A heteroencoder architecture for prediction of failure locations in porous metals using variational inference

Wyatt Bridgman[1], Xiaoxuan Zhang[2], Greg Teichert[2], Mohammad Kahlil[1]
Krishna Garikipati[2], Reese Jones[1]

[1] *Sandia National Laboratories, Livermore, CA*
[2] *University of Michigan, Ann Arbor, MI*

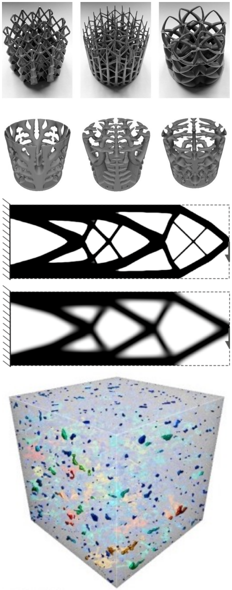
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

March 18, 2022

boratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LL
Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE

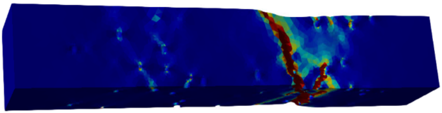
Motivation: AM porosity and failure

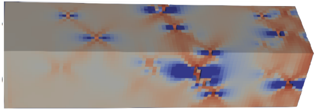
- **Additive manufacturing (AM)** allows design of components with more complex structure than traditional techniques.
- Often combined with **topology optimization** to create optimally performing components.
- AM parts typically suffer from **porosity** issues that induce failure in a complex manner.
- How to address this issue with measurement and/or modeling techniques?



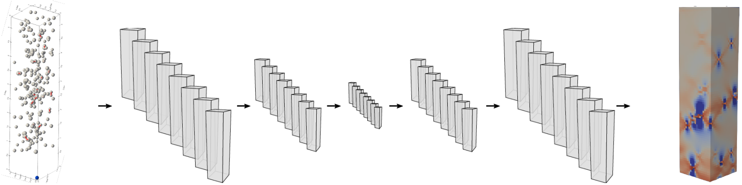
Goal: predicting failure locations

- Tools exist for measuring porosity in AM materials like Computed Tomography (CT) but failure highly sensitive to void locations.
- Want: model for reliably predicting failure from porosity. Accurate DNS models exist but are expensive:





- **Neural networks** have been used successfully as surrogates for a number of physical problems.



- Goal: train Convoutional Neural Network (CNN) on DNS failure model.

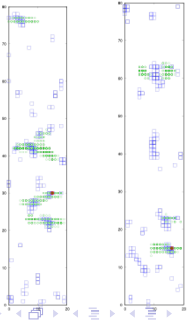
Failure model: overview

DNS model

- Calibrated plasticity model of AM 17-4 PH stainless steel to model material behavior affected by accumulation of damage.
- Stress linear, elastic $\boldsymbol{\sigma} = (1-\phi)\mathbb{C}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_p)$, with $\mathbb{C}$ isotropic elastic modulus tensor, $\boldsymbol{\epsilon}$ total strain, $\boldsymbol{\epsilon}_p$ plastic strain, ϕ void fraction.
- ϕ, $\boldsymbol{\epsilon}_p$ evolve according to complex system of ODEs modeling failure process.

Porosity realizations

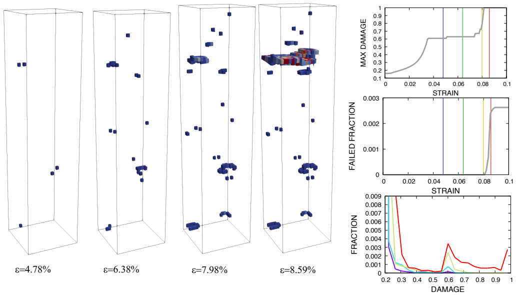
Pores in **blue**



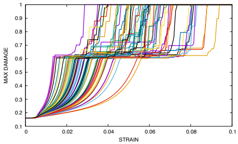
- **Small pores** are implicit in constitutive model.
- **Large pores** generated on mesh through Karhunen–Loève (KL) process with $\approx 12,000$ modes & power-exponential correlation function fit via CT scans.
- Many modes needed for high frequencies

