimension a=[5,...25].
Plot the average error on this boundary as a function of a.
Comment with respect to de Saint Venant's Principle