Data-Efficient Reliability Assessment Using Machine Learning: Application to Lifetime Estimation of Power Electronic Modules

Mehdi GHRABLI

SATIE & LMPS, ENS Paris-Saclay

Thesis committee

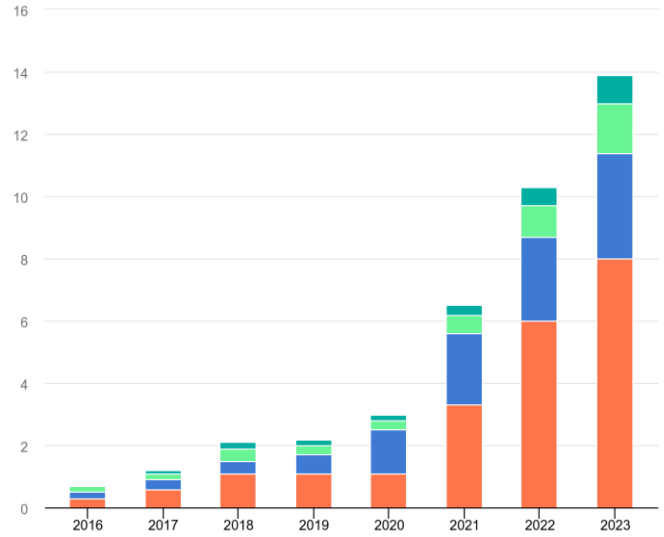
Emmanuelle ABISSET-CHAVANNE	Professor, École nationale supérieure d'arts et métiers	Reviewer
Huai WANG	Professor, Aalborg University	Reviewer
Zeina AL-MASRY	Associate professor, École nationale supérieure de mécanique et des microtechniques	Examiner
Bruno ALLARD	Professor, Institut national des sciences appliquées de Lyon	Examiner

Thesis supervisors

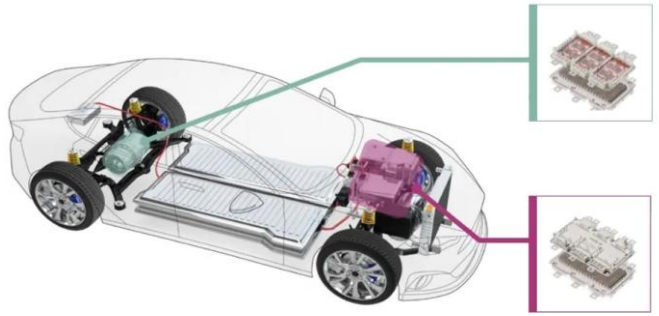
Mounira BOUARROUDJ	Senior researcher, SATIE, Université Gustave Eiffel	Supervisor
Ludovic CHAMOIN	Professor, LMPS, ENS Paris-Saclay	Co-supervisor
Emanuel ALDEA	Associate professor, SATIE, Université Paris-Saclay	Co-supervisor

PhD defense - ENS Paris-Saclay - 07/05/2026

- Urgency of changing the energy production landscape given the current climate conditions
- Effort towards electrification throughout legislations and production strategies:
  - LOM (No. 2019-1428 of December 24, 2019)
  - Fit for 55 package



Electric vehicles' sales growth from 2016 to 2023 (IEA 2023)



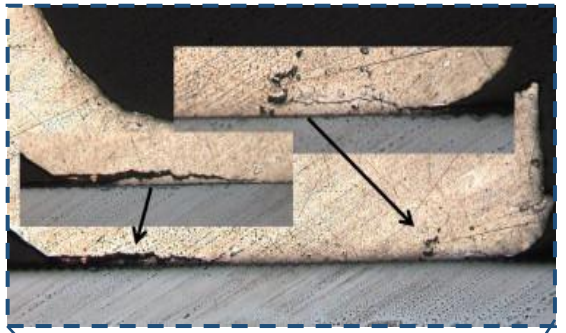
Power modules in transportation systems (Infineon 2024)

- Power electronic modules sit at the core of this electrification, namely in transportation:
  - Energy conversion
  - Modulation

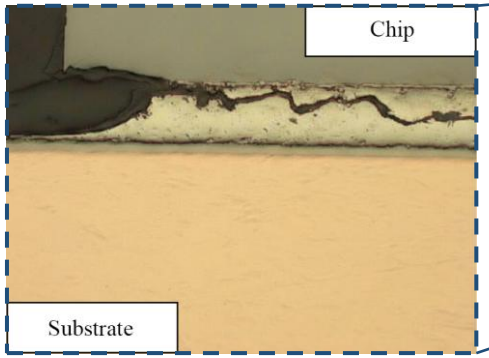
Global context Power modules Research question Methodology

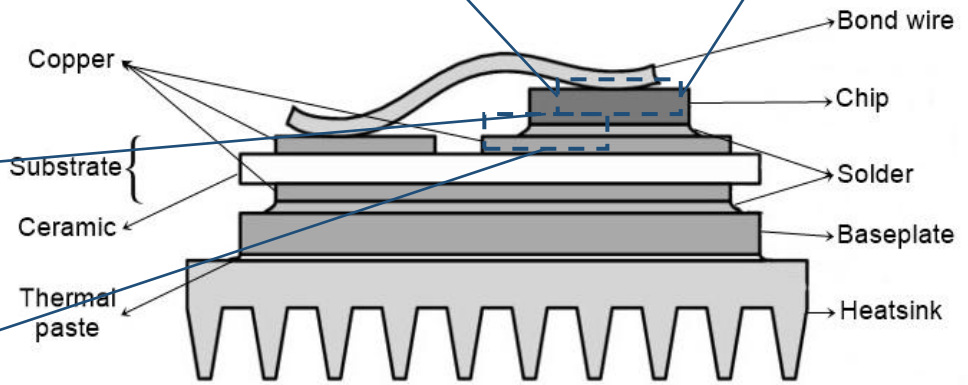
- High operation temperatures: Self-heating + external sources
- Creation of thermal stress
- Degradation at critical locations
  - Solder fatigue
  - Bond wire lift-off

Degradation of the bond wire (Schilling 2012)



Degradation at the solder layer (Deplanque 2006)



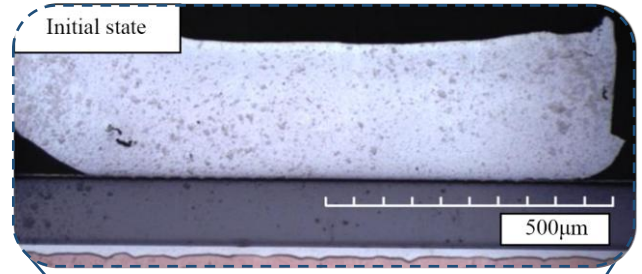


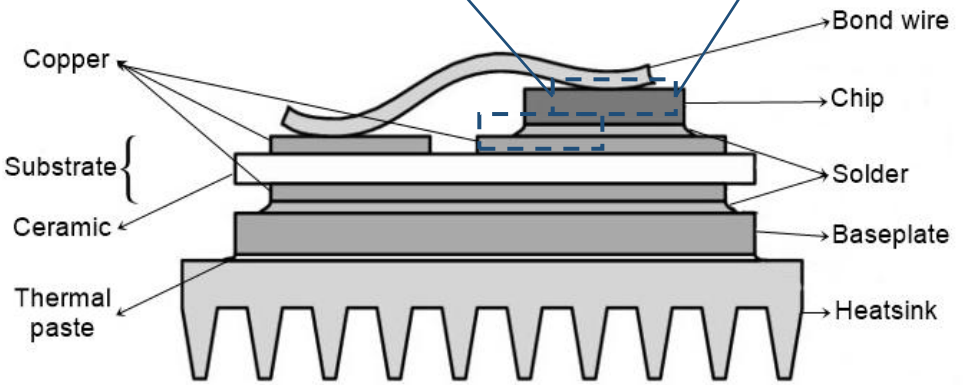
Architecture of a power electronic module (Buttay 2007)

Global context | Power modules | Research question | Methodology

- High operation temperatures: Self-heating + external sources
- Creation of thermal stress
- Degradation at critical locations
  - Solder fatigue
  - Bond wire lift-off
- Creation and propagation of a crack under repeated loading which leads to failure

Degradation states of the bond wire (Dornic 2019)



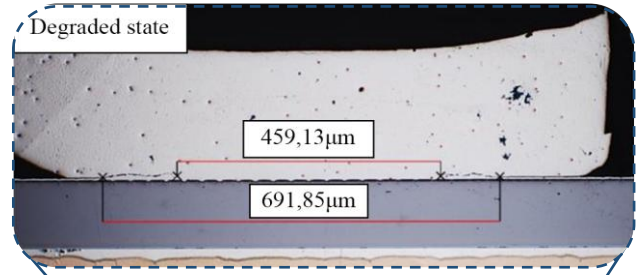


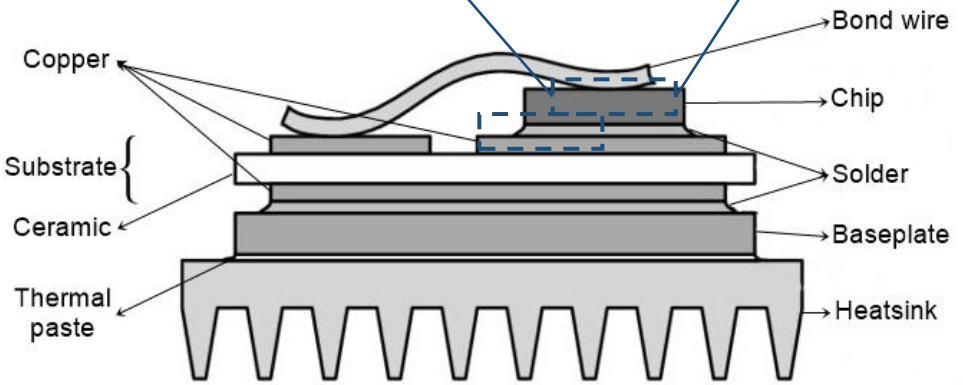
Architecture of a power electronic module (Buttay 2007)

Global context | Power modules | Research question | Methodology

- High operation temperatures: Self-heating + external sources
- Creation of thermal stress
- Degradation at critical locations
  - Solder fatigue
  - Bond wire lift-off
- Creation and propagation of a crack under repeated loading which leads to failure

Degradation states of the bond wire (Dornic 2019)



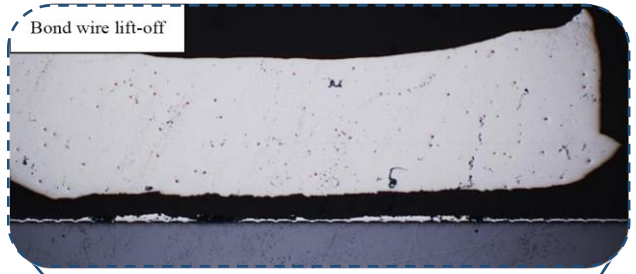


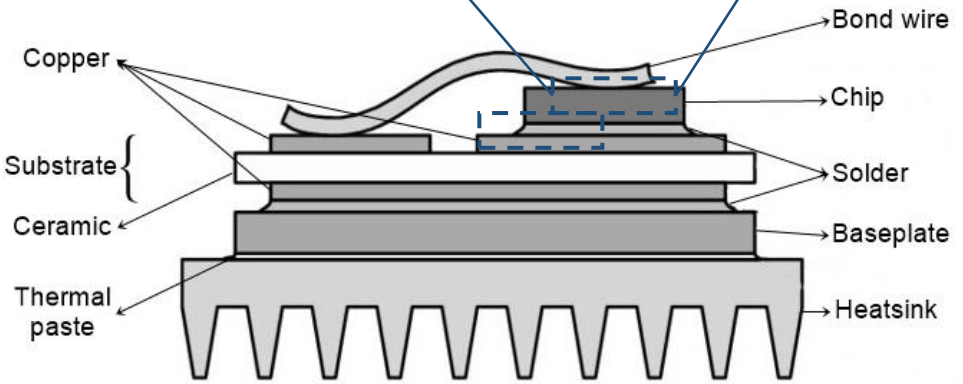
Architecture of a power electronic module (Buttay 2007)

Global context | Power modules | Research question | Methodology

- High operation temperatures: Self-heating + external sources
- Creation of thermal stress
- Degradation at critical locations
  - Solder fatigue
  - Bond wire lift-off
- Creation and propagation of a crack under repeated loading which leads to failure

Degradation states of the bond wire (Dornic 2019)

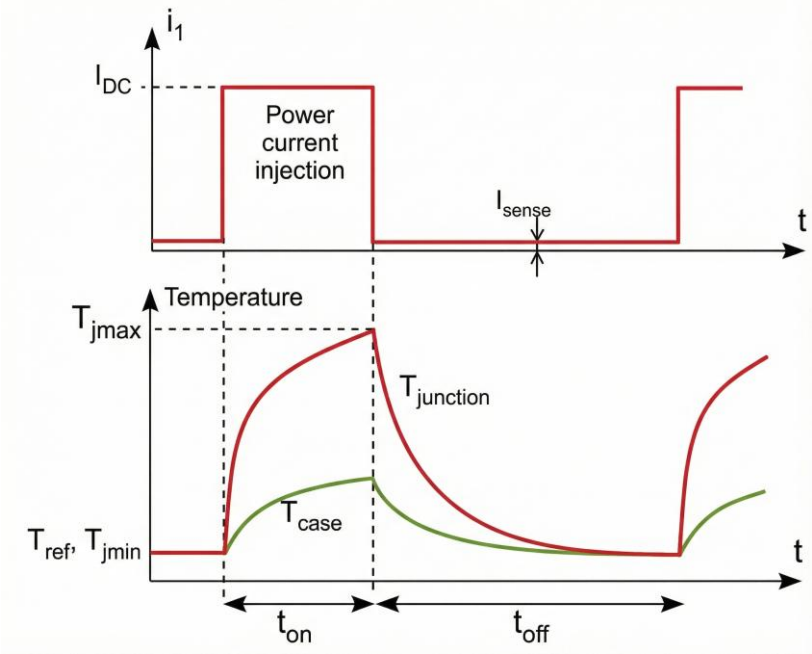




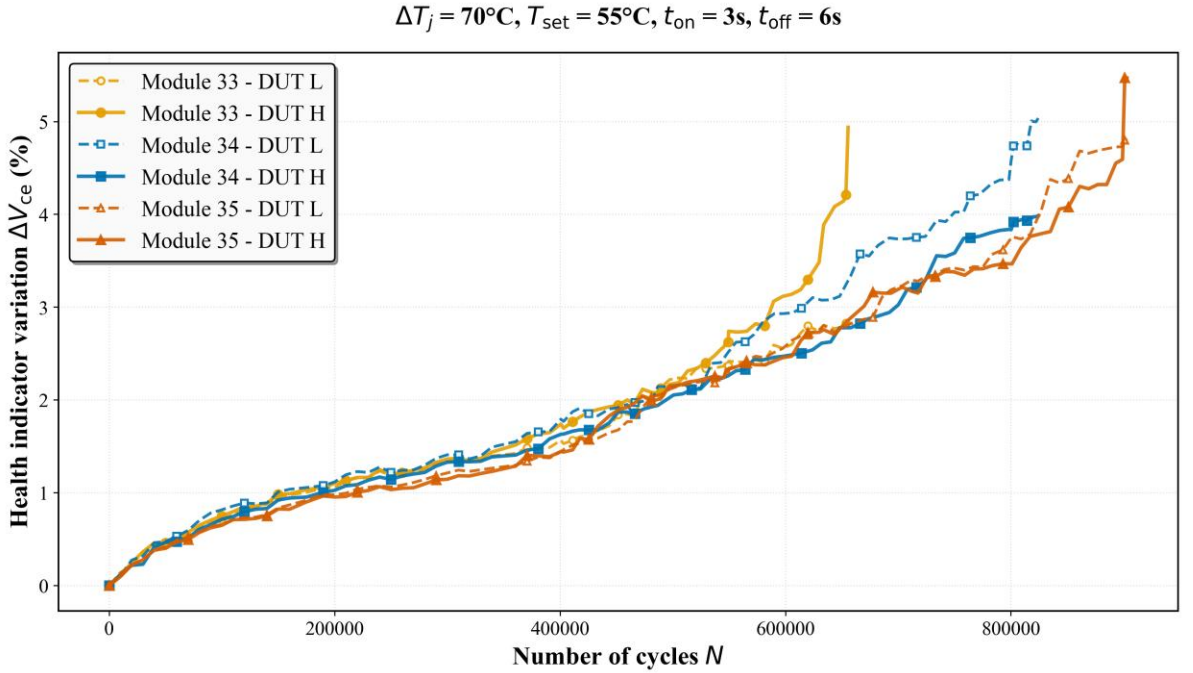
Architecture of a power electronic module (Buttay 2007)

Global context Power modules Research question Methodology

- Lifetime is studied experimentally through repeated loading and health monitoring
- Loading is amplified compared to real-life conditions to accelerate failure and observe lifetime within a reasonable window



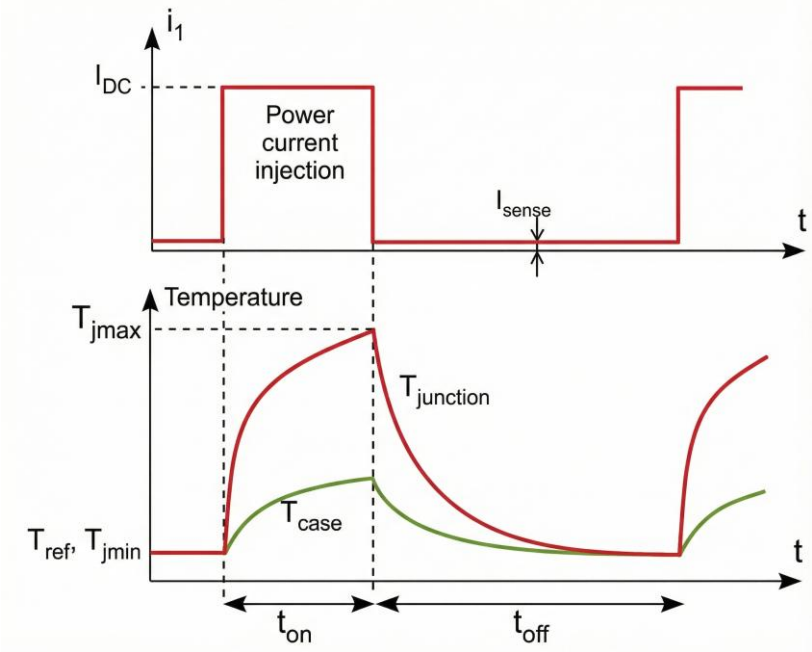
Profile of a single loading cycle (Dornic 2019)



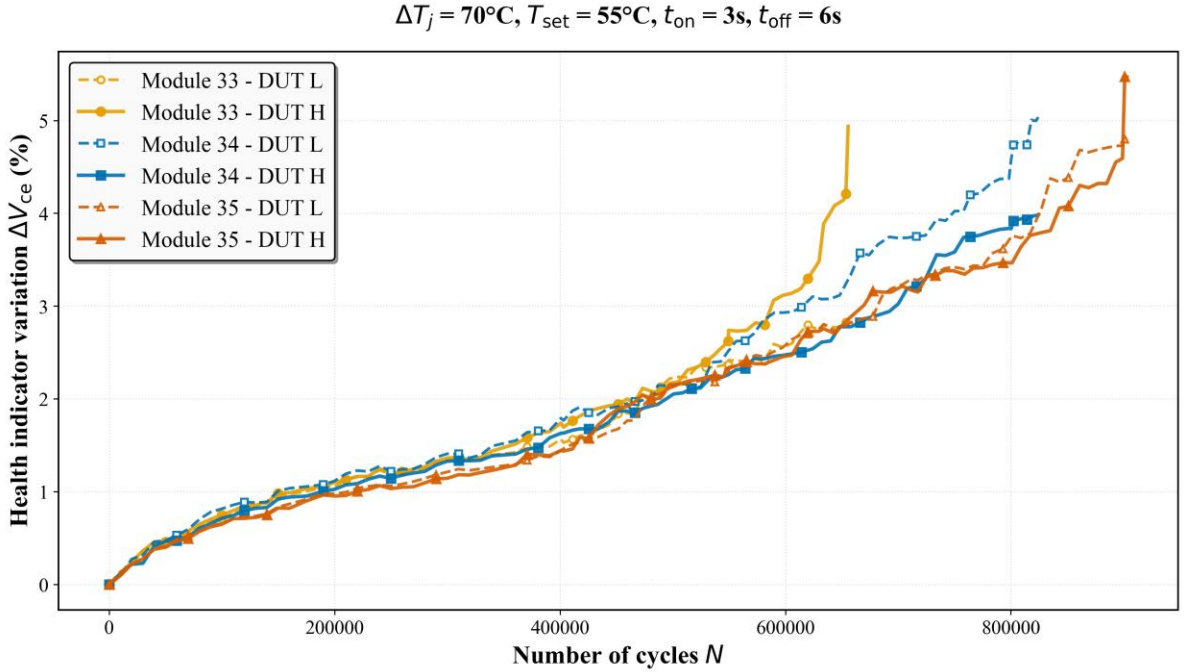
Evolution of ΔV_{ce} as a health indicator during cycling tests (Dornic 2019)

Global context Power modules Research question Methodology

- Cyclic loading: Heating phase (t_{on}) increasing the base temperature by ΔT followed by a cooling phase (t_{off}) returning to the base temperature T_{set}
- Health monitored through observation of the collector-emitter voltage V_{ce}



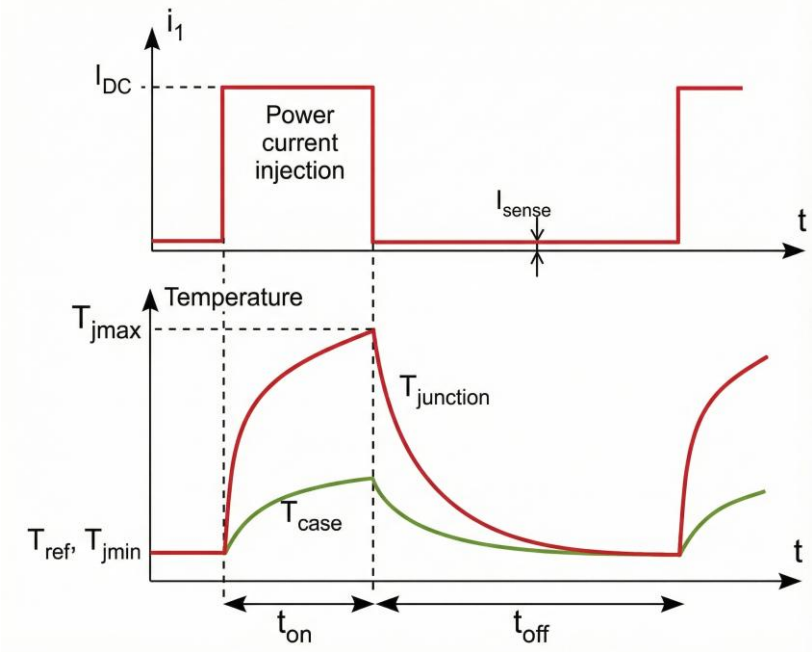
Profile of a single loading cycle (Dornic 2019)



Evolution of ΔV_{ce} as a health indicator during cycling tests (Dornic 2019)

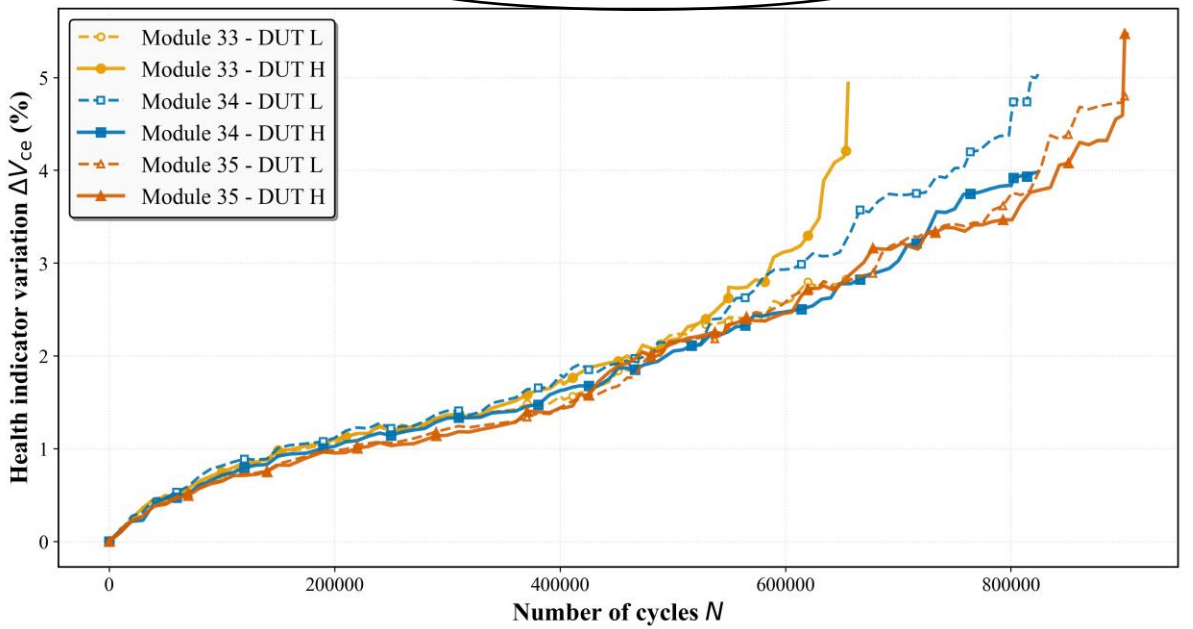
Global context Power modules Research question Methodology