Failure model: phenomenology & sensitivity to pores

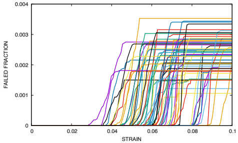
Damage evolution - representative example



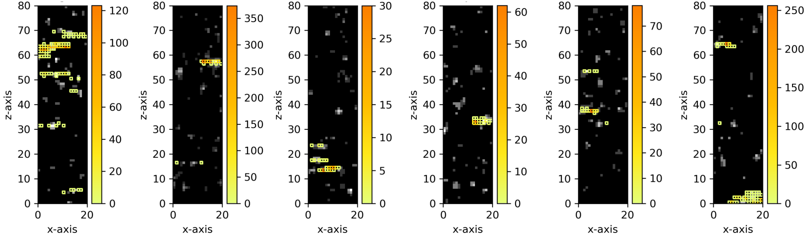
Max dmg.



Vol. fraction



Sensitivity - perturbed failure locations in **yellow (□)**



Neural network surrogate failure model

Machine learning model objectives

1. Construct CNN to predict failure locations.
2. Build comparative Bayesian CNN (BCNN) model to capture uncertainty and/or sensitivities.

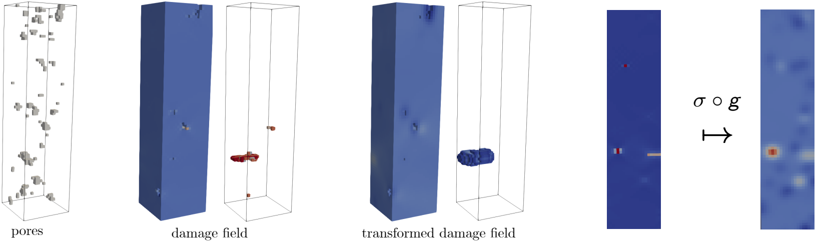
CNN model setup

Goal: Binary classification of failure $\Phi(\mathbf{X})$ given porosity $\varphi(\mathbf{X})$.
Issue: Failed to not-failed ratio $\approx 1:10^4 \longrightarrow$ Class Imbalance.

Initial **regularization** of classification problem by

- Recasting as regression of damage field $\phi(\mathbf{X}, t_{\text{fail}})$ at time of failure with data $\mathcal{D} = \{\varphi_i, \phi_i\}$ and MSE loss.
- Transforming damage $\phi' = \sigma \circ g(\phi)$ where σ **softmax** and g **Gaussian filter**.

Smoothing emphasizes low-freq. content consistent with latent space while **softmax** emphasizes high-damage regions of interest:

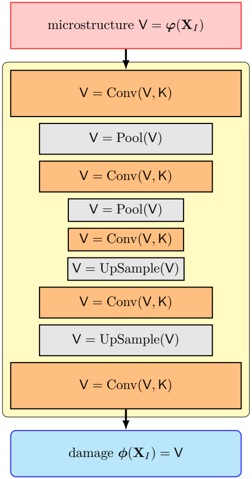


Dimensionality reduction & network architecture

Various aspects of model motivate **low-dimensional** latent space:

- KL expansion represents process with $\approx 12,000$ linear modes.
  - Motivates initial linear encoder-decoder structure $\phi = \mathsf{W}_2\mathsf{W}_1\varphi + \mathsf{b}$
- Can further reduce number of modes through nonlinear dimensionality reduction.
- Spatial smoothing via convolution layers reduces high-freq. content.
- Results in heteroencoder architecture with intermediate low-dimensional latent space layer of smaller dimension than output space.