- Cyclic loading: Heating phase (t_{on}) increasing the base temperature by ΔT followed by a cooling phase (t_{off}) returning to the base temperature T_{set}
- Health monitored through observation of the collector-emitter voltage V_{ce}



Profile of a single loading cycle (Dornic 2019)

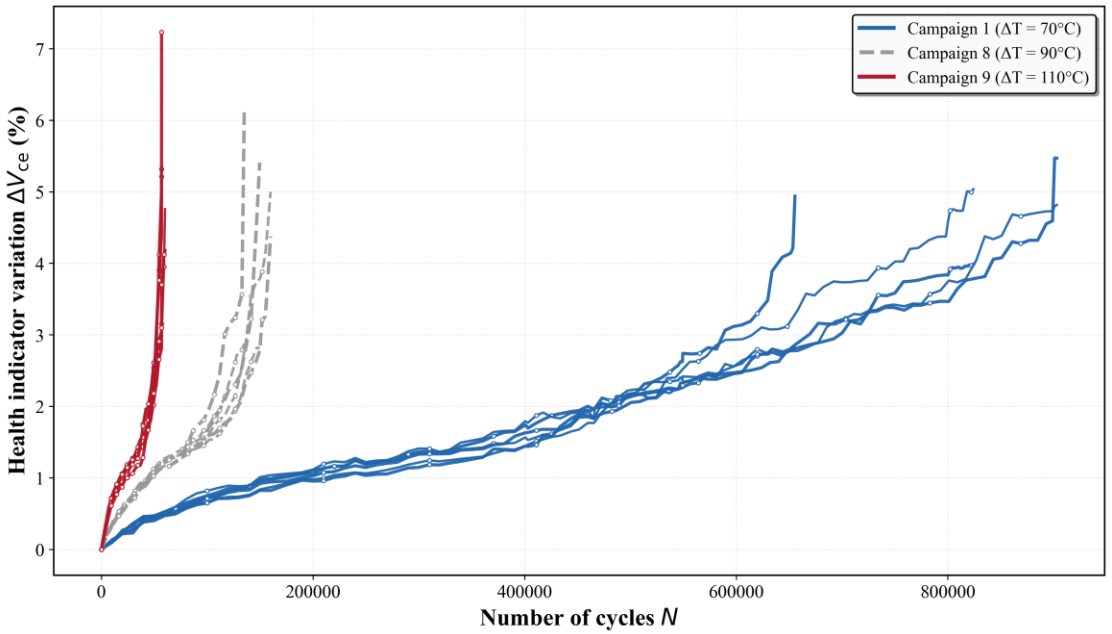
$\Delta T_j = 70°C, T_{set} = 55°C, t_{on} = 3s, t_{off} = 6s$ Loading conditions



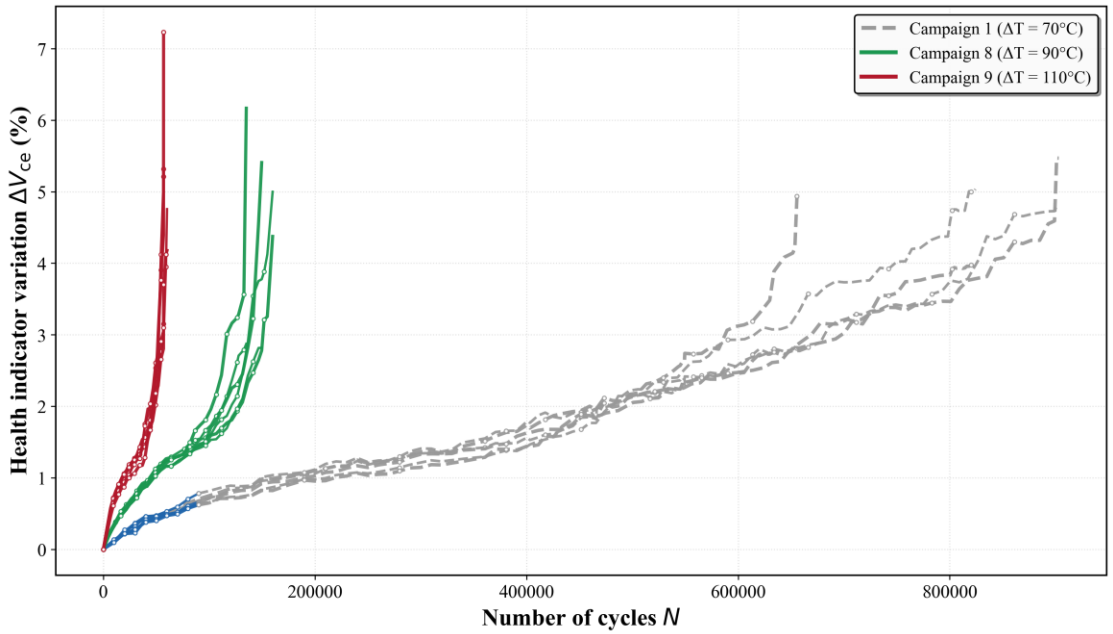
Evolution of ΔV_{ce} as a health indicator during cycling tests (Dornic 2019)

Global context Power modules Research question Methodology

Developing a fast and accurate prognostic framework to estimate the health state of power modules under load profiles that were not represented in the experimental data



Predicting the health state evolution based on higher temperature $(\Delta T = 110°C)$ *and lower temperature* $(\Delta T = 70°C)$ *experimental data*

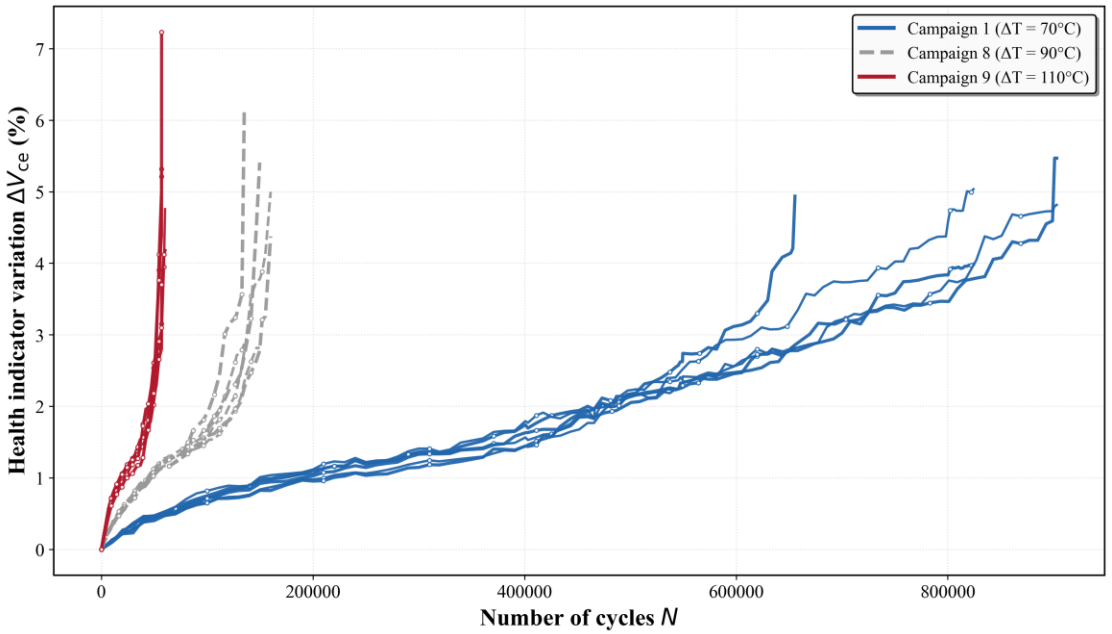


Predicting the health state evolution based on higher temperature $(\Delta T = 110°C$ *and* $\Delta T = 90°C)$ *experimental data*

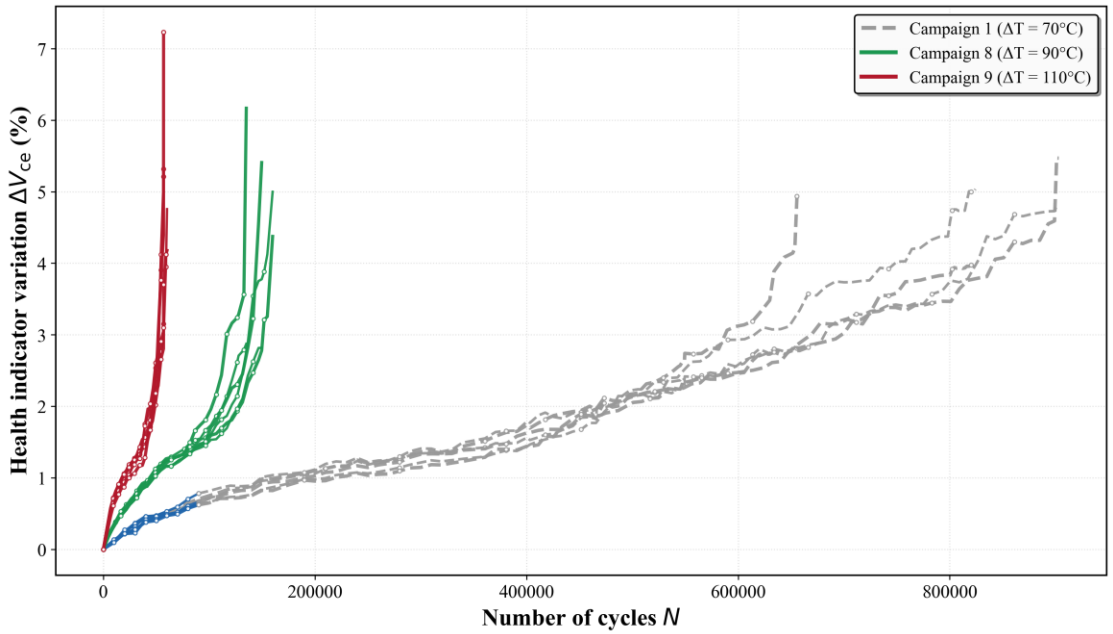
Global context Power modules Research question Methodology

Developing a fast and accurate prognostic framework to estimate the health state of power modules under load profiles that were not represented in the experimental data

Addresses the problem of computational speed and out-of-distribution predictions



Predicting the health state evolution based on higher temperature $(\Delta T = 110°C)$ *and lower temperature* $(\Delta T = 70°C)$ *experimental data*

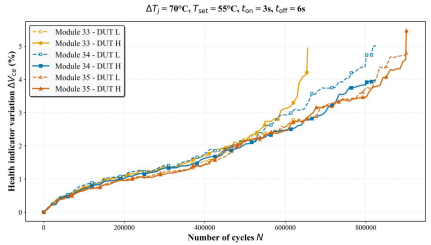


Predicting the health state evolution based on higher temperature $(\Delta T = 110°C$ *and* $\Delta T = 90°C)$ *experimental data*

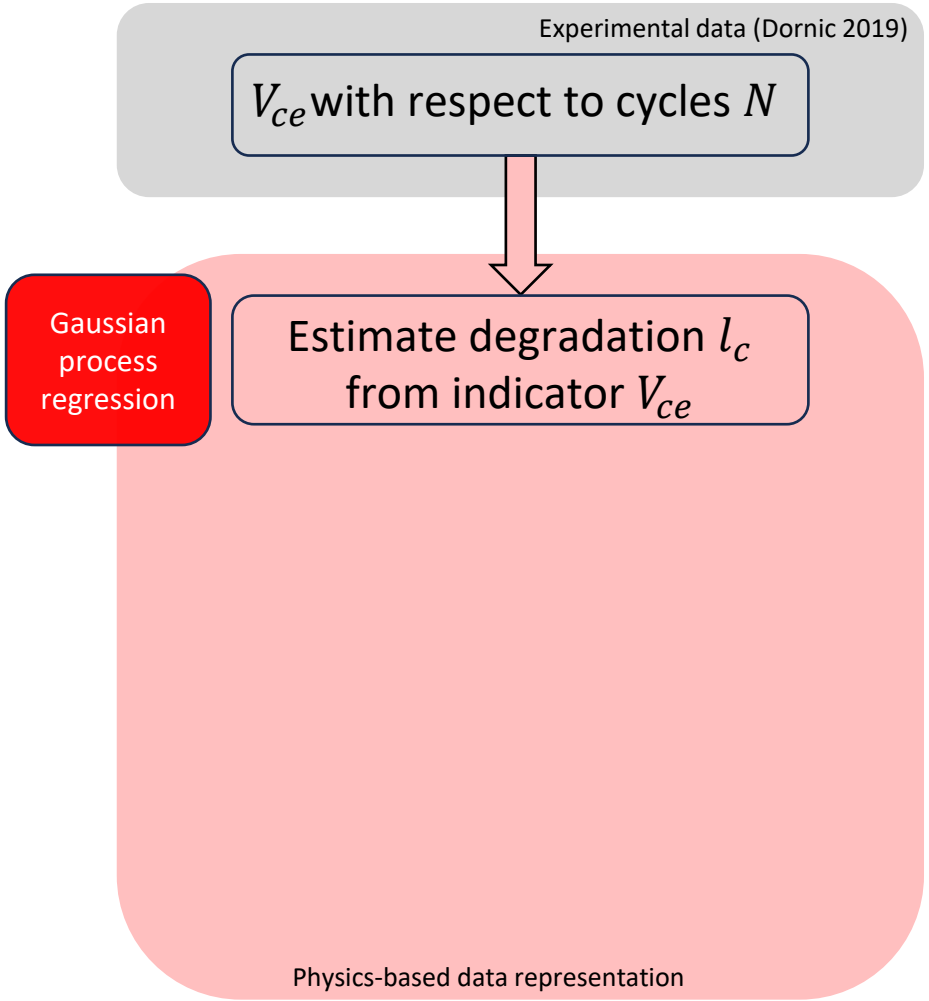
Global context Power modules Research question Methodology

Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

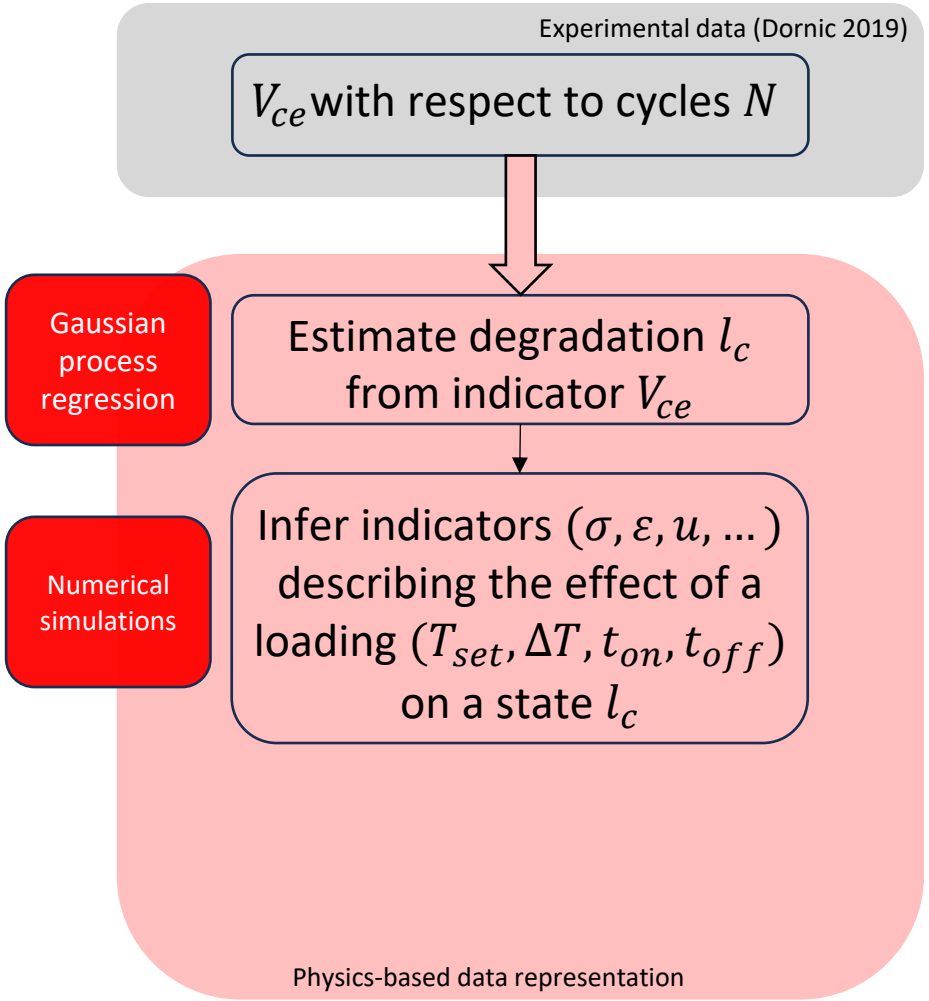


Global context Power modules Research question Methodology



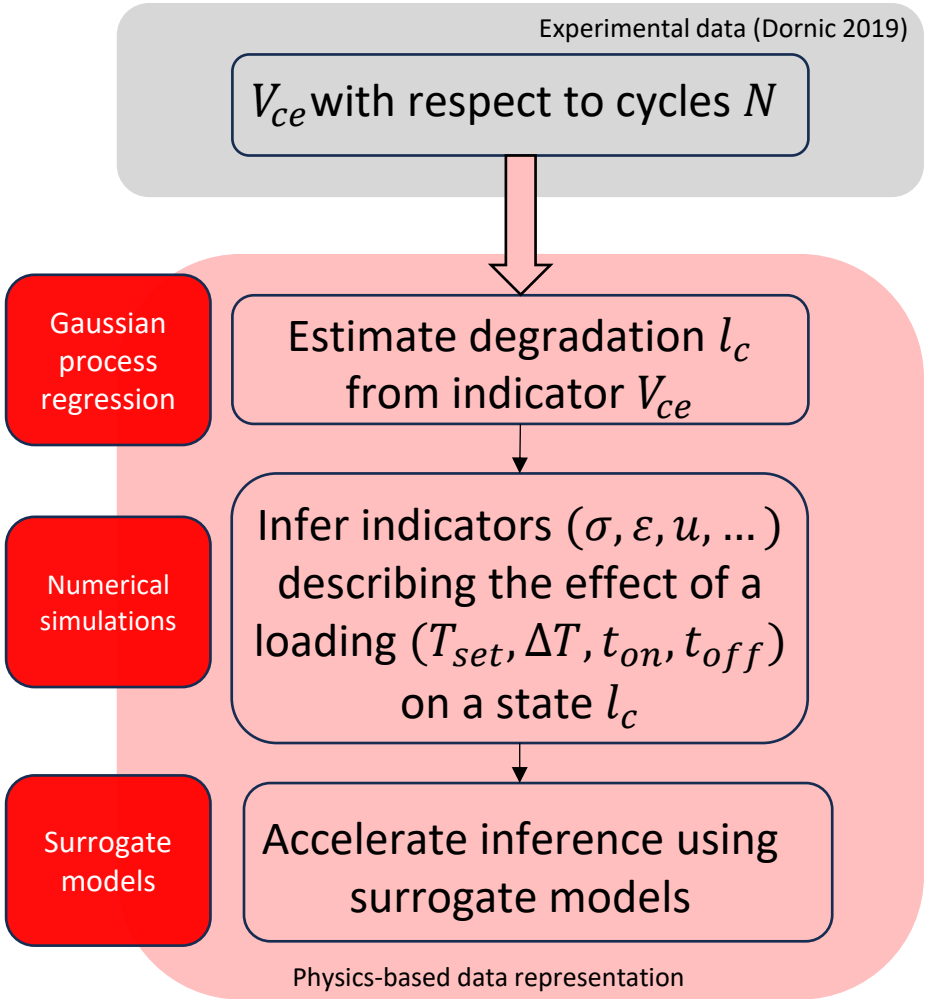
Analytical (Tims 1999)
Simulation-based (Nazar 2023)

Global context Power modules Research question Methodology



Analytical (Halouani 2023)

Global context Power modules Research question Methodology



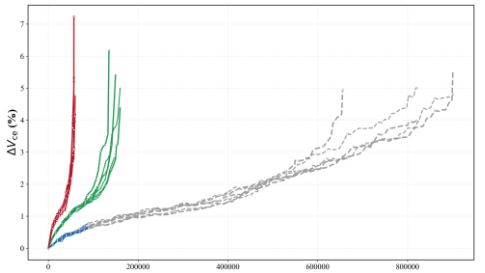
Reduced order models (Schuler 2023)

Global context

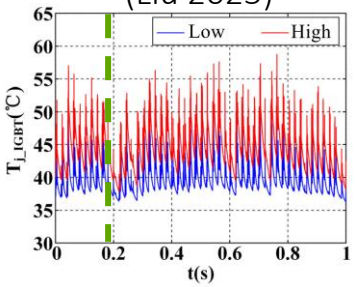
Power modules

Research question

Methodology



(Liu 2025)



Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

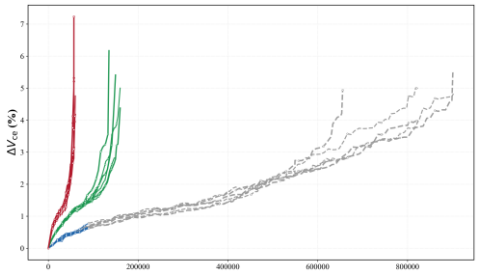
Observation V_{ce}

Global context

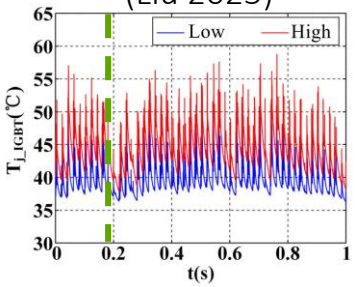
Power modules

Research question

Methodology



(Liu 2025)



Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Degradation l_c

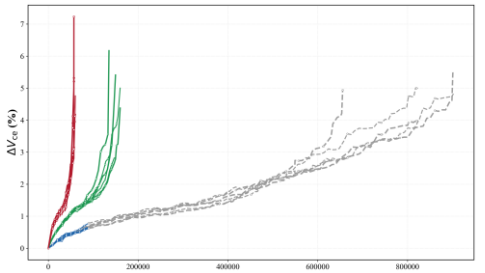
Observation V_{ce}

Global context

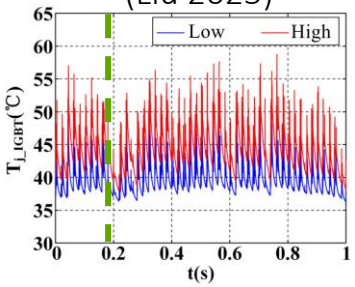
Power modules

Research question

Methodology



(Liu 2025)



Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Degradation l_c

Surrogate models

Observation V_{ce}

Indicators $(\sigma, \varepsilon, u, \dots)$

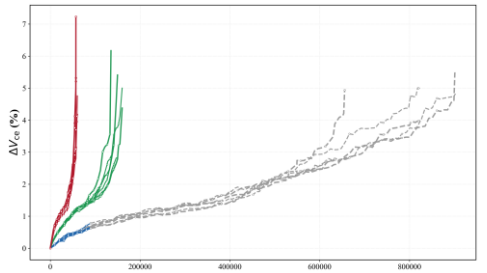
Global context

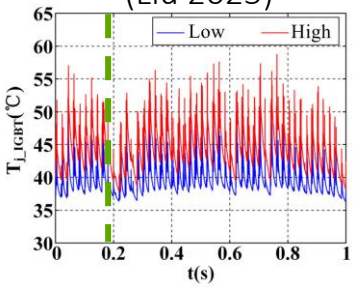
Power modules

Research question

Methodology

(Liu 2025)





Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Surrogate models

Degradation l_c

Observation V_{ce}

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Probabilistic data-driven model

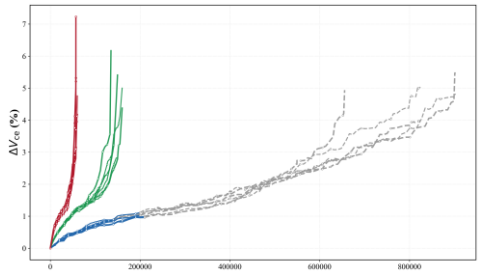
Global context

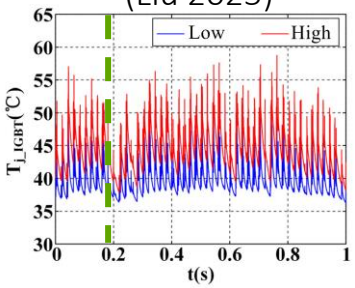
Power modules

Research question

Methodology

(Liu 2025)





Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Numerical simulations

Surrogate models

Estimate degradation l_c from indicator V_{ce}

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Surrogate models

Degradation l_c

Observation $V_{ce} + \Delta V_{ce}$

Autoregressive lifetime estimation

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Gaussian process regression

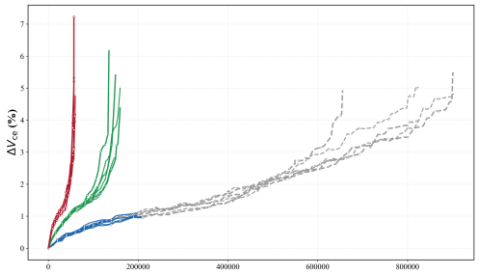
Probabilistic data-driven model

Global context

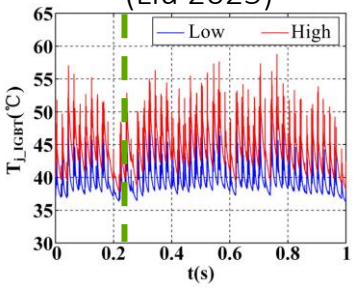
Power modules

Research question

Methodology



(Liu 2025)



Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Surrogate models

Degradation l_c

Observation V_{ce}

Autoregressive lifetime estimation

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Gaussian process regression

Probabilistic data-driven model

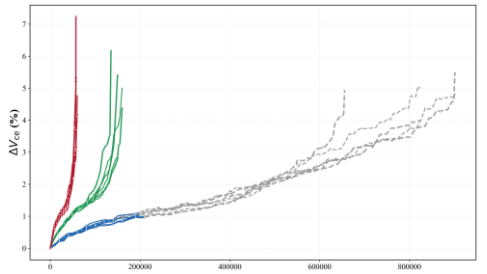
Global context

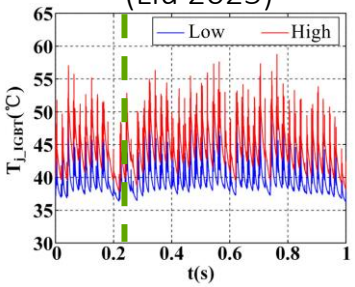
Power modules

Research question

Methodology

(Liu 2025)





Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Degradation l_c

Surrogate models

Observation V_{ce}

Autoregressive lifetime estimation

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Gaussian process regression

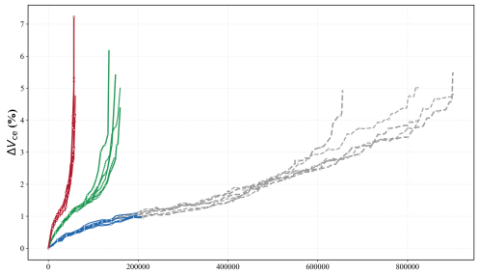
Probabilistic data-driven model

Global context

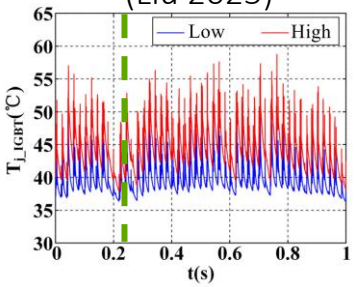
Power modules

Research question

Methodology



(Liu 2025)



Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Degradation l_c

Surrogate models

Observation V_{ce}

Autoregressive lifetime estimation

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Gaussian process regression

Probabilistic data-driven model

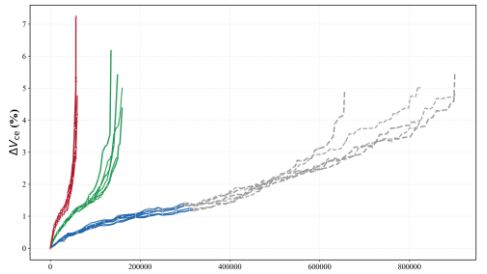
Global context

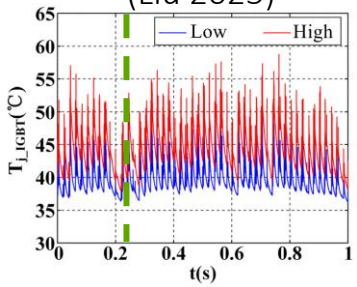
Power modules

Research question

Methodology

(Liu 2025)





Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

Surrogate models

Accelerate inference using surrogate models

Physics-based data representation

Data-driven lifetime estimation

Gaussian process regression

Degradation l_c

Surrogate models

Observation $V_{ce} + \Delta V_{ce}$

Autoregressive lifetime estimation

Indicators $(\sigma, \varepsilon, u, \dots)$

Damage increment Δl_c

Gaussian process regression

Probabilistic data-driven model

Global context | Power modules | Research question | Methodology

Experimental data (Dornic 2019)

V_{ce} with respect to cycles N

Gaussian process regression

Estimate degradation l_c from indicator V_{ce}

Numerical simulations

Infer indicators $(\sigma, \varepsilon, u, \dots)$ describing the effect of a loading $(T_{set}, \Delta T, t_{on}, t_{off})$ on a state l_c

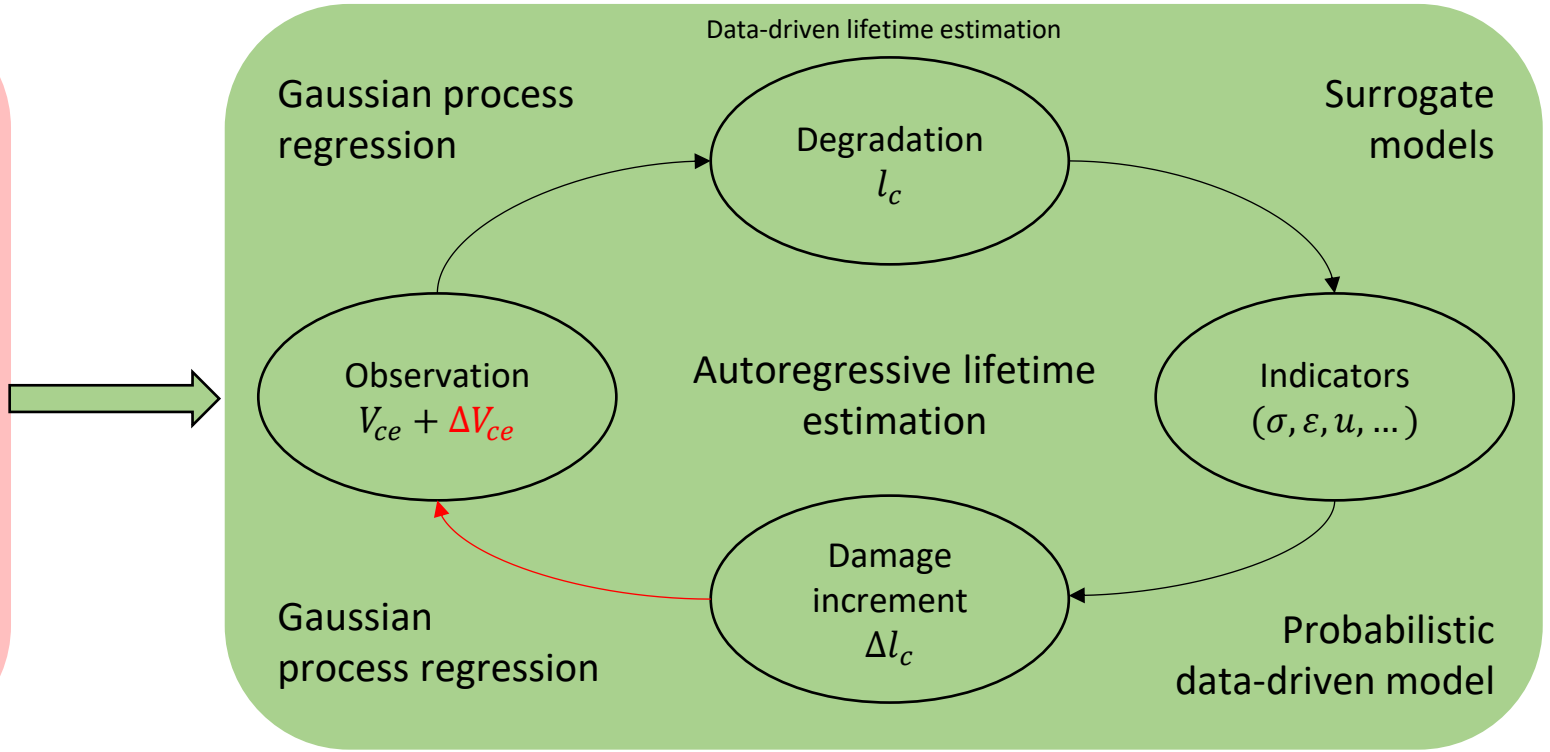
Surrogate models

Accelerate inference using surrogate models

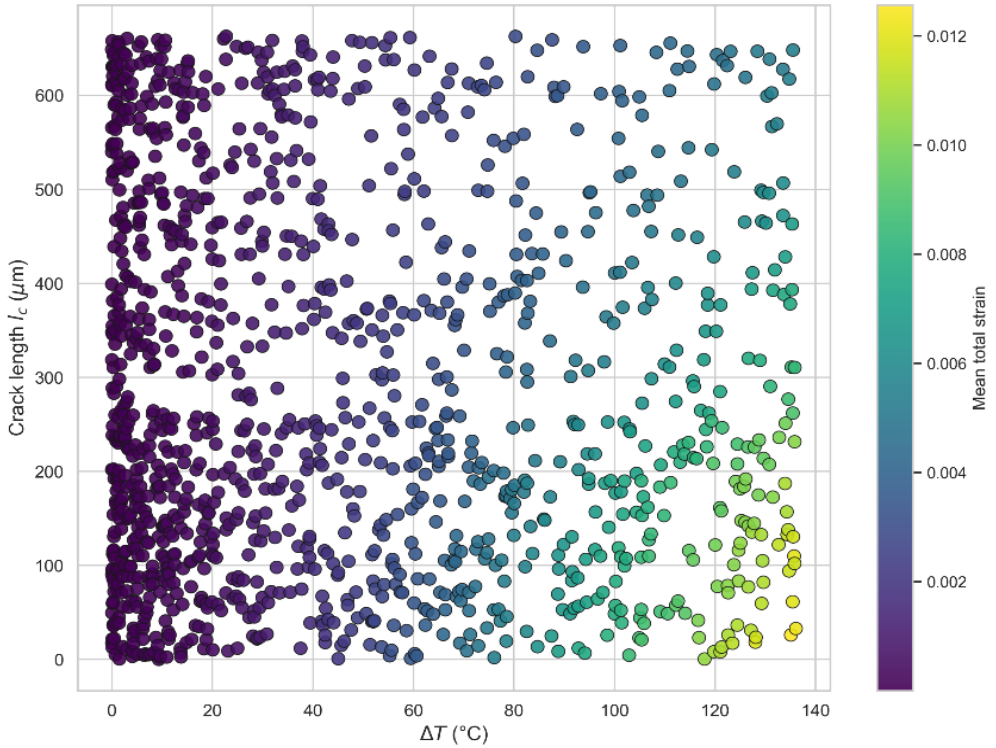
Physics-based data representation

Combining physical knowledge and machine learning models for lifetime estimation of power modules

- Overstress failure (Chen 2026)
- Substantial research in different domains (Raissi 2019)

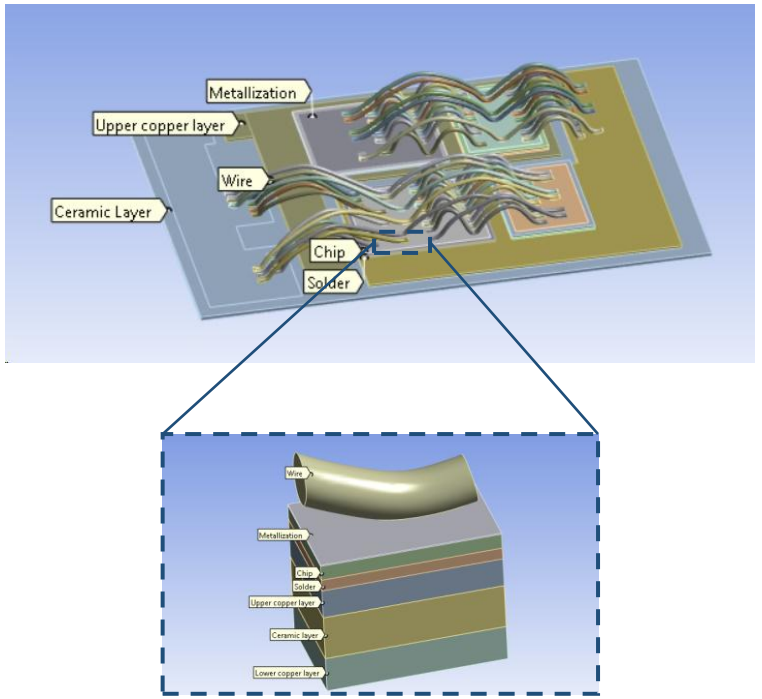


- Lifetime estimation depends on modeling the relationships between
  - State l_c
  - Load $\mathrm{C} = \{\Delta T, t_{on}, t_{off}, T_{set}\}$
  - Response $M = \{\sigma, \varepsilon, u, \dots\}$



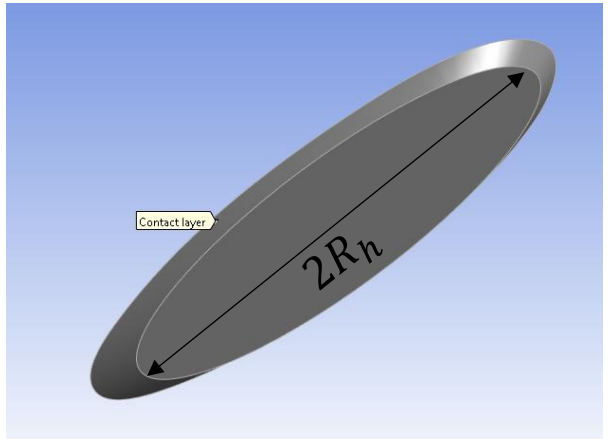
- Lifetime estimation depends on modeling the relationships between
  - State l_c
  - Load $\mathrm{C} = \{\Delta T, t_{on}, t_{off}, T_{set}\}$
  - Response $M = \{\sigma, \varepsilon, u, \dots\}$
- Established via numerical simulations
- M and l_c are mutually dependent:
  - Requires geometric adaptation

Full geometric model of the power module (Schuler 2023)

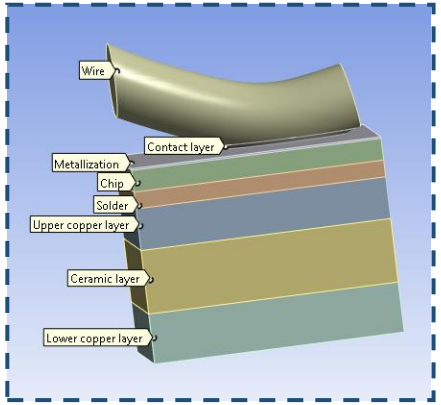


Simplified geometric model

- Contact layer:
  - Concentric bases with equal eccentricity
  - Fixed upper face
  - Lower surface adjusted according to state l_c
  - $2R_h + l_c =$ Initial contact length
- Quasi-planar crack propagation
- Symmetric with respect to the contact's center



Frustum-shaped contact layer



Integrating degradation effect

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

- Given arbitrary state l_c and loading C, calculate $M = \{\sigma, \varepsilon, u, \dots\}$
  - Nodal + Elemental + Global quantities
- Strong thermo-electrical + Weak thermo-mechanical coupling

1. Adjust the contact layer according to l_c
2. Set initial and boundary conditions (T_{set}, ground reference, fixed support, cooler…)
3. Apply loading current I for a duration t_{on} such that $T_{max} - T_{set} = \Delta T$
4. Stop the current during the cooling phase t_{off}
5. Retrieve relevant mechanical indicators

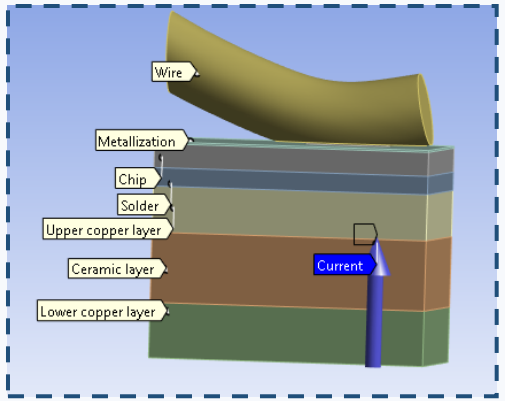
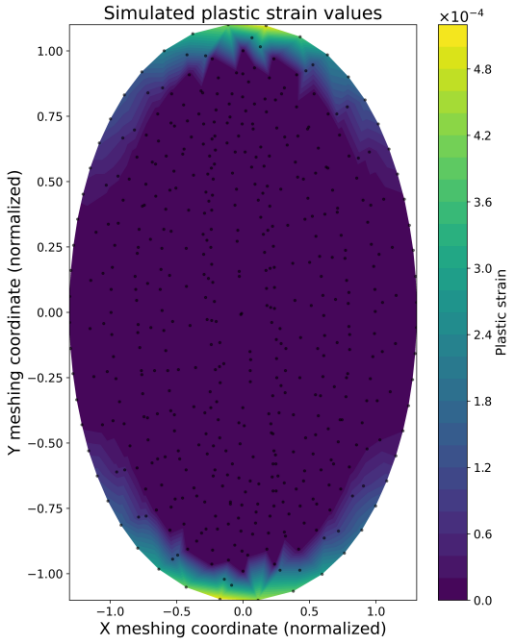


Illustration of loading through current injection

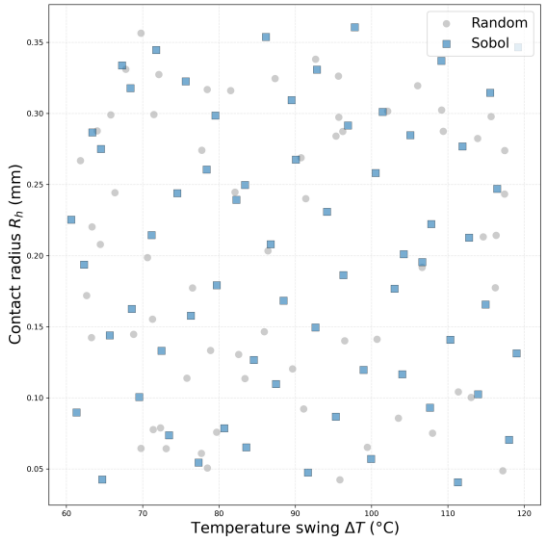


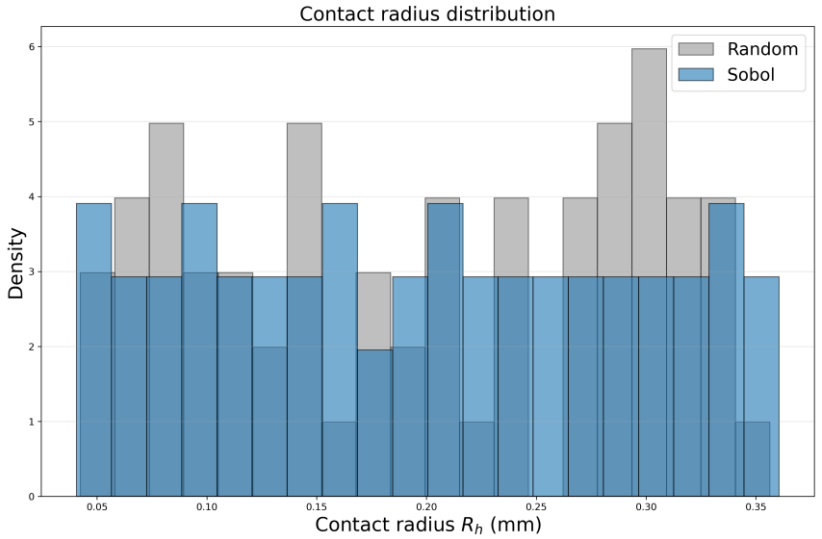
- Numerical simulations are too costly to determine the outcome of each possible scenario
- A deliberate choice of simulation is needed given this constraint

- Numerical simulations are too costly to determine the outcome of each possible scenario
- A deliberate choice of simulation is needed given this constraint

- Objective: Optimal coverage of the data space given a simulation budget
- ⇒ Sobol sampling
  - Quasi Monte-Carlo scheme
  - Provides uniform space-filling to minimize gaps in the input space

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model





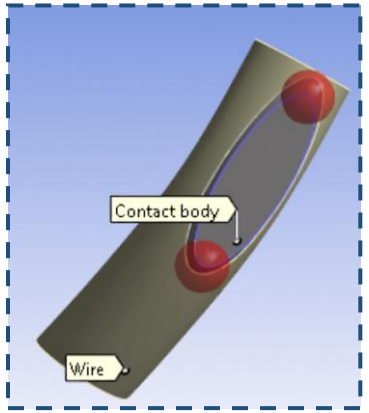
Input space coverage using Sobol sequence vs. random uniform sampling

- Objective: Optimal coverage of the data space given a simulation budget
- ⇒ Sobol sampling
  - Quasi Monte-Carlo scheme
  - Provides uniform space-filling to minimize gaps in the input space

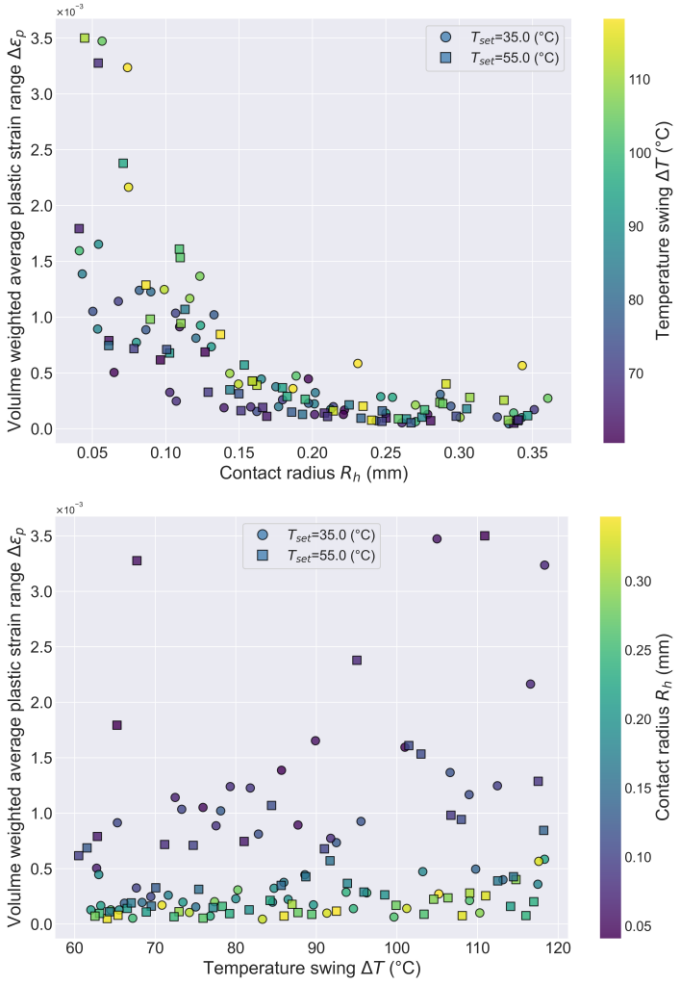
- 1024 total simulations
- 15-30 minutes/simulation

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

- Volume-weighted average (VWA) plastic strain range measured near the crack tips as an indicator of degradation
  - Averaging radius calculated according to the Grid Convergence Index (GCI)
- High impact of crack length on plastic strain
  - ⇒Importance of degradation modeling
- Results consistent with the literature
  - Plastic strain increases with ΔT and l_c



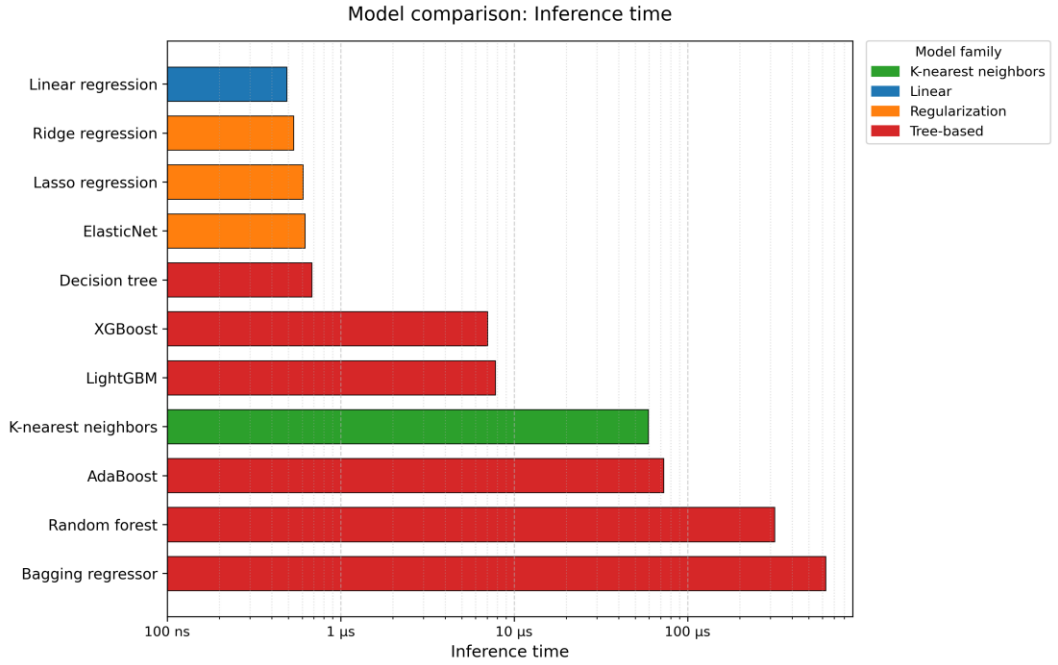
Averaging sphere for VWA calculation

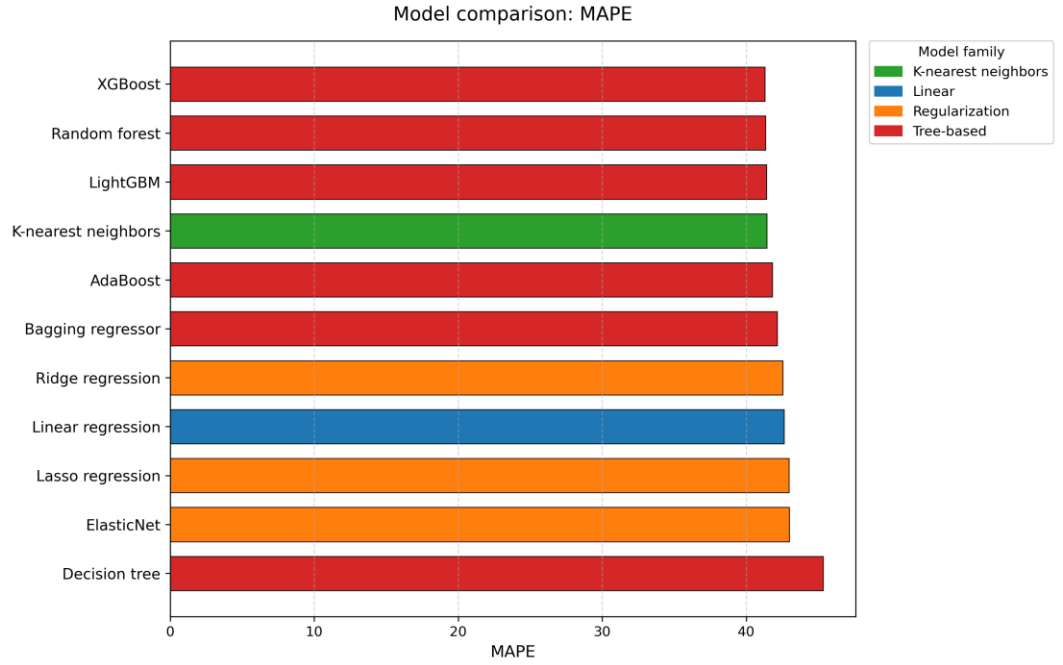


VWA plastic strain range w.r.t simulation parameters (First 128 samples of the Sobol sequence)