CNN architecture



Class imbalance & loss re-weighting

- MSE loss $\frac{1}{N_v N_s} \sum_{s=1}^{N_s} \|\phi_s - \hat{\phi}_s\|^2$ in context of class-imbalance

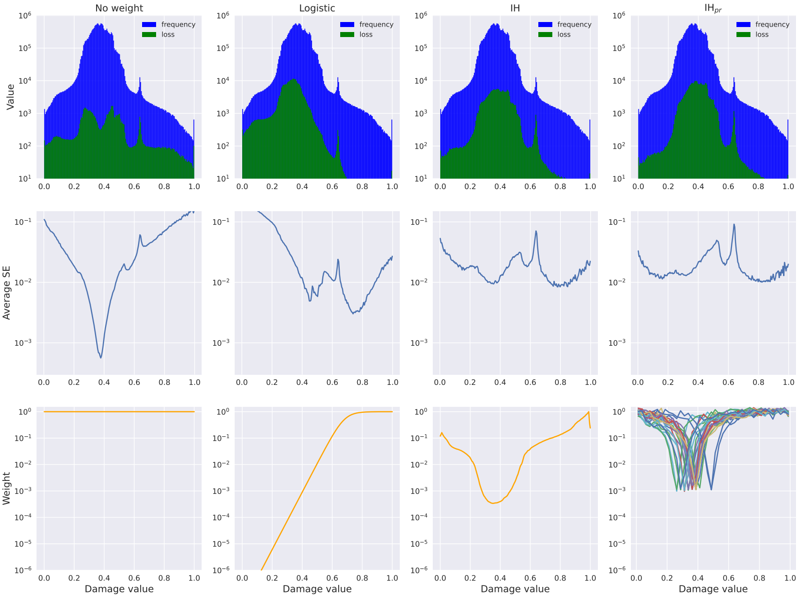
$$\frac{1}{N_v N_s} \left[\sum_{\phi_s, \hat{\phi}_s \in \mathcal{D}_{\text{low}}} \|\phi_s - \hat{\phi}_s\|^2 + \sum_{\phi_l, \hat{\phi}_l \in \mathcal{D}_{\text{high}}} \|\phi_l - \hat{\phi}_l\|^2 \right]$$

  where $|\mathcal{D}_{\text{low}}| \gg |\mathcal{D}_{\text{high}}|$ are partioned sets of low and high damage data.
- Optimizer tends to find poor local minimum.
- Re-weight loss function to address class imbalance

$$\frac{1}{N_v N_s} \sum_{s=1}^{N_s} \|\phi_s - \hat{\phi}_s\|^2 \, w(\phi_s)$$

  where $w(\phi)$ is the per-voxel weighting function of true damage values ϕ

Loss re-weighting effect on optimizer solution



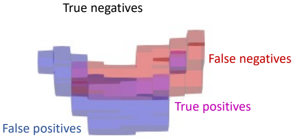
Overall network performance via overlap metrics

- Cluster overlap metrics as final performance metric over MSE.

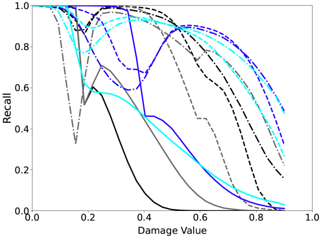
$$\text{Precision} = \frac{TP}{TP + FP}$$

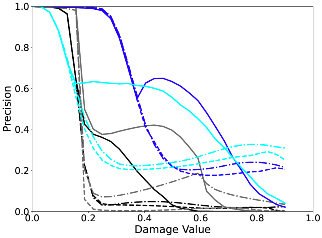
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Overlap} = \frac{TP}{TP + FP + FN}$$

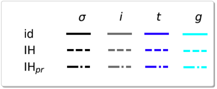


Data transformation	Notation	**Loss re-weightings**	Notation
None	id	None	id
Softmax	σ	Inverse histogram	IH
Gaussian filter	g	Inverse histogram per realization	IH_{pr}
Softmax ∘ Gaussian filter	t		





Recall: fraction of **true** values capture by predictions.