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

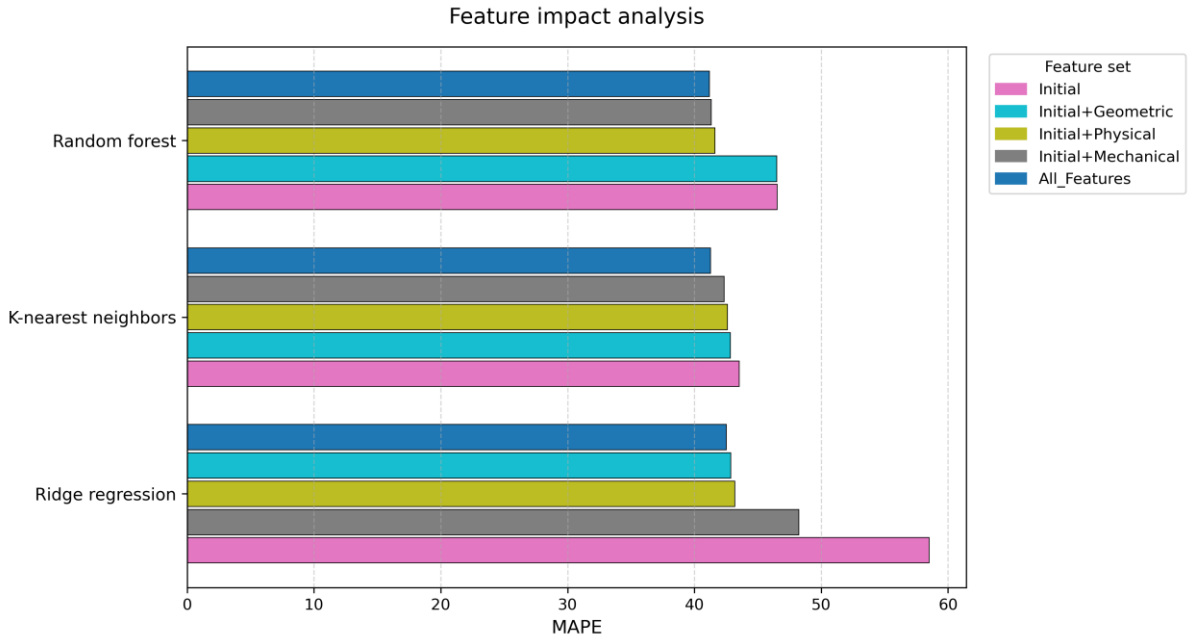
- Machine learning models benchmarked on the simulated dataset to generalize over the input space: illustration on WVA plastic strain prediction
- Substantial speed-up ($\approx 10^6$) at the cost of accuracy ($\approx 40\%$ MAPE)
- Marginal differences between models due to appropriate feature engineering





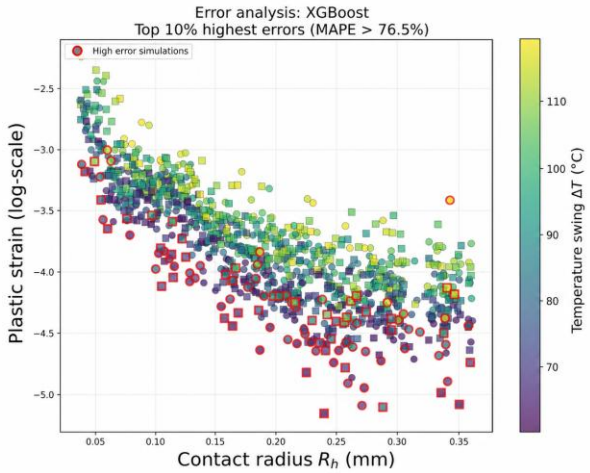
Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

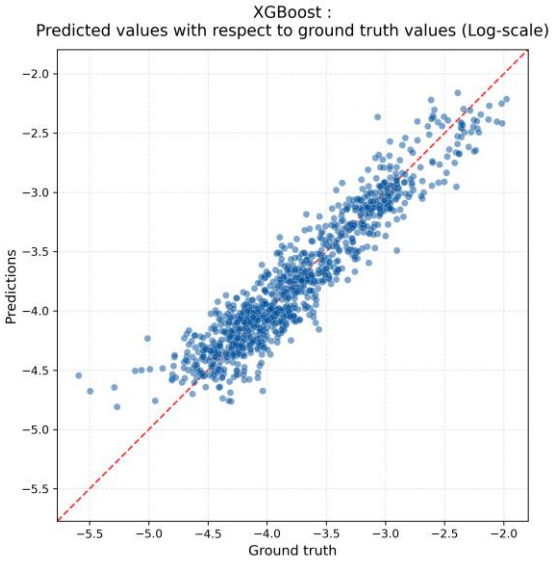
- Machine learning models benchmarked on the simulated dataset to generalize over the input space: illustration on WVA plastic strain prediction
- Substantial speed-up ($\approx 10^6$) at the cost of accuracy ($\approx 40\%$ MAPE)
- Marginal differences between models due to appropriate feature engineering



Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

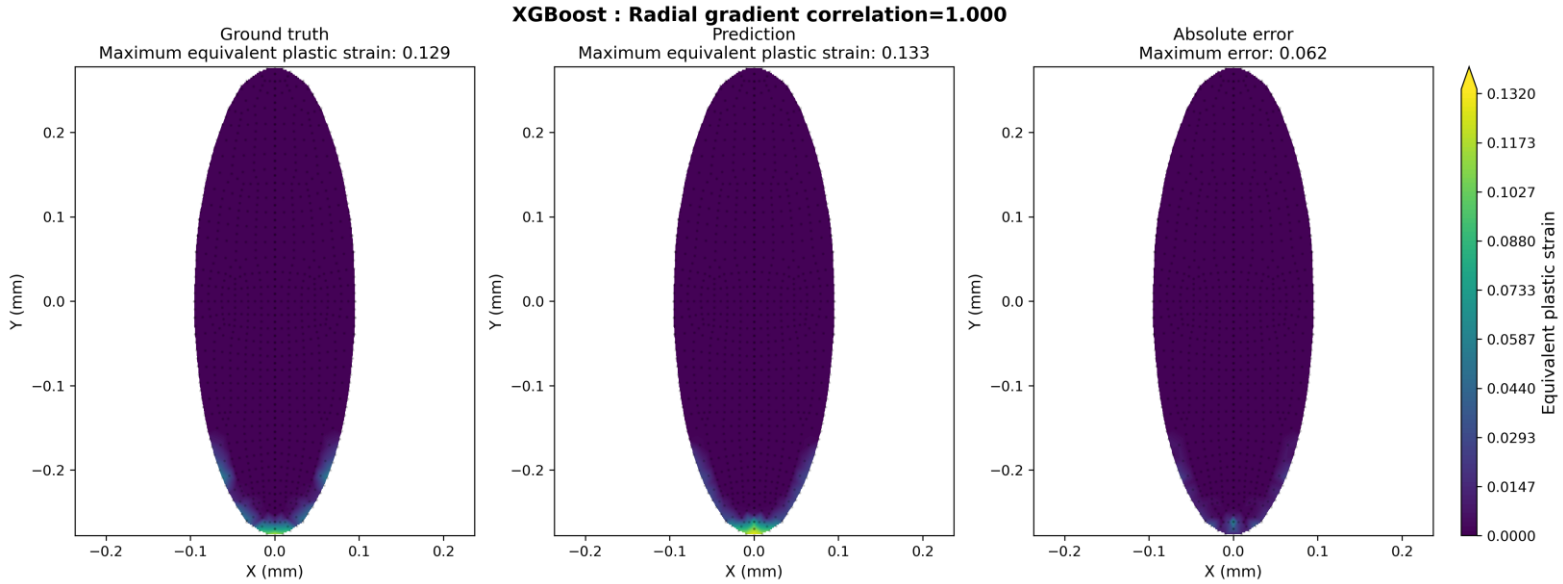
- High MAPE values are mostly attributed
  - Boundary values: chosen beyond the training and test conditions $\Delta T \in [70°C, 110°C]$ in our dataset
  - Outliers caused by diverging simulations $R_h \in [0.26, 0.32] mm$ due meshing inadequacy
  - Low and variable data scale $(10^{-5} – 10^{-3})$
  - Smoothing due to VWA
- Error effect mitigated in kernel-based methods, provided low residual variance





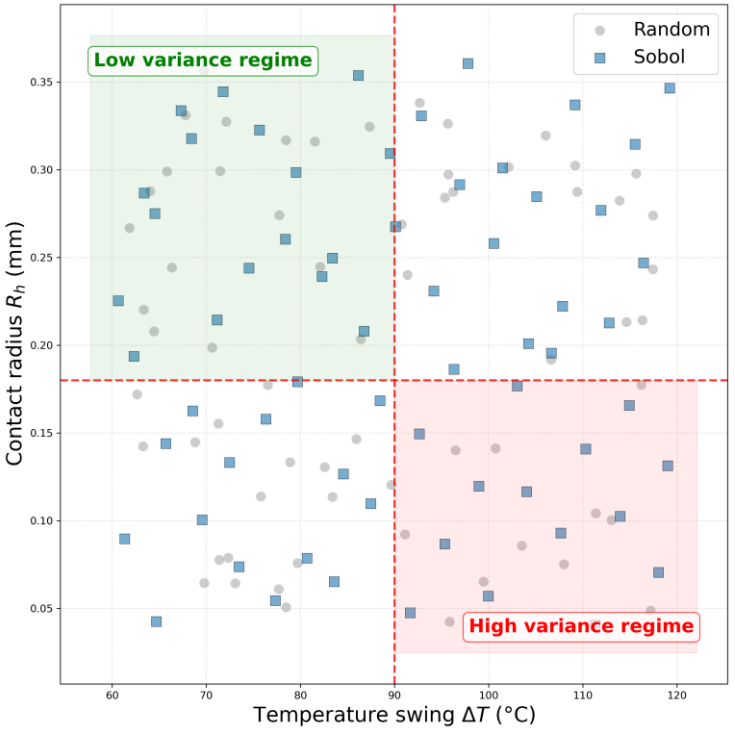
Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

- Spatial distribution predictions showcase surrogate models' capability of reproducing the physical phenomenon: peak plastic strain near the contact tip
- Asymmetric distribution due to the wire's asymmetric geometry



Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

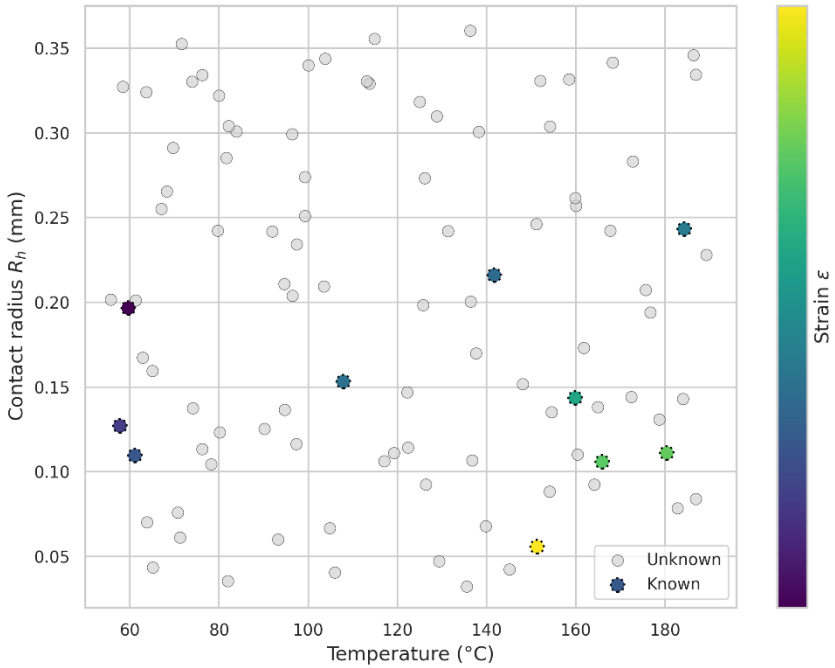
- Sobol sampling ensures uniform input space coverage without prior knowledge of the output
- Treats all regions of the parameter space with equal weight
- Suboptimal when the underlying input-output relationship is non-uniform
- Improvement potential through sequential knowledge integration



Equal weight distribution using the Sobol sequence despite semantic differences

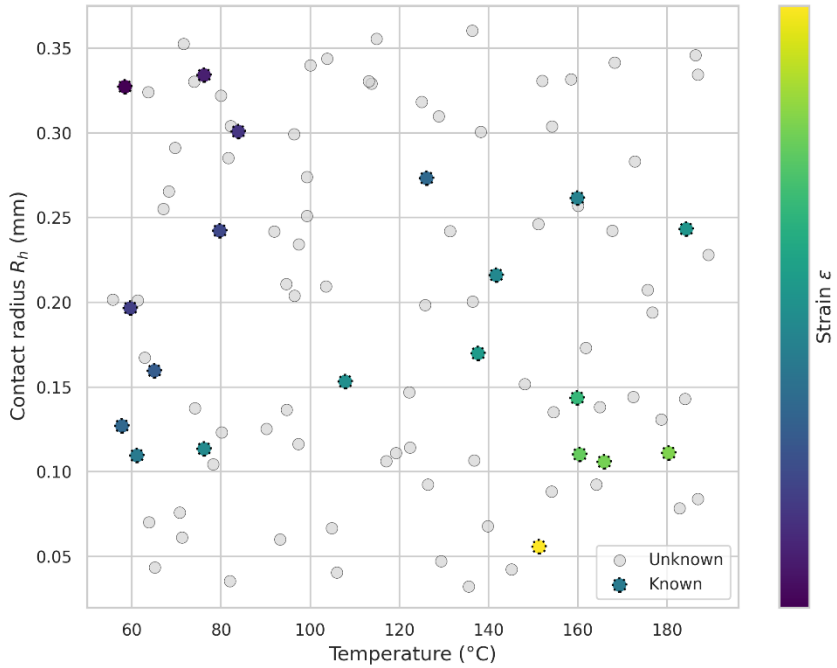
Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

1. Choose an ensemble of highly uncorrelated surrogate models (exp: RF, KNN, Ridge)
2. Select simulations to provide initial insight (burn-in)
3. Identify all candidate simulations to be conducted
4. Fit the ensemble on the burn-in set
5. Forward pass of all candidate scenarios (cheap)
6. Measure the variance of the ensemble's prediction $\sigma^2_{ensemble}$ for each candidate simulation



Sampling using active learning (illustrative example)

7. Prioritize candidate simulations with high variance to be added to the training data
8. Conduct simulations on the top candidates by variance to be added to the training set
9. Fit the ensemble on the extended training set

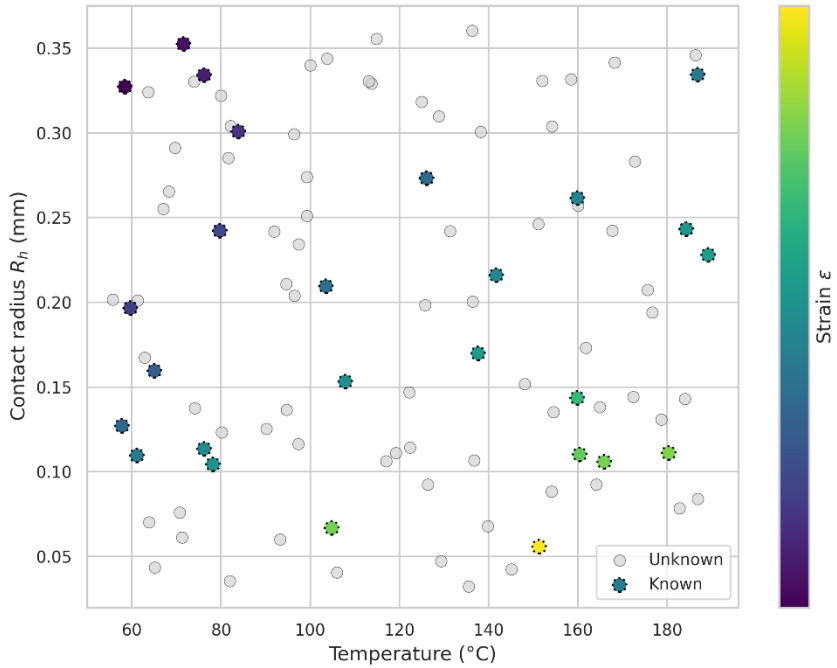


Sampling using active learning (illustrative example)

High $\sigma^2_{ensemble}$ ⇒ Disagreement within the committee ⇒ Highly contentious (informative) simulation

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

7. Prioritize candidate simulations with high variance to be added to the training data
8. Conduct simulations on the top candidates by variance to be added to the training set
9. Fit the ensemble on the extended training set
10. Repeat the acquisition step until threshold or budget limit



Sampling using active learning (illustrative example)

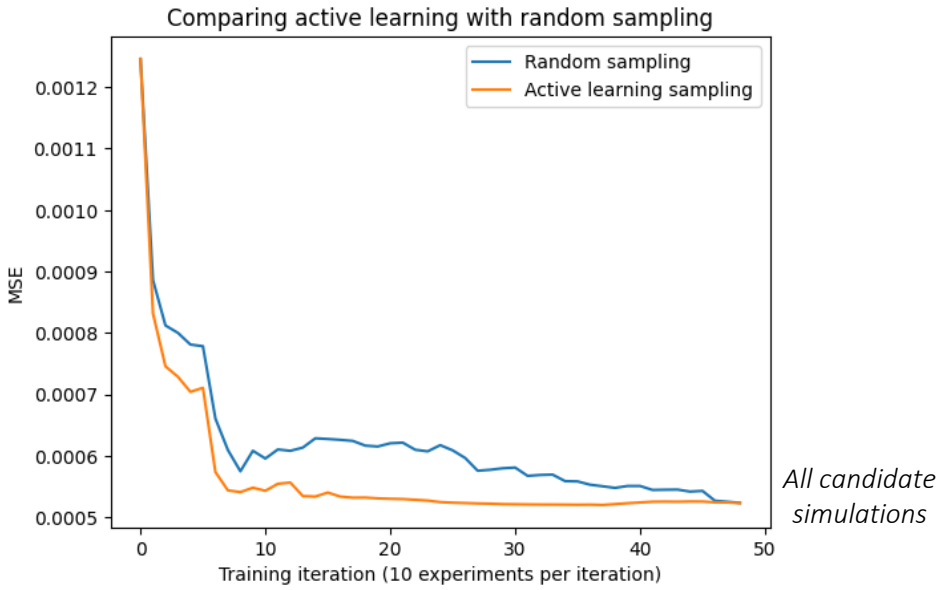
High $\sigma^2_{ensemble}$ ⇒ Disagreement within the committee ⇒ Highly contentious (informative) simulation

Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

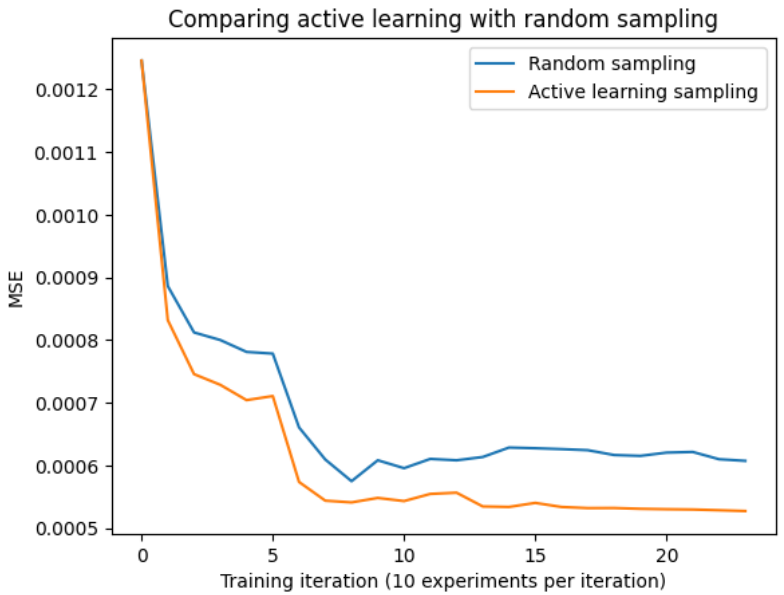
Active learning vs. random uniform sampling:

Active learning converges faster when exhausting all simulations, and to a lower error when a simulation budget is imposed

- 50 burn-in simulations
- 500 total candidates
- 10 simulations per acquisition step

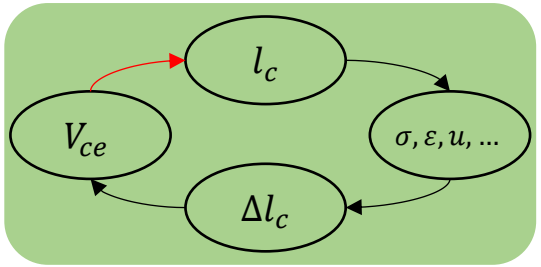


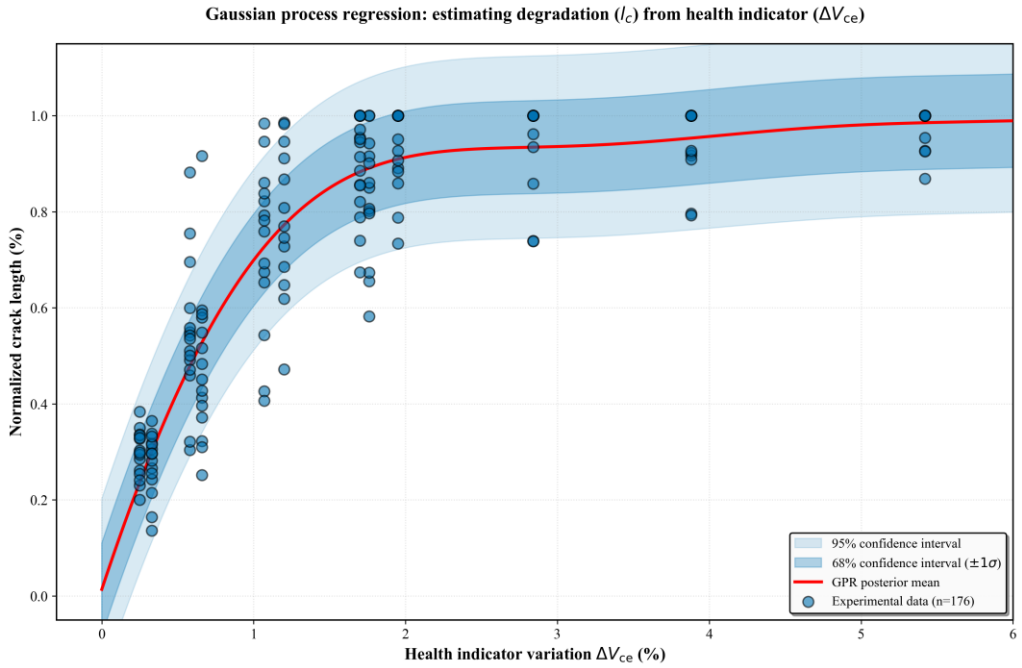
All candidate simulations



50% *candidate simulations*

- Destructive data (Dornic 2019) is used to fit a gaussian process regression model
- Assumptions:
  - Degradation l_c can fully be determined from V_{ce}
  - $l_c \sim \mathcal{N}(\mu(V_{ce}), \sigma(V_{ce})^2)$
- Design choice:
  - Probabilistic model due to high data variance and sparsity
  - Non-parametric adaptation





Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

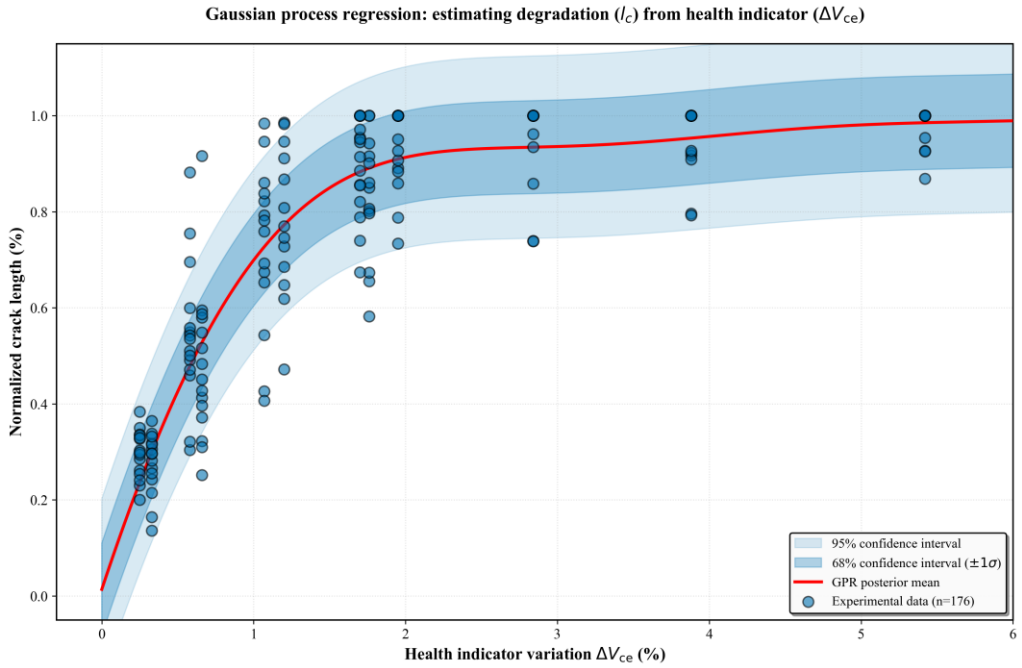
- Kernel $\mathcal{K}(.,.)$: Base kernel + White noise kernel
- Parameters chosen through maximum likelihood
- 12-fold cross validation to avoid leakage

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}(m(x), \begin{bmatrix} \mathcal{K}(x,x) & \mathcal{K}(x,x_*) \\ \mathcal{K}(x_*,x) & \mathcal{K}(x_*,x_*) \end{bmatrix}$$

Distribution of observations (x, y) and predictions (x_*, y_*) according to the gaussian process assumption

Base kernel	MAE	CRPS	Coverage (%)	Log-likelihood
Matérn $\nu = \frac{5}{2}$	**0.0368**	0.0678	88.2	225.246
Matérn $\nu = \frac{3}{2}$	0.0436	0.0702	**88.7**	224.011
RBF	**0.0368**	**0.0669**	88.2	**225.965**

Benchmarking different base kernels. Metrics averaged across all fold variations.



Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

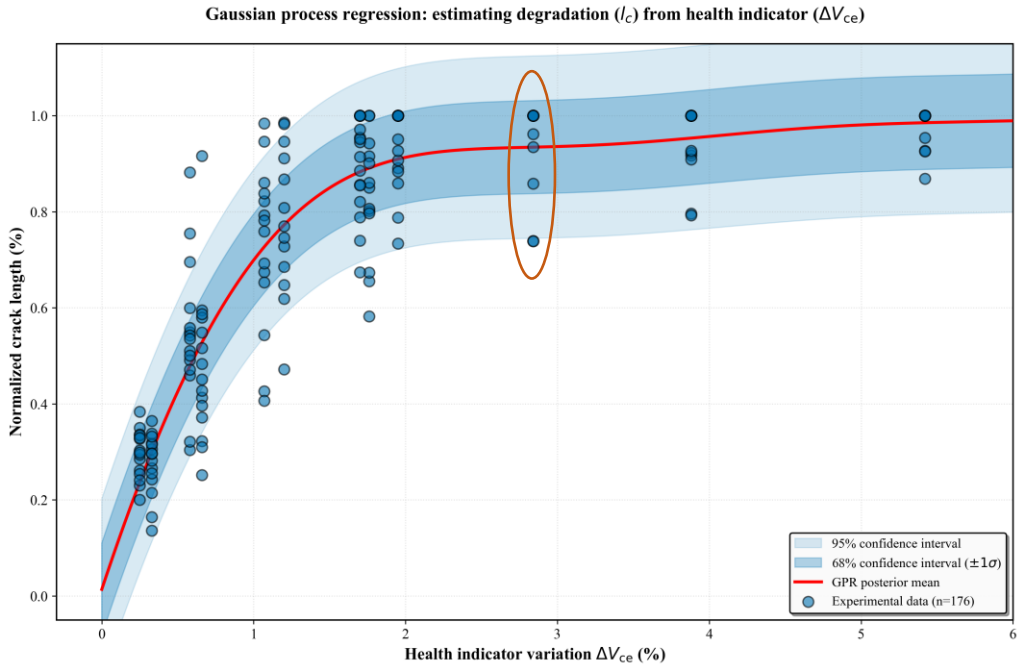
- Kernel $\mathcal{K}(.,.)$: Base kernel + White noise kernel
- Parameters chosen through maximum likelihood
- 12-fold cross validation to avoid leakage
  - 1 fold = 1 cluster

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}(m(x), \begin{bmatrix} \mathcal{K}(x,x) & \mathcal{K}(x,x_*) \\ \mathcal{K}(x_*,x) & \mathcal{K}(x_*,x_*) \end{bmatrix}$$

Distribution of observations (x, y) and predictions (x_*, y_*) according to the gaussian process assumption

Base kernel	MAE	CRPS	Coverage (%)	Log-likelihood
Matérn $\nu = \frac{5}{2}$	**0.0368**	0.0678	88.2	225.246
Matérn $\nu = \frac{3}{2}$	0.0436	0.0702	**88.7**	224.011
RBF	**0.0368**	**0.0669**	88.2	**225.965**

Benchmarking different base kernels. Metrics averaged across all fold variations.



Geometric model | Simulation protocol | Visualization | Surrogate models | Active learning | GPR damage model

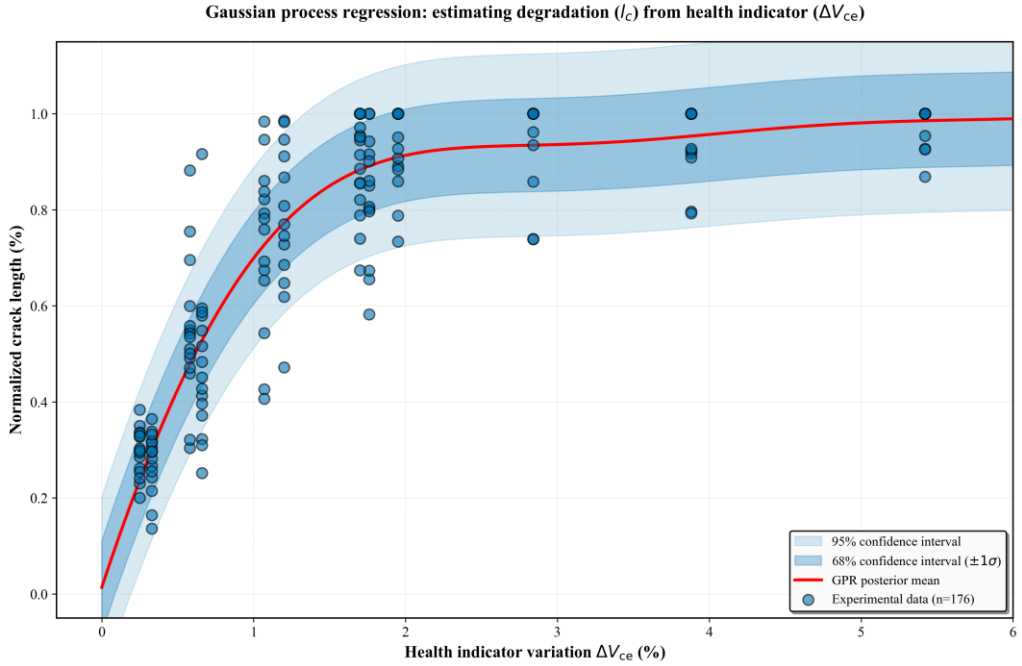
- Kernel $\mathcal{K}(.,.)$: Base kernel + White noise kernel
- Parameters chosen through maximum likelihood
- 12-fold cross validation to avoid leakage

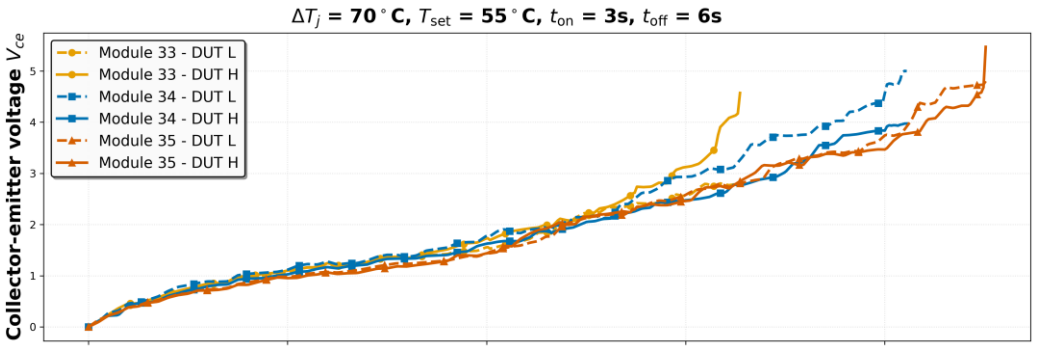
$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}(m(x), \begin{bmatrix} \mathcal{K}(x,x) & \mathcal{K}(x,x_*) \\ \mathcal{K}(x_*,x) & \mathcal{K}(x_*,x_*) \end{bmatrix}$$

Distribution of observations (x, y) and predictions (x_*, y_*) according to the gaussian process assumption

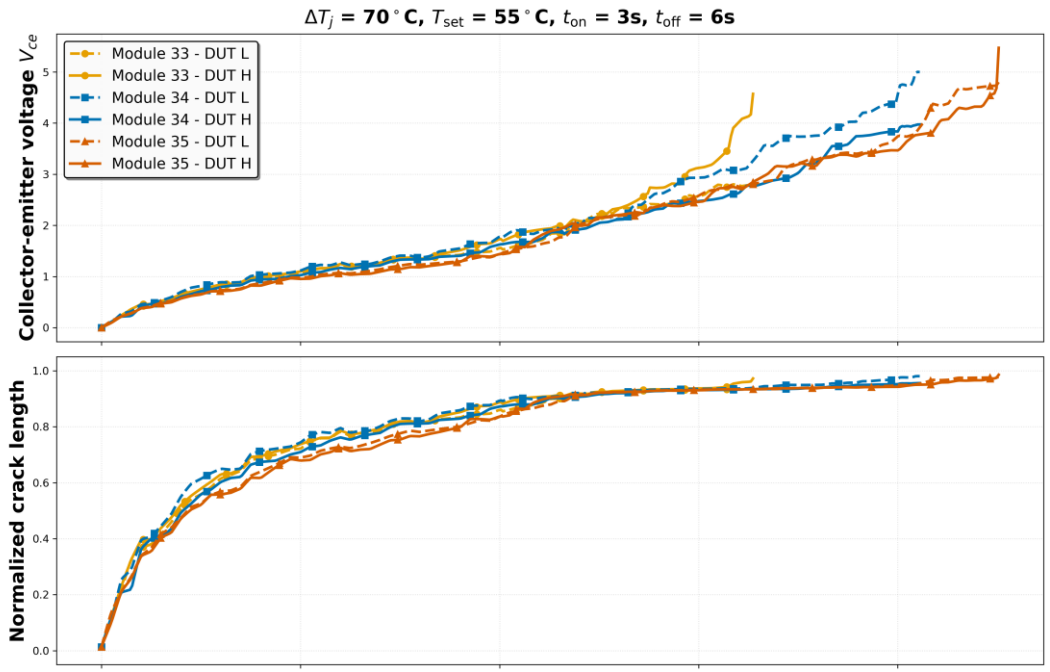
Base kernel	MAE	CRPS	Coverage (%)	Log-likelihood
Matérn $\nu = \frac{5}{2}$	**0.0368**	0.0678	88.2	225.246
Matérn $\nu = \frac{3}{2}$	0.0436	0.0702	**88.7**	224.011
RBF	**0.0368**	**0.0669**	88.2	**225.965**

Benchmarking different base kernels. Metrics averaged across all fold variations.

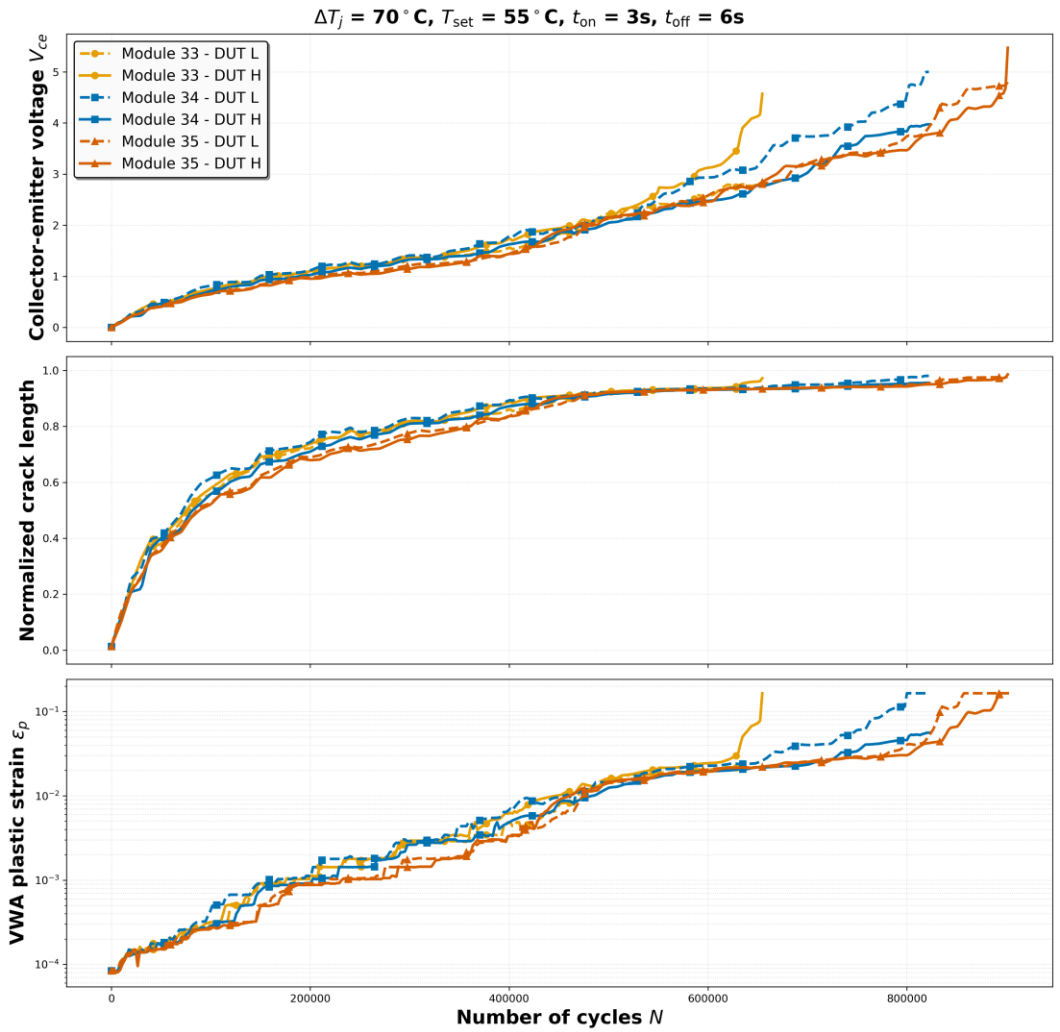



ΔT_j = 70 °C, T_{set} = 55 °C, t_{on} = 3s, t_{off} = 6s
Collector-emitter voltage V_{ce}
Module 33 - DUT L
Module 33 - DUT H
Module 34 - DUT L
Module 34 - DUT H
Module 35 - DUT L
Module 35 - DUT H
0
1
2
3
4
5

Intuition Sampling Evaluation Results BEDTime



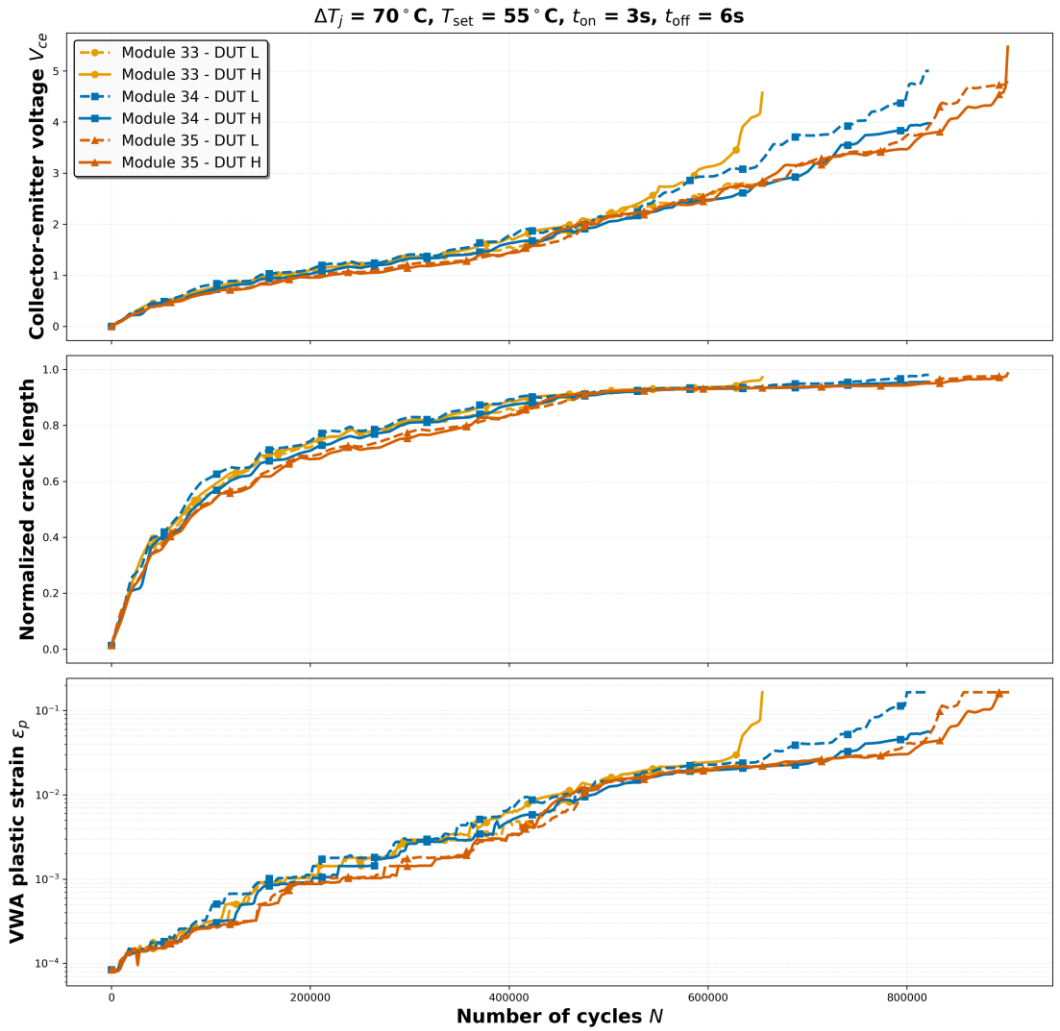
Gaussian process regression



Surrogate model

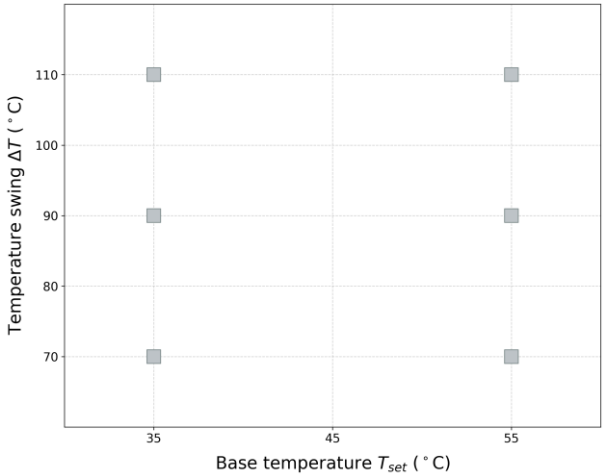
Physics-based data representation producing a degradation-aware multivariate signal

Intuition Sampling Evaluation Results BEDTime



Physics-based data representation producing a degradation-aware multivariate signal

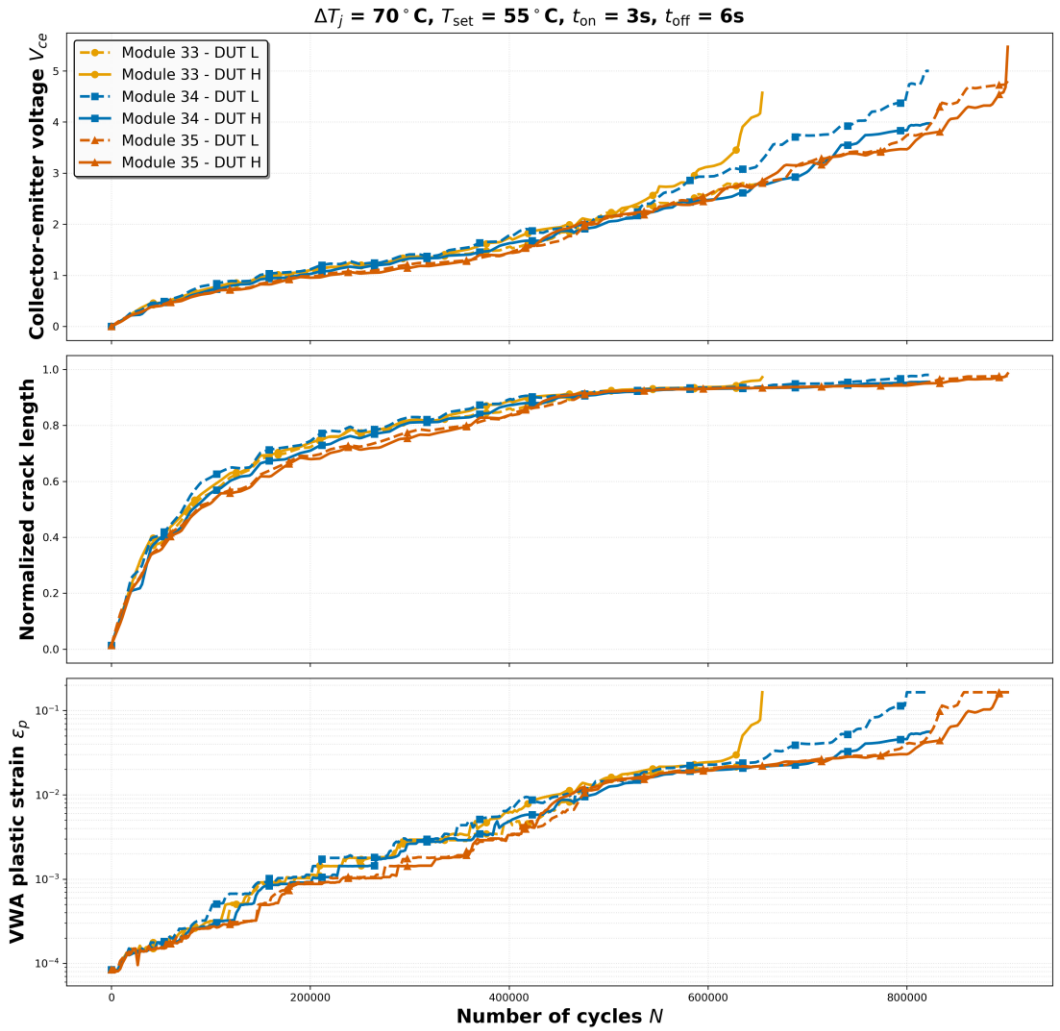
- Sparse representation in the loading conditions' space



Campaign	$T_{set}(°C)$	ΔT(°C)	$t_{on}(s)$	$t_{off}(s)$
1	55	70	3	6
2	35	70	3	6
4	35	90	3	6
6	35	110	3	6
8	55	90	3	6
9	55	110	3	6

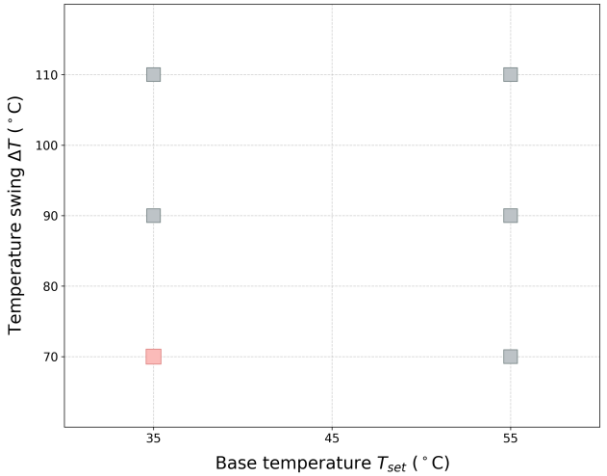
Cycling test campaigns (6 runs per campaign)