CNN reflects sensitivity of physics

- CNN learns sensitivity, i.e., alternate failure locations, in addition to main failure location via false positives:

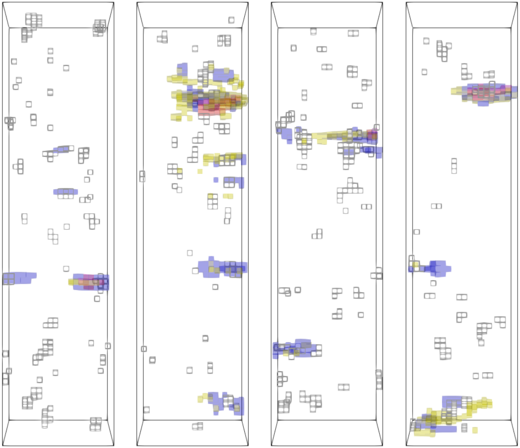


Figure: CNN prediction of four realizations. **Red**: true damage, **blue**: CNN prediction, **yellow** :failure locations from the sensitivity analysis, **gray**: pore location.

Bayesian Convolutional Neural Network (BCNN)

Why consider a Bayesian neural network model?

- **Uncertainty Quantification (UQ)** in the sense of capturing data sufficiency and (possibly) sensitivities.
- **Regularization** provided by the Bayesian prior distribution.

UQ with Bayesian inference

- UQ by treating model parameters as RVs whose distributions are calibrated to training data.
- Model $\mathrm{NN}_{\mathbf{w}}(\varphi)$, parameters $\mathbf{w}$, data $\mathcal{D} = \{\varphi_i, \phi_i\}$, then $\phi = \mathrm{NN}_{\mathbf{w}}(\varphi) + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ captures model discrepancy.
- Posterior probability of parameters $\mathbf{w}$ given $\mathcal{D}$ provided by Baye's rule:

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}$$

Approximate posteriors with variational inference (VI)

Posterior is intractable so we seek approximation $q_{\boldsymbol{\theta}}$ (often mean-field Gaussian) from known parametric family minimizing KL-divergence

$$q_{\boldsymbol{\theta}} = \min_{q_{\boldsymbol{\theta}} \in \mathcal{F}} D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{w}) \parallel p(\mathbf{w} \mid \mathcal{D}))$$

recast as minimizing negative Evidence Lower Bound (ELBO)

$$-\mathcal{L}(\boldsymbol{\theta}) = D_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\mathbf{w}) \parallel p(\mathbf{w})) - \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})}[\log p(\mathcal{D} \mid \mathbf{w})]$$

usually through a type of gradient descent.

Challenge with VI: non-convexity

- Non-convexity of ELBO loss leads to multiple local minima. Also reported in literature, addressed with strategies like annealing.
- Observed poor local minima inherited from MSE, i.e., see similar poor solution in deterministic CNN.
- Avoid by warm-starting means of params $\mathbf{w}$ from CNN solution.

Relationship between Bayesian and deterministic

BCNN has same network architecture as CNN but convolutional layers replaced by layers with random parameters. What's the difference?

- Likelihood loss component resembles MSE:

$$-\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})}\left[\log p(\mathcal{D} \mid \mathbf{w})\right] = c + \frac{1}{2\sigma^2}\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})}\left[\sum_{s=1}^{N_s} \|(\phi_s - \mathrm{NN}_{\mathbf{w}}(\varphi_s))\|^2\right]$$