Intuition Sampling Evaluation Results BEDTime



Physics-based data representation producing a degradation-aware multivariate signal

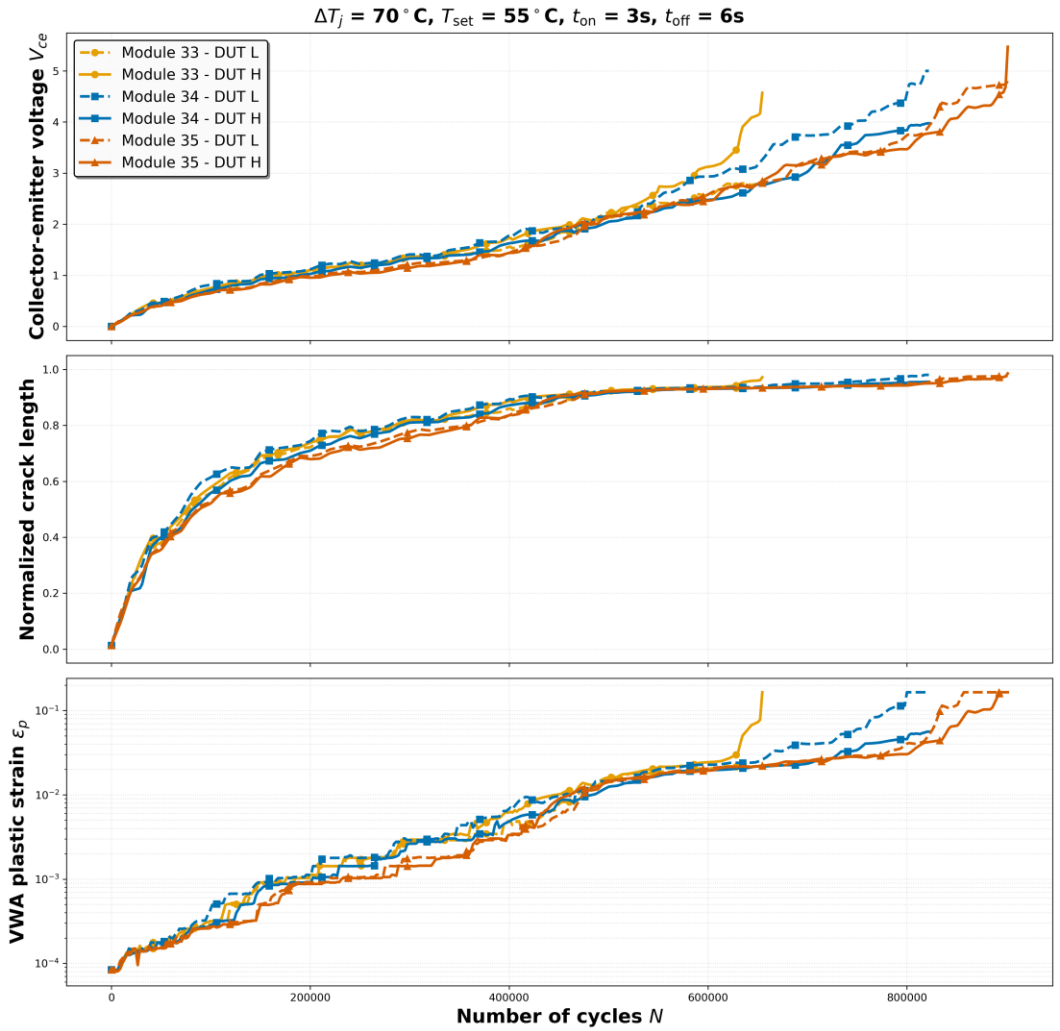
- Sparse representation in the loading conditions' space
- Lack of physical foundation for extrapolation



Campaign	$T_{set}(°C)$	ΔT(°C)	$t_{on}(s)$	$t_{off}(s)$
1	55	70	3	6
2	35	70	3	6
4	35	90	3	6
6	35	110	3	6
8	55	90	3	6
9	55	110	3	6

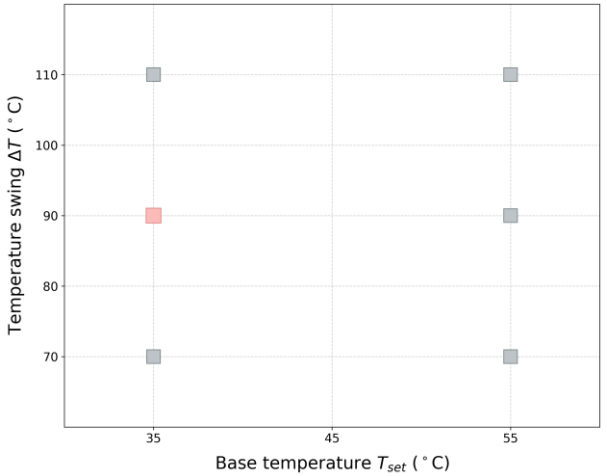
Cycling test campaigns (6 runs per campaign)

Intuition Sampling Evaluation Results BEDTime



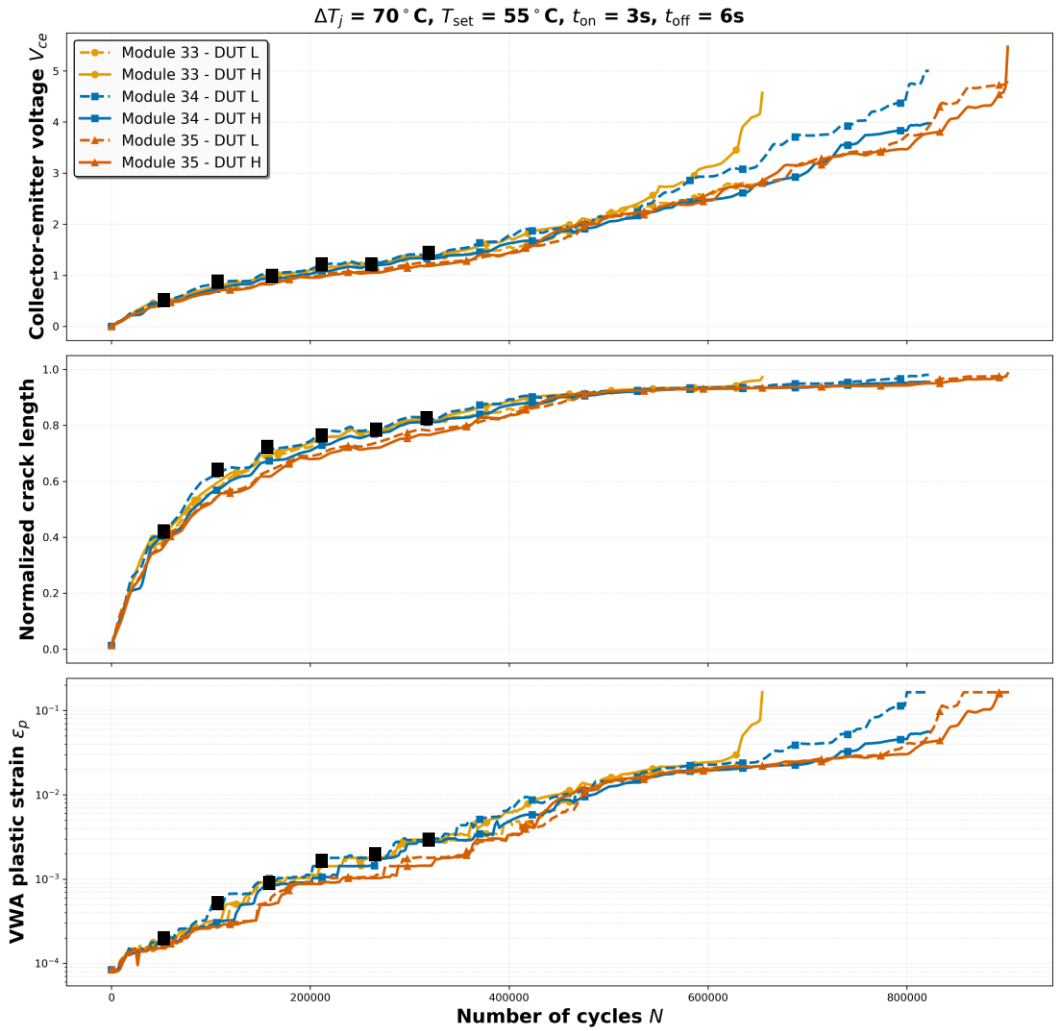
Physics-based data representation producing a degradation-aware multivariate signal

- Sparse representation in the loading conditions' space
- Interpolation is difficult due to sparsity



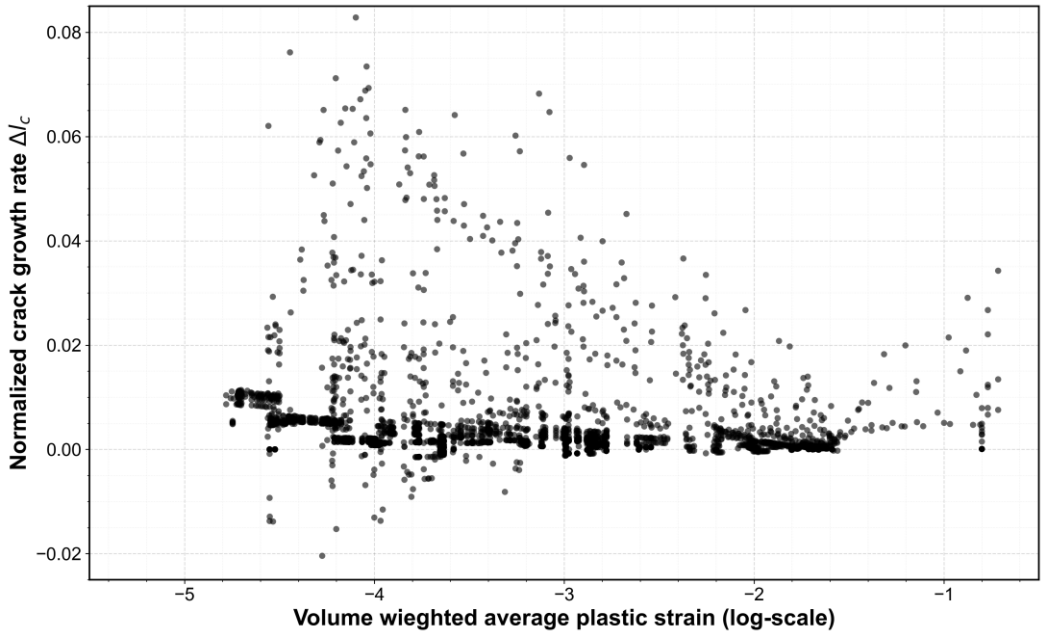
Campaign	$T_{set}(°C)$	ΔT(°C)	$t_{on}(s)$	$t_{off}(s)$
1	55	70	3	6
2	35	70	3	6
4	35	90	3	6
6	35	110	3	6
8	55	90	3	6
9	55	110	3	6

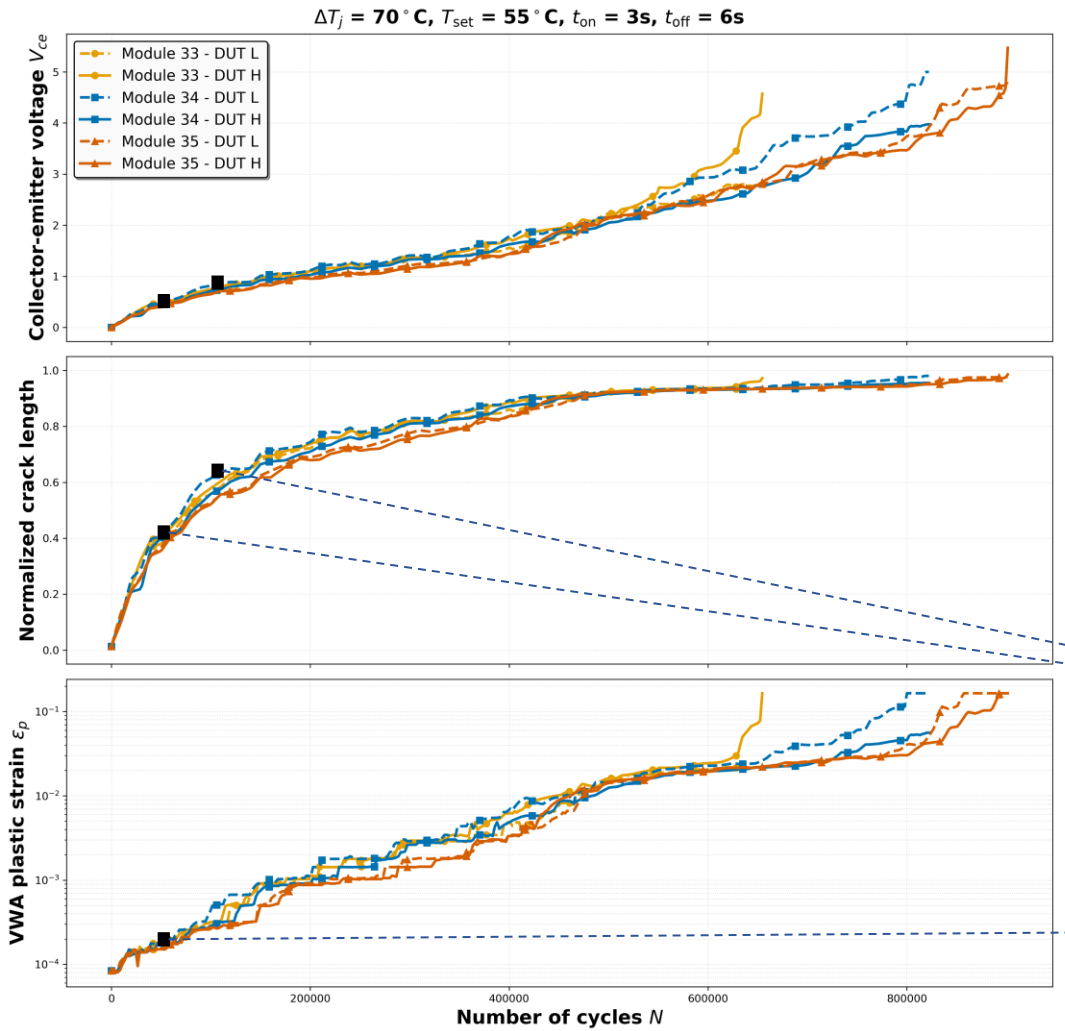
Cycling test campaigns (6 runs per campaign)



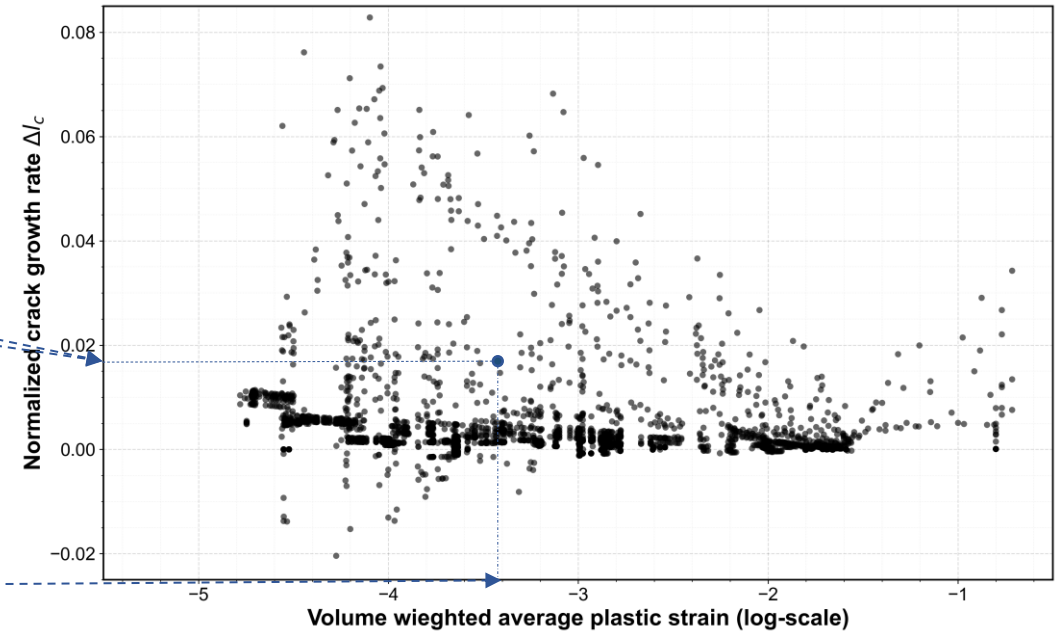
Physics-based data representation producing a degradation-aware multivariate signal

- Dense representation through crack growth increments and corresponding mechanical indicators





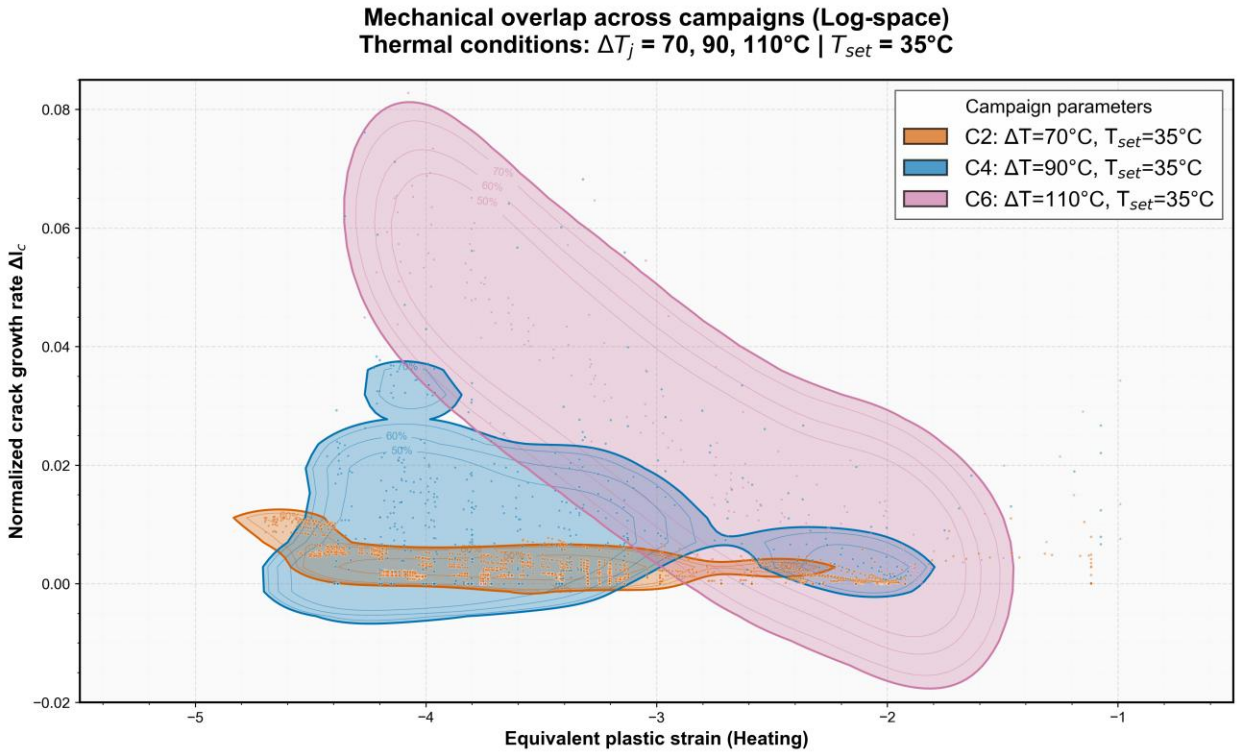
- Dense representation through crack growth increments and corresponding mechanical indicators



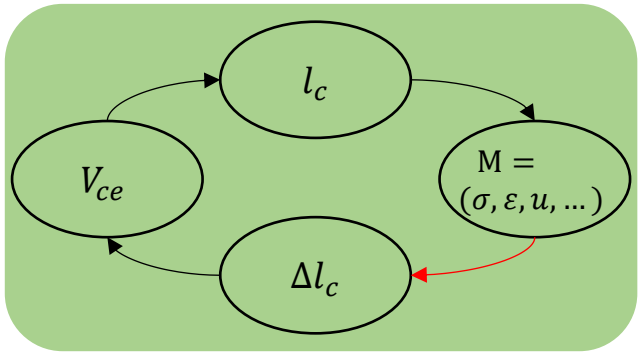
Physics-based data representation producing a degradation-aware multivariate signal

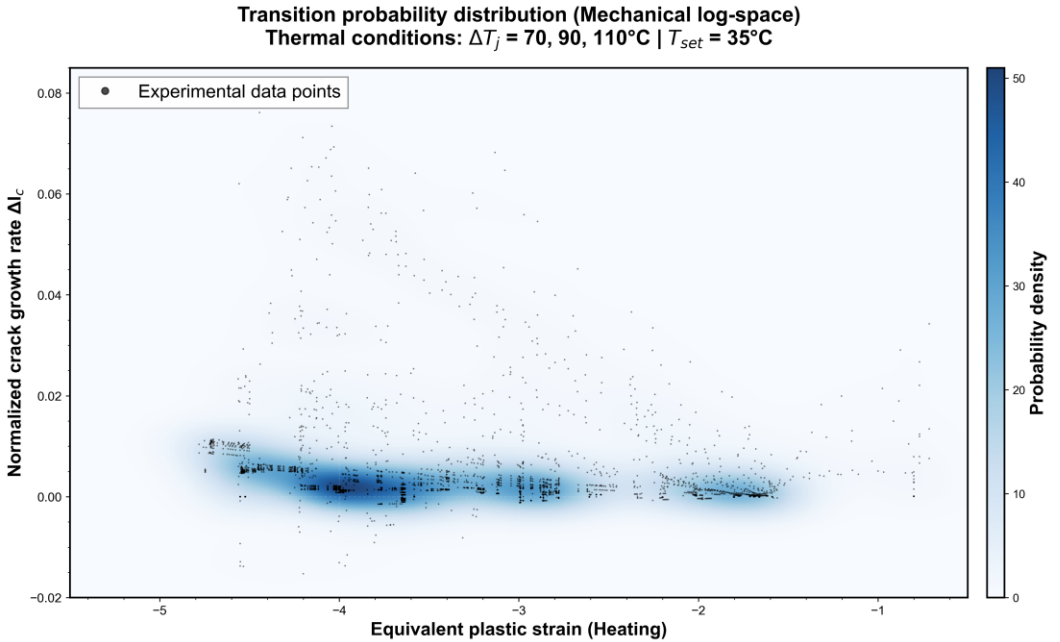
- The dense increment-based representation creates and overlap between experiments conducted at different conditions

⇒ physical foundation for interpolation and extrapolation through mechanical overlap

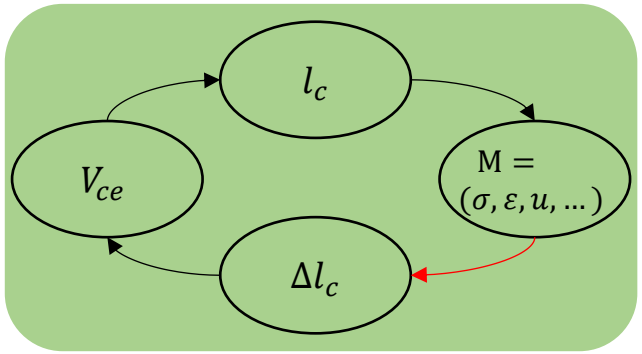


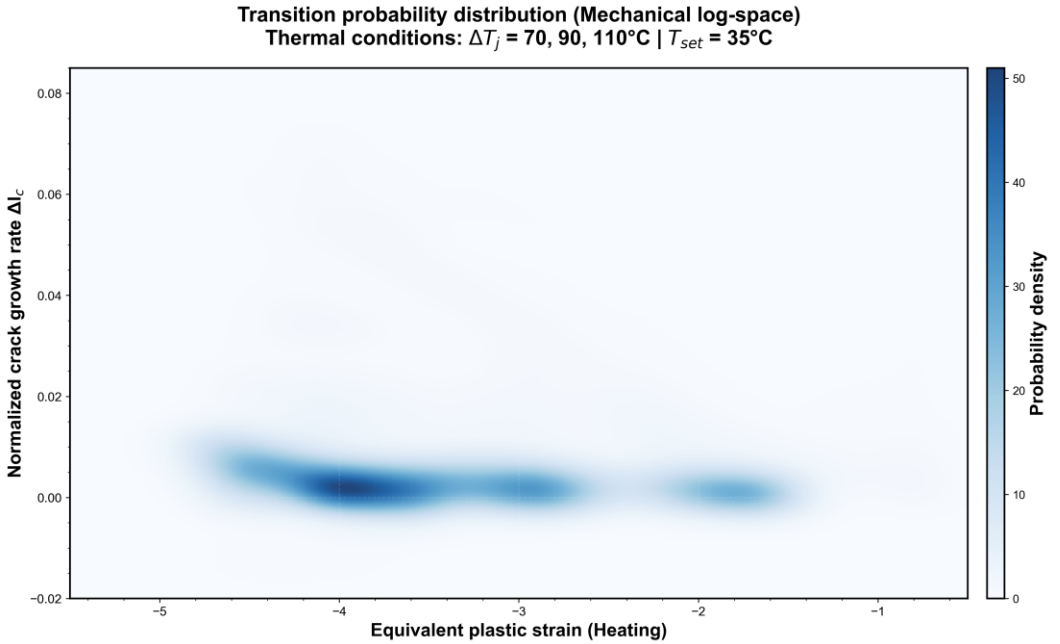
- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Estimate the joint distribution of the Markov chain using kernel density estimation $P(M, \Delta l_c)$



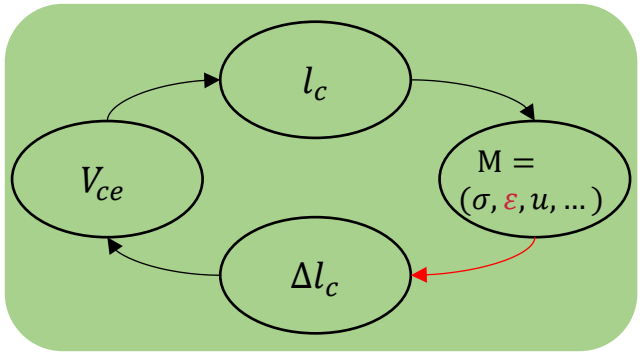


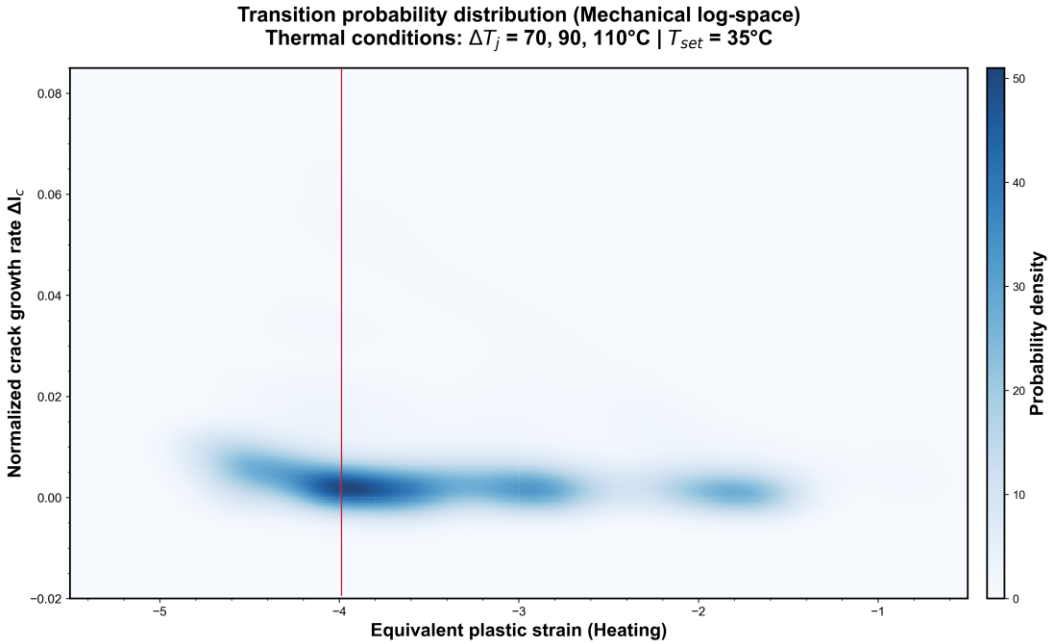
- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Use Gibbs to sample a candidate crack growth increment from the conditional probability $\Delta lc \sim P(.|M)$



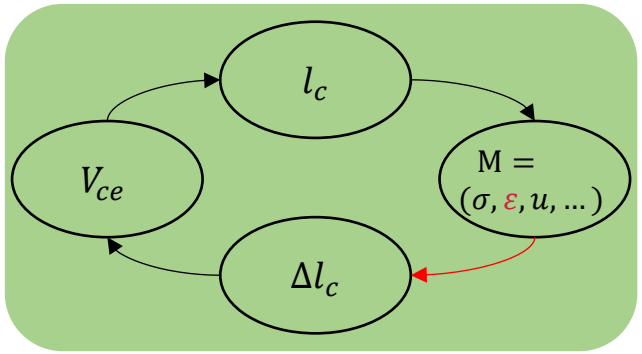


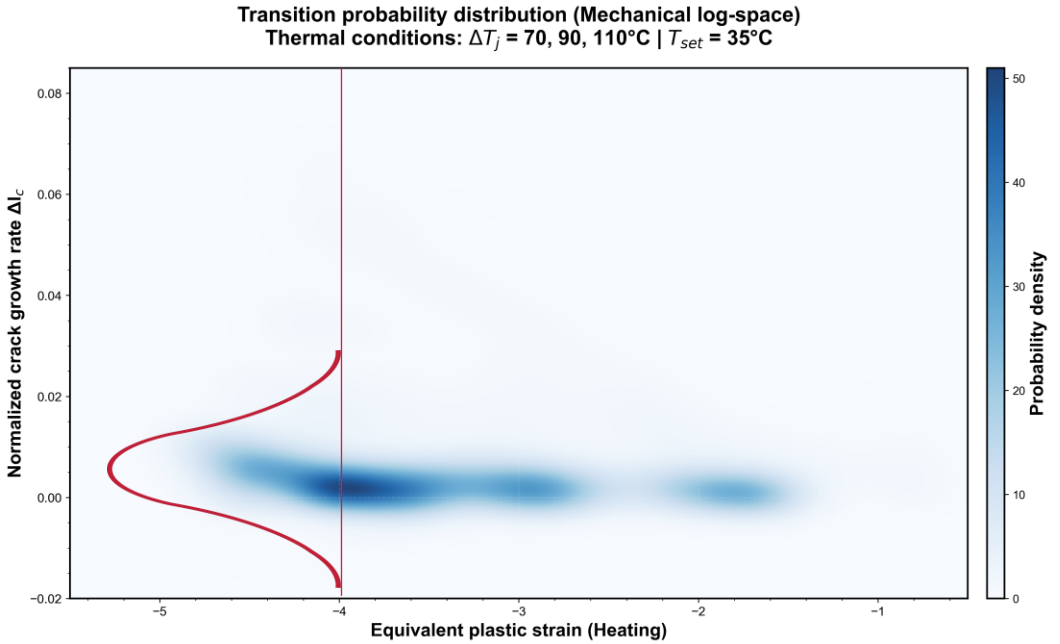
- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Use Gibbs to sample a candidate crack growth increment from the conditional probability $\Delta lc \sim P(.|M)$



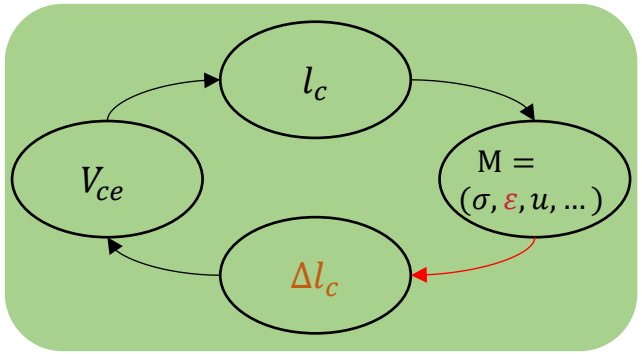


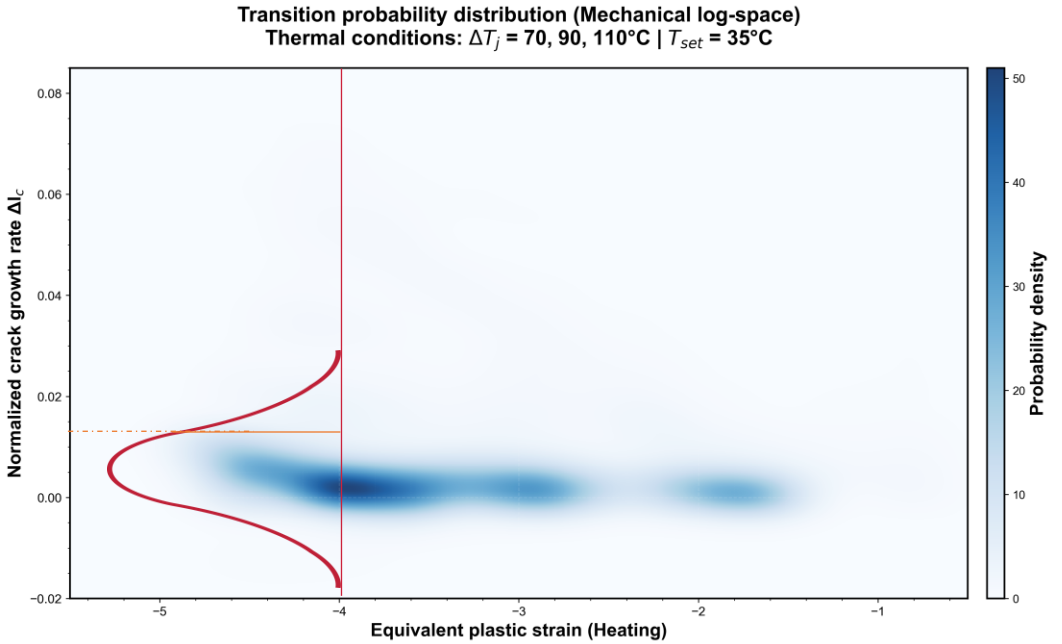
- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Use Gibbs to sample a candidate crack growth increment from the conditional probability $\Delta lc \sim P(.|M)$





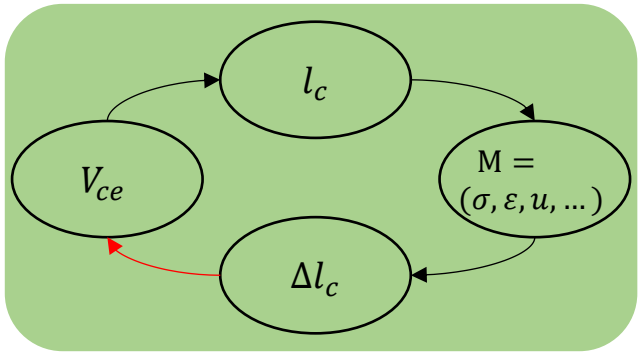
- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Use Gibbs to sample a candidate crack growth increment from the conditional probability $\Delta lc \sim P(.|M)$



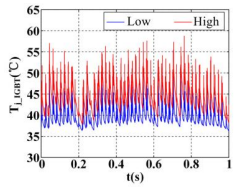


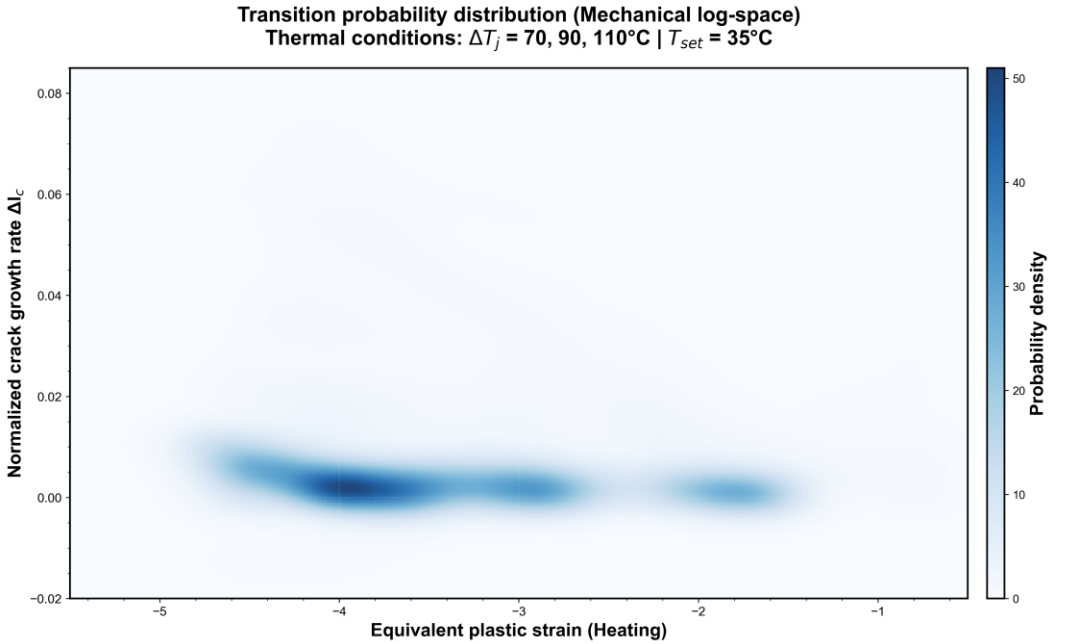
Intuition Sampling Evaluation Results BEDTime

- Increments represent transitions from consecutive states ⇒ Markov chain-based degradation model
- Given a set of mechanical indicators M, infer the corresponding degradation Δl_c
- Use Gibbs to sample a candidate crack growth increment from the conditional probability $\Delta lc \sim P(.|M)$



(Liu 2025)





- Validated via emulation of in-distribution and out-of-distribution scenarios ensuring reliability across varying experimental conditions
- Model initialization using a 10-observation historical window

Campaign	$T_{set}(°C)$	$\Delta T(°C)$	$t_{on}(s)$	$t_{off}(s)$
1	55	70	3	6
2	35	70	3	6
4	35	90	3	6
6	35	110	3	6
8	55	90	3	6
9	55	110	3	6

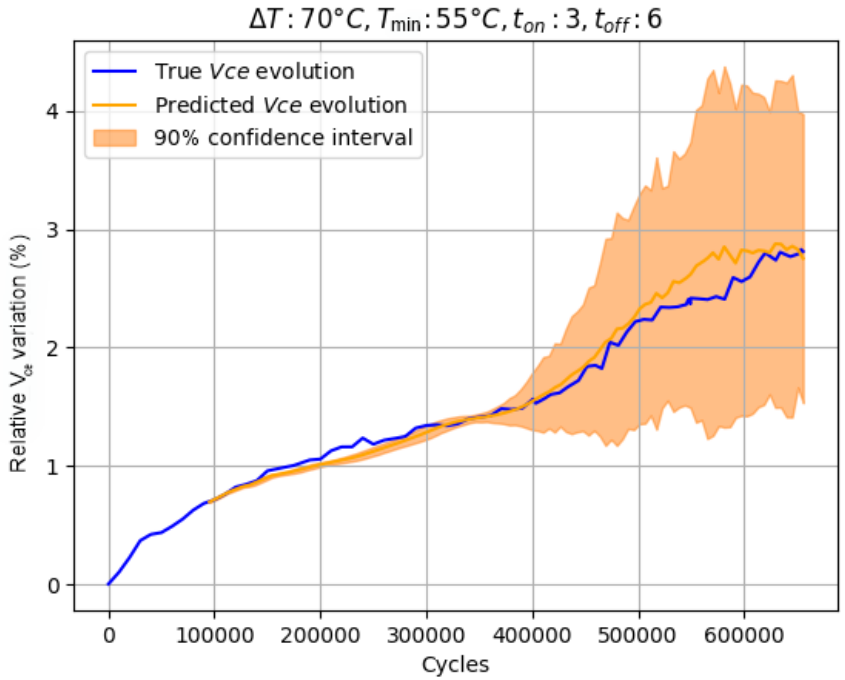
Cycling test campaigns (6 runs per campaign)

- Higher-order Markov Chains implemented to capture complex degradation dynamics and momentum.
- 100 Monte-Carlo iterations per prediction to generate 90% confidence intervals.

- In-distribution prediction:
  - Low KDE bandwidth due to the existence of in-distribution data
  - Tight confidence interval until reaching a high variance regime prior to failure
  - Coverage metrics indicate plausible predictions

Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
70	{70,90,110}	0,702	0,637	0,01
90	{70,90,110}	0,802	0,802	0,01
110	{70,90,110}	0,509	0,509	0,01

In-distribution prediction results

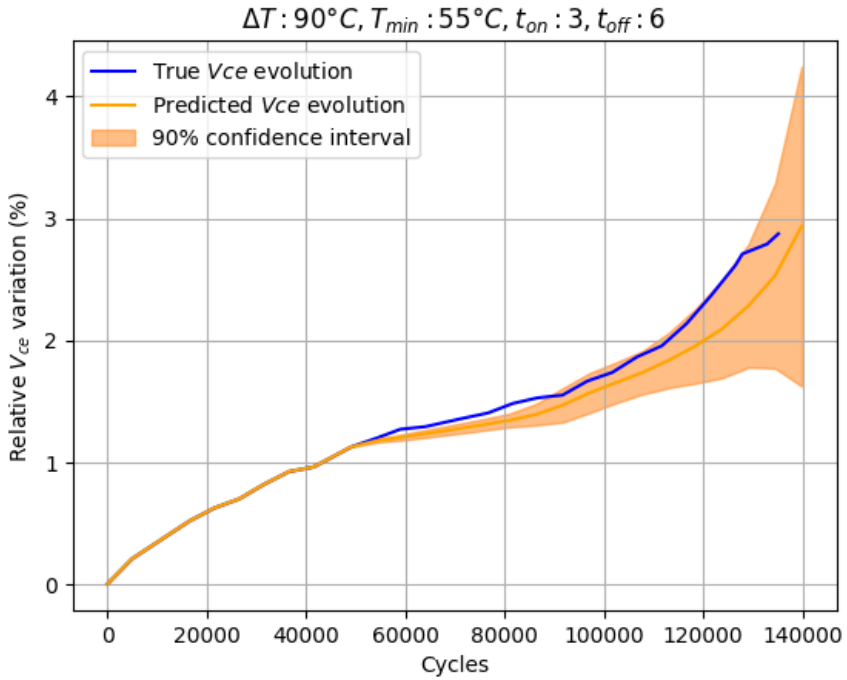


In-distribution predictions of a cycling test with a 90% confidence interval

- Out-of-distribution predictions (interpolation):
  - Low KDE bandwidth due to data proximity
  - Tight confidence interval indicating low variance in the Monte-Carlo simulations
  - Coverage metrics are lower than in-distribution predictions but remain plausible

Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
90	{70,110}	0,574	0,574	0,008

Out-of-distribution prediction results: interpolation

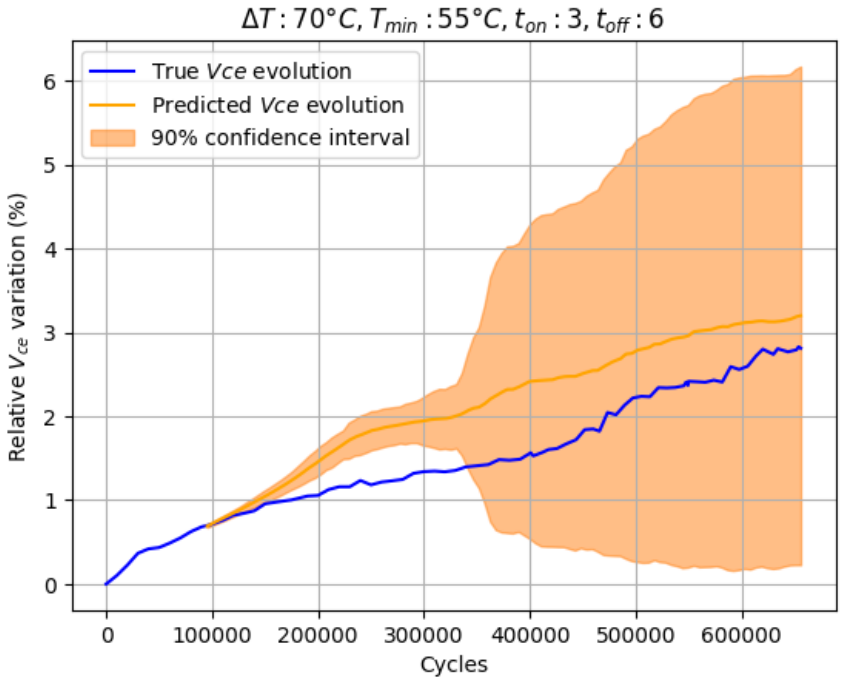


Out-of-distribution predictions of a cycling test with a 90% confidence interval (interpolation)

- Out-of-distribution predictions (extrapolation):
  - High KDE bandwidth due to significant ΔT mismatch
  - Large confidence intervals indicating high uncertainty
  - Pessimistic predictions due to higher temperature train data
  - Coverage metrics affected by confidence interval width

Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
70	{90,110}	0,615	0,563	0,19

Out-of-distribution prediction results: extrapolation



Out-of-distribution predictions of a cycling test with a 90% confidence interval (extrapolation)

- Combining physics-based representation with state-transition modeling enables effective information sharing across different cycling conditions
- The framework demonstrates high in-distribution prediction and robust performance in the interpolation task

Task	Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
In-distribution	70	{70,90,110}	0,702	0,637	0,01
In-distribution	90	{70,90,110}	0,802	0,802	0,01
In-distribution	110	{70,90,110}	0,509	0,509	0,01
Interpolation	90	{70,110}	0,574	0,574	0,008
Extrapolation	70	{90,110}	0,615	0,563	0,19

Prediction results across different tasks using the Markov chain-based lifetime estimation model

- While divergence increases in the extrapolation task, the model approximates the global trend and identifies the high variance regime
- Prediction uncertainty is seen through the expansion of the confidence interval, usually near module failure
- Potential for extension to variable loading profiles and broader experimental validation

Task	Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
In-distribution	70	{70,90,110}	0,702	0,637	0,01
In-distribution	90	{70,90,110}	0,802	0,802	0,01
In-distribution	110	{70,90,110}	0,509	0,509	0,01
Interpolation	90	{70,110}	0,574	0,574	0,008
Extrapolation	70	{90,110}	0,615	0,563	0,19

Prediction results across different tasks using the Markov chain-based lifetime estimation model

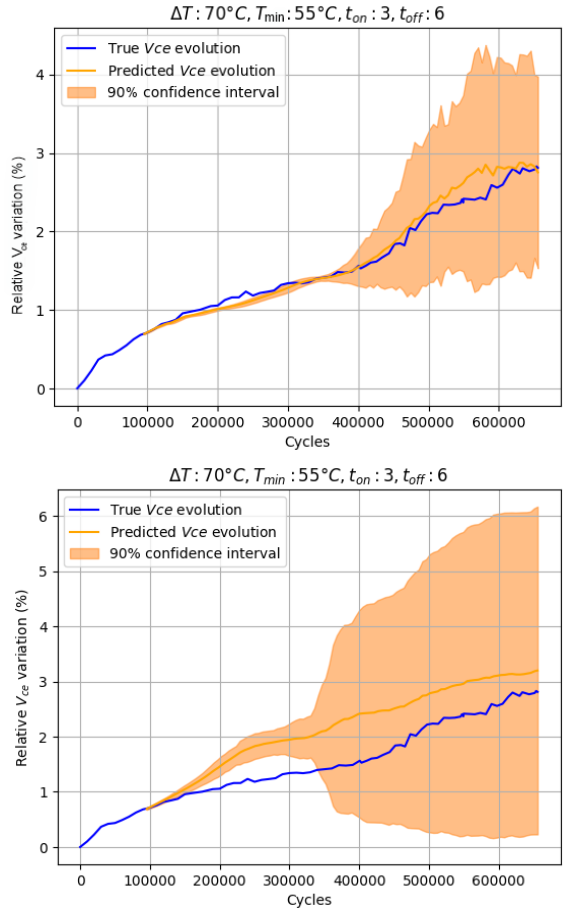
Intuition Sampling Evaluation Results BEDTime

- Adding meaningful cycling test data (Similar to the test data) yields substantial accuracy and uncertainty gains (extrapolation vs in-distribution predictions)
- Meaningful runs are costly and difficult to conduct

⇒ Partially conduct a costly run and exploit the information provided by the test

Task	Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
In-distribution	70	{70,90,110}	0,702	0,637	0,01
Extrapolation	70	{90,110}	0,615	0,563	0,19

Prediction results highlighting the effect of adding in-distribution samples

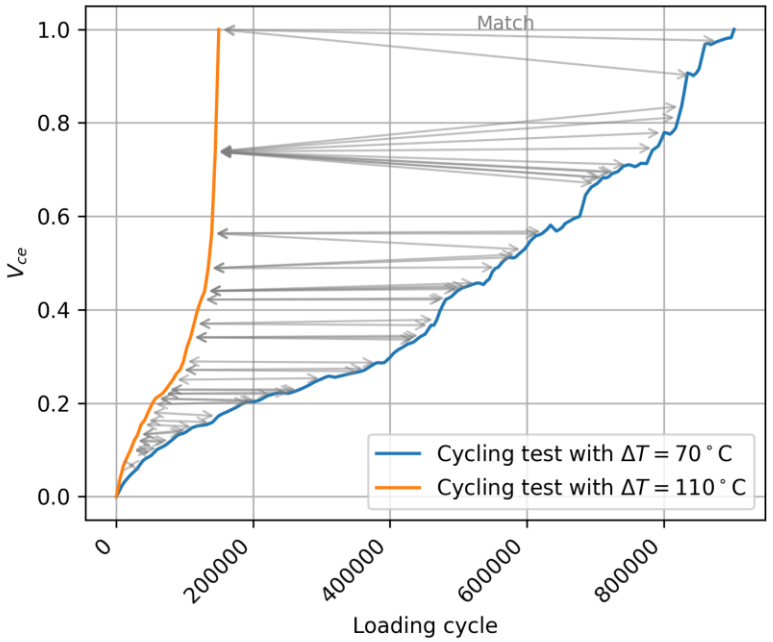


Comparison of predicting a run in the in-distribution (top) and out-of-distribution cases (bottom)