- If $\mathrm{NN}_{\mathbf{W}}(\mathbf{x}) = \mathbf{W}\mathbf{x}$ linear, $\mathrm{mean}(\mathbf{W}) = \boldsymbol{\mu}_q$, $\mathrm{var}(\mathbf{W}) = \boldsymbol{\Sigma}_q$, ELBO is

$$-2\mathcal{L}_{\boldsymbol{\theta}} = \frac{1}{\sigma^2}\mathrm{tr}\{(\mathbf{Y} - \boldsymbol{\mu}_q\mathbf{X})^T(\mathbf{Y} - \boldsymbol{\mu}_q\mathbf{X})\} + (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)$$
$$+ \log\det(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p) + \mathrm{tr}(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_q) + \frac{1}{\sigma^2}\mathrm{tr}\{\mathbf{V}\mathbf{X}\mathbf{X}^T\}$$

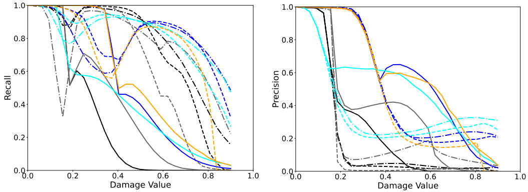
  which takes the form of least squares in means $\boldsymbol{\mu}_q$ with quadratic regularization. Variance $\boldsymbol{\Sigma}_q$ balanced between prior $\boldsymbol{\Sigma}_p$ and $\mathbf{0}$.

Non-linear case: view likelihood componentas as convolution with Gaussian: $\frac{N_s}{2\sigma^2}(\mathcal{N}(\mathbf{w} \mid \mathbf{0}, \boldsymbol{\Sigma}_q) * \mathrm{MSE}(\mathbf{w}))(\boldsymbol{\mu}_q) \Rightarrow$ inherit MSE local min.

Mean predictions and uncertainty distribution

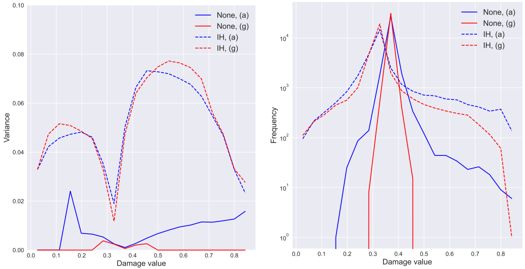
- Uncertainty in parameters pushed forward through model to get pushed forward posterior (PFP) distribution over outputs.
- Mean, variance of PFP estimated through Monte Carlo sampling.

Recall (left) and precision (right) of mean BCNN predictions in orange.



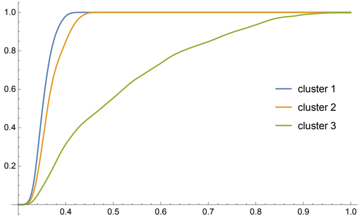
Distribution of variance w.r.t. damage value (left) and frequency w.r.t. damage value (right).

Uncertainty dist. reflects data sparsity. Consistent with Bayesian behavior.

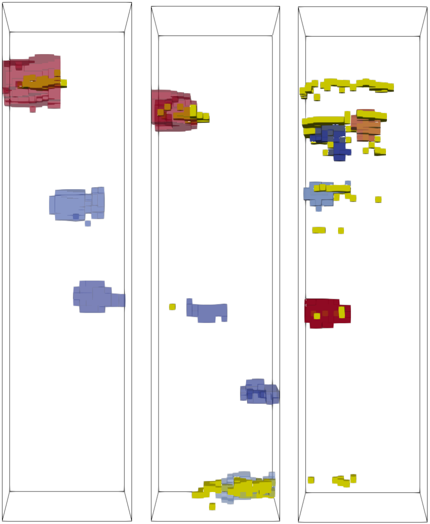


Leveraging uncertainty to rank damage clusters

- Compute empirical CDF to measure probability mass above a threshold and use this to rank clusters.
- Ranking reflects actual failure location and alternative locations via sensitivity analysis.



Empirical CDFs for 3 clusters



Red-high rank, **blue**-low rank, **yellow**-failures.