Intuition Sampling Evaluation Results BEDTime

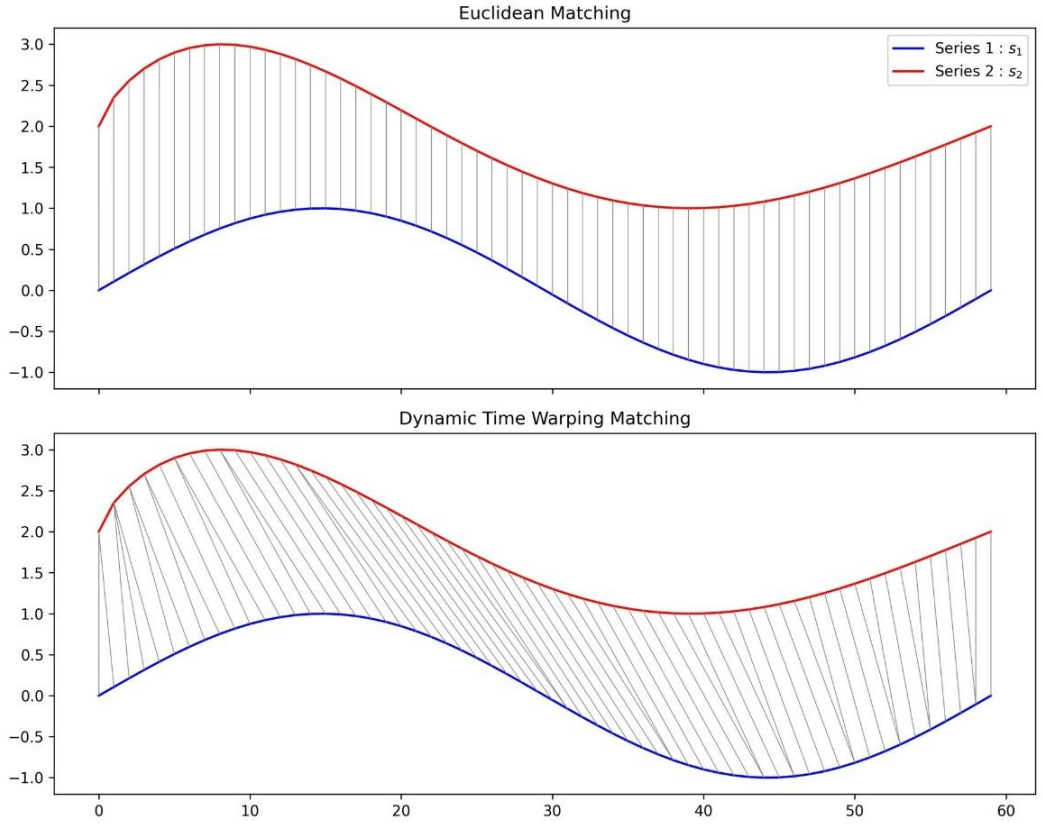
- Runs to failure on different conditions share mutual information present in the degradation curve
  - Health indicator curves share the same characteristic shape of steady evolution followed by a sharp increase signaling failure

Shape similarities in runs to failure under different conditions

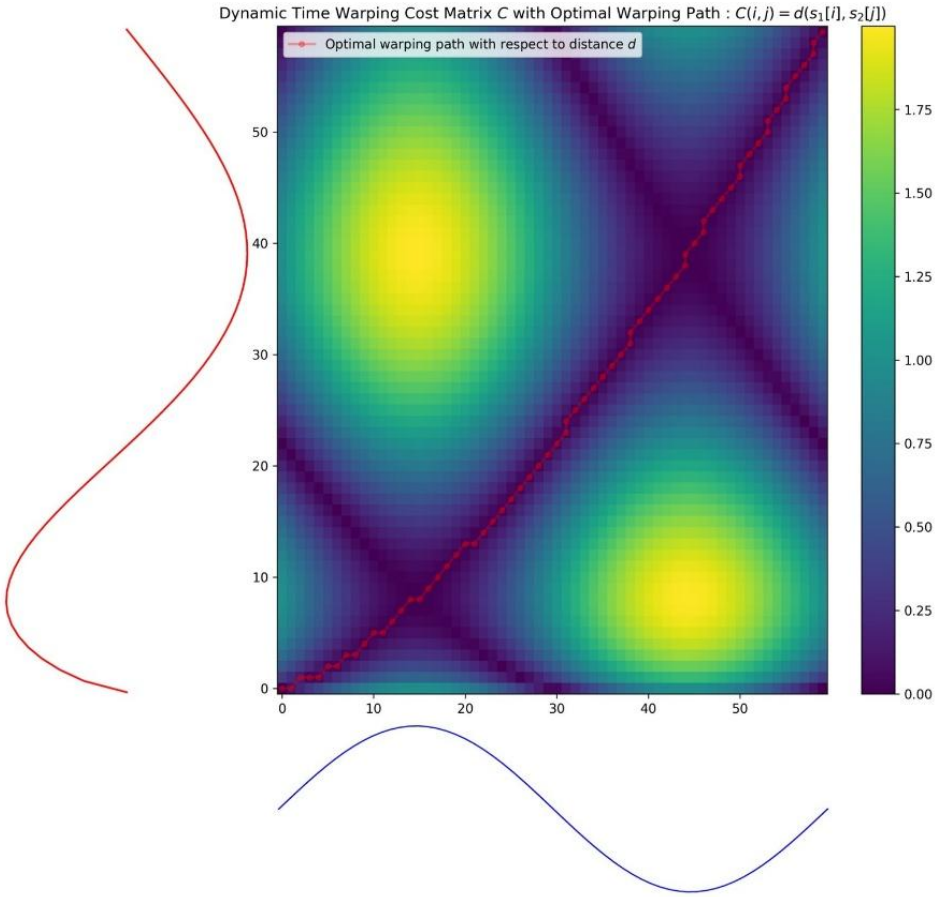


- Warping effect seen in health indicator data, as high temperature loading causes early failure (compression) compared to slower low temperature tests (stretching)

- Dynamic warping (DTW): a time series matching algorithm robust to time warping (compression and stretching)
- DTW aligns peaks and troughs appropriately despite temporal offset (shortcomings of Euclidean matching)



Robustness of DTW-base alignment in the presence of temporal distortion



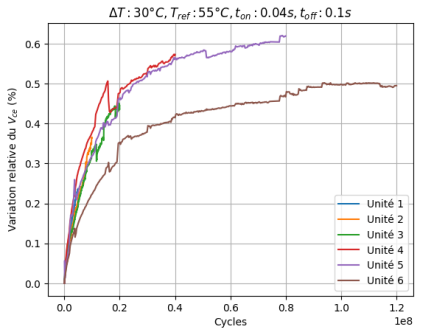
- DTW defines a cost matrix $\mathcal{C}$ using the signal values and a distance measure:

$$\mathcal{C}[i,j] = d(s_1[i], s_2[j])$$

- Matching is determined through the warping path: path of the lowest cost in the matrix $\mathcal{C}$ calculated using dynamic programming
- Holds information on the relative speed between both signals

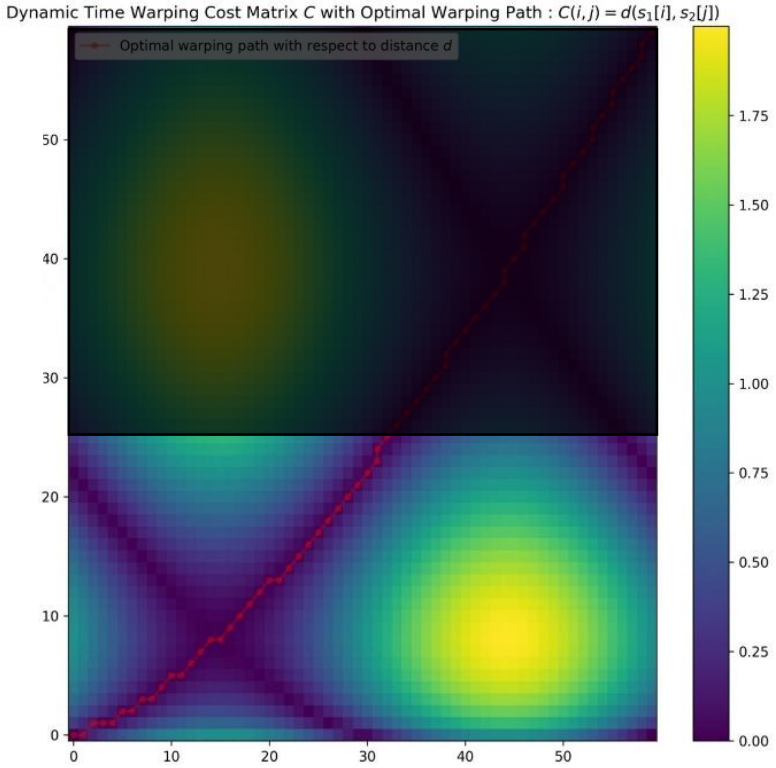
Intuition

- Partial observation of the signal ⇔ Partial observation of the cost matrix

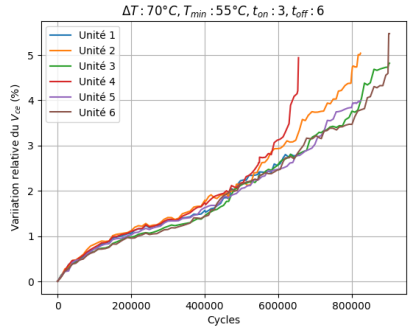


Partial run to failure (Target) to be completed based on a reference run

Sampling



Evaluation



Complete run to failure (Reference)

Results

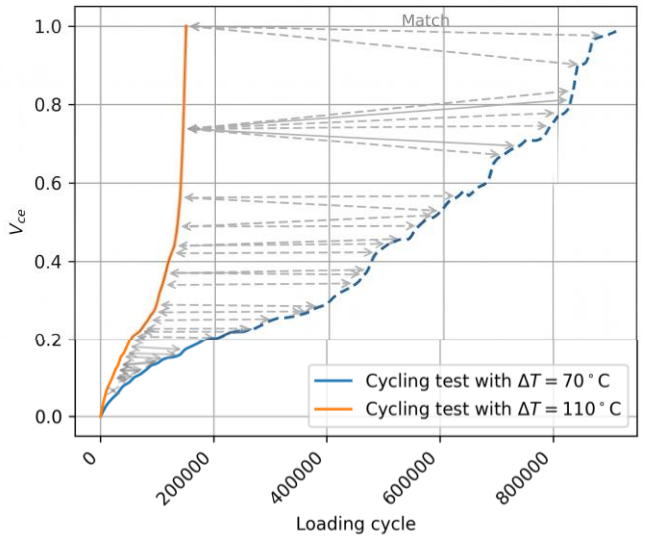
- Forecasting the warping path in the hidden area ⇒ **B**lindfolded **E**xploration using **D**ynamic **time** warping

⇒ Use time series models (backbone) trained on relevant warping data (≠ zero-shot) to forecast the partial warping in the future (ARIMA, Chronos, WaveNet, etc…)

BEDTime

1. Use dynamic time warping to compute the relative speed (Warping path) between a fast reference (High ΔT, low t_{on}) and a slow incomplete test (Low ΔT, high t_{on})
2. Forecast the warping path in the future using time series prediction models (Backbone)
3. Use the predicted warping path and the reference signal to complete the slow test ("Dynamic Time Unwarping")

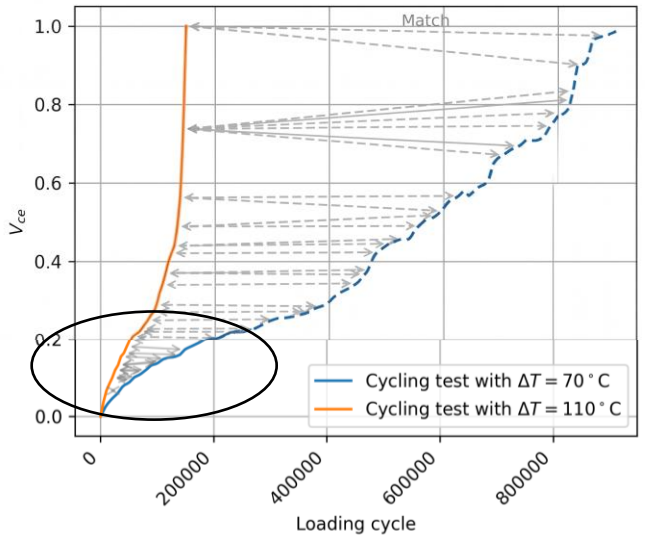
DTW-based alignment of health indicator evolution obtained under different conditions



Intuition Sampling Evaluation Results BEDTime

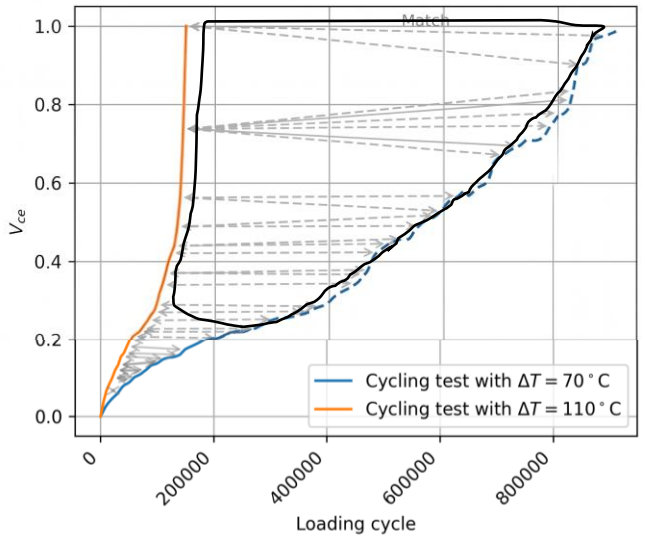
1. Use dynamic time warping to compute the relative speed (Warping path) between a fast reference (High ΔT, low t_{on}) and a slow incomplete test (Low ΔT, high t_{on})
2. Forecast the warping path in the future using time series prediction models (Backbone)
3. Use the predicted warping path and the reference signal to complete the slow test ("Dynamic Time Unwarping")

DTW-based alignment of health indicator evolution obtained under different conditions



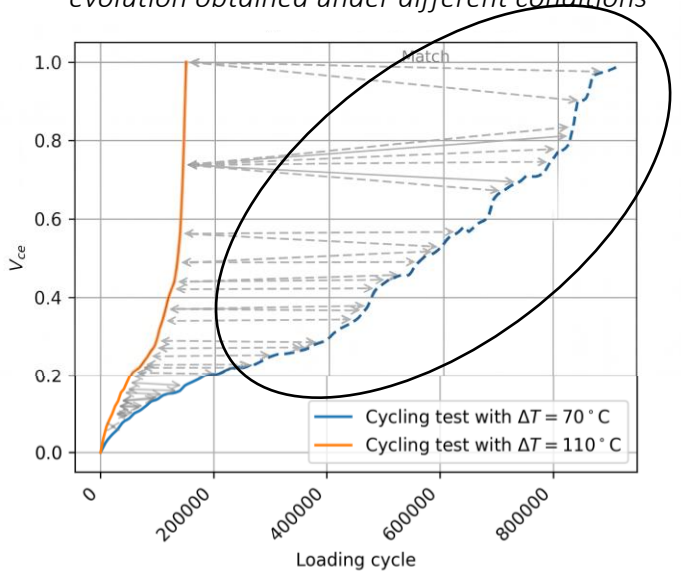
1. Use dynamic time warping to compute the relative speed (Warping path) between a fast reference (High ΔT, low t_{on}) and a slow incomplete test (Low ΔT, high t_{on})
2. Forecast the warping path in the future using time series prediction models (Backbone)
3. Use the predicted warping path and the reference signal to complete the slow test ("Dynamic Time Unwarping")

DTW-based alignment of health indicator evolution obtained under different conditions



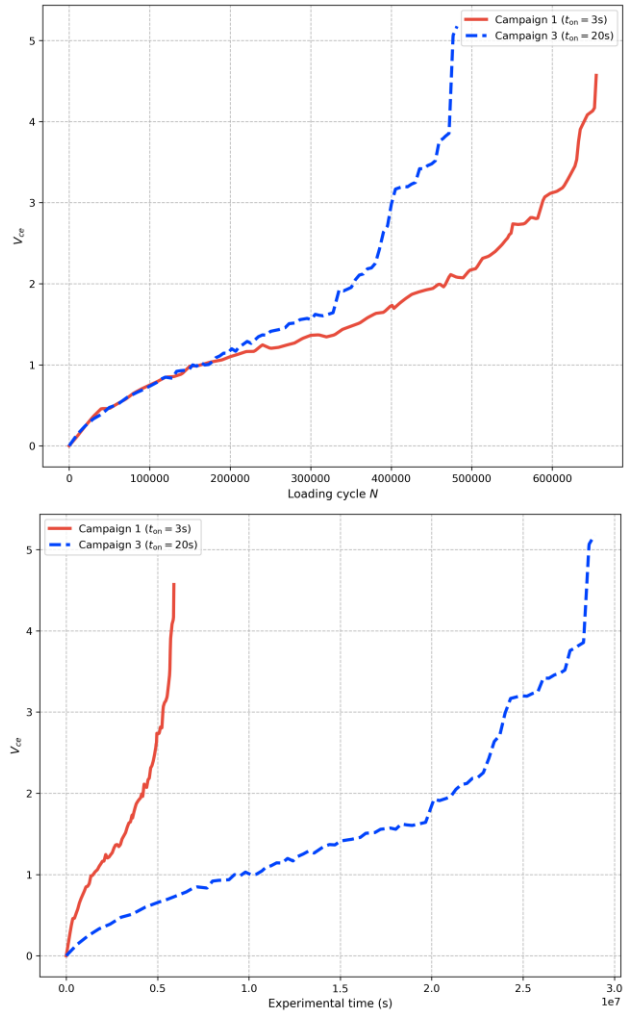
1. Use dynamic time warping to compute the relative speed (Warping path) between a fast reference (High ΔT, low t_{on}) and a slow incomplete test (Low ΔT, high t_{on})
2. Forecast the warping path in the future using time series prediction models (Backbone)
3. Use the predicted warping path and the reference signal to complete the slow test ("Dynamic Time Unwarping")

DTW-based alignment of health indicator evolution obtained under different conditions



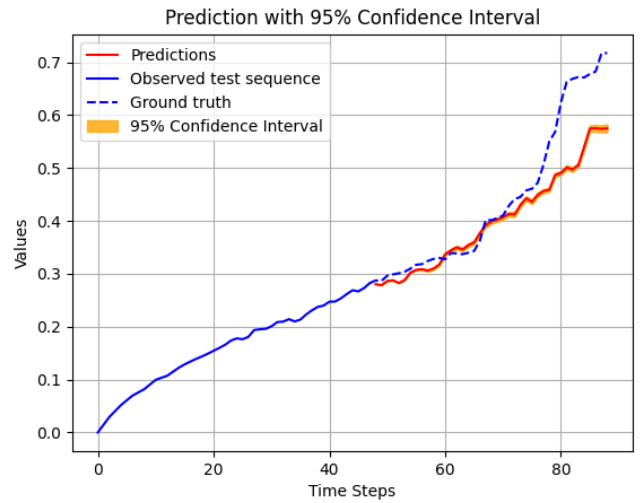
- Application: Predicting a slow target run obtained under loading time $t_{on} = 20s$ (campaign 3) using a fast reference run having $t_{on} = 3s$ (campaign 1).

- Cycling using shorter loading durations yields faster end-of-life, despite the harsher (yet slower) campaign reaches failure in fewer cycles.



Experimental time computation of the slow target (blue) and fast reference (red)

- Using BEDTime with a WaveNet backbone, we run predictions with the fast run-to-failure as a reference and the initial 50% of the slow run as a target to extrapolate.
- BEDTime accurately extrapolates the slow run to failure when using a faster reference.
- Despite slight mismatches and errors of the confidence interval, the overall trend and magnitude of the lifetime evolution is captured.
- Computationally, halting campaign 3's run at 50% would give a time gain ≈ 87 days, admitting BEDTime's errors



Prediction results of a slow run to failure using BEDTime and a fast reference

Main contributions Takeaways Future work

- Integrated degradation-aware simulation framework: Developed a novel modeling and automation pipeline that enables estimation of a module's mechanical response under loading while accounting for an arbitrary degradation state.
- Surrogate modeling: Performed an exhaustive benchmarking of machine learning surrogates as high-speed alternatives to numerical methods, thus enabling fast physics-informed feature engineering

Main contributions Takeaways Future work

- Probabilistic lifetime estimation: Developed a probabilistic model for power module reliability validated against both interpolation and extrapolation tasks, demonstrating predictive power in challenging reliability assessment scenarios.
- Few-shot time series prediction: End-to-end deployment of a few-shot time series prediction framework. This handles data characterized by time-warping effects, enabling forecasting even with limited historical datasets.

Main contributions Takeaways Future work

- Data and simulation scarcity: The lack of open-source datasets and simulation environments remains a primary bottleneck. The development of robust, ML-based solutions is currently constrained by the nature of proprietary industrial data.
- Need for standardized benchmarking: Notable absence of community-wide consensus on benchmarks. Without standardized tests and metrics, performing an objective comparative analysis of model performance remains a significant challenge.
- Data-centric experimental design: To fully leverage the power of data-driven approaches, experimental protocols should be designed with data requirements in mind, prioritizing feature diversity, sampling consistency, and metadata richness.

Main contributions Takeaways Future work

- Physics-informed surrogates based on non-linear material models: Incorporate visco-plastic material models in simulations, accounting for rate-dependent plasticity and developing physics-informed surrogates according to Anand's equations.
- Physics-informed neural networks: Exploit the physics-based data representation and relevant physical laws to harness neural networks' predictive power despite the low-data regime

Main contributions Takeaways Future work

- Custom BEDTime backbone development: Design a specialized architecture for warping signals:
  - Leverage large-scale warping datasets for pretraining to improve zero-shot and few-shot generalization (Chronos)
  - Integrate specialized architectures to model transition probability signals (WaveNet)
- Active learning validation: Conduct extensive experimental validation of the proposed active learning framework to quantify the reduction in required simulations and experimental time to reach a target accuracy.

Thank you for your attention

Appendix

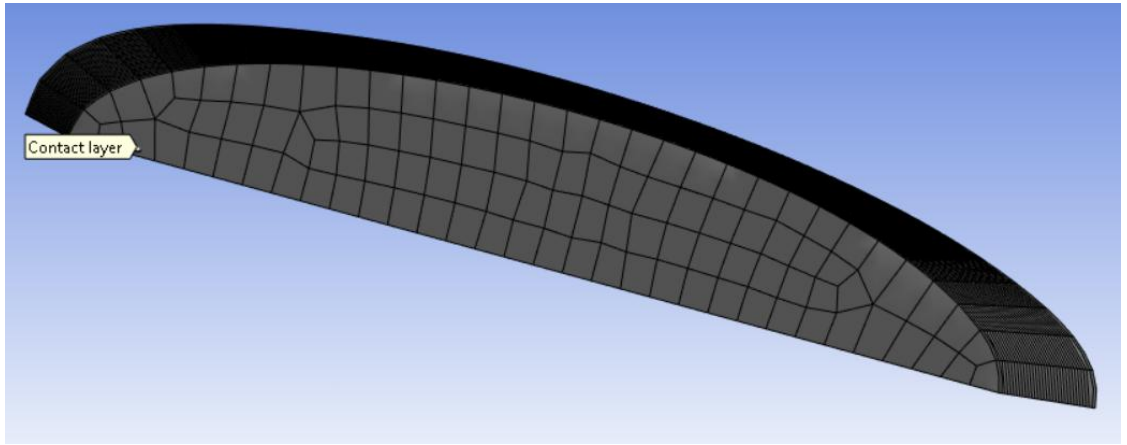
- The method assumes that the discrete solution of a finite element model $f(h)$ approaches a theoretical exact solution f_{exact} as the grid spacing h approaches zero.
- This relationship is expressed through a Taylor series expansion of the discrete solution: $f(h) \approx f_{exact} + C_{FEM} \cdot h^{P_{FEM}} + \mathcal{O}(h^{P_{FEM}+1})$
  - C_{FEM}: a constant dependent on the specific problem being solved.
  - term P_{FEM} represents the order of convergence, which dictates how quickly the error reduces as the mesh is refined.
- To solve for these unknowns, the analysis requires solutions from three distinct mesh settings: fine (h_1), medium (h_2), and coarse (h_3).

Appendix

- A constant refinement ratio $R_{GCI} = \frac{h_3}{h_2} = \frac{h_2}{h_1}$ must be maintained across these three settings to ensure mathematical consistency.
- By using the solutions$f(h_1), f(h_2), f(h_3)$ from these three meshes the actual order of convergence $P_{GCI}$is calculated as follows: $P_{GCI} = \frac{\ln\left(\frac{f(h_3)-f(h_2)}{f(h_2)-f(h_1)}\right)}{\ln(R_{GCI})}$
- The GCI method utilizes Richardson Extrapolation to estimate the theoretical exact mechanical values that would be achieved with an infinitely fine mesh.
- The GCI value itself provides a confidence interval for the fine grid solution, indicating how much further the solution might change with even more refinement.

Appendix

- GCI is calculated using the relative error between the fine and medium grids ε_{21} and a safety factor F_s set to $1{,}25$ $\mathrm{GCI}_{12} = \frac{\mathrm{F_S} \cdot |\epsilon_{21}|}{\mathrm{R}_{\mathrm{GCI}}^{\mathrm{P_{GCI}}} - 1}$
- Meshes were defined by the number of divisions n_D and an averaging radius r_{wva}, rather than direct element size h, necessitating an additional verification step for refinement consistency.
- Convergence behavior varied significantly across different physical quantities (e.g., equivalent plastic strain vs. strain energy).



Appendix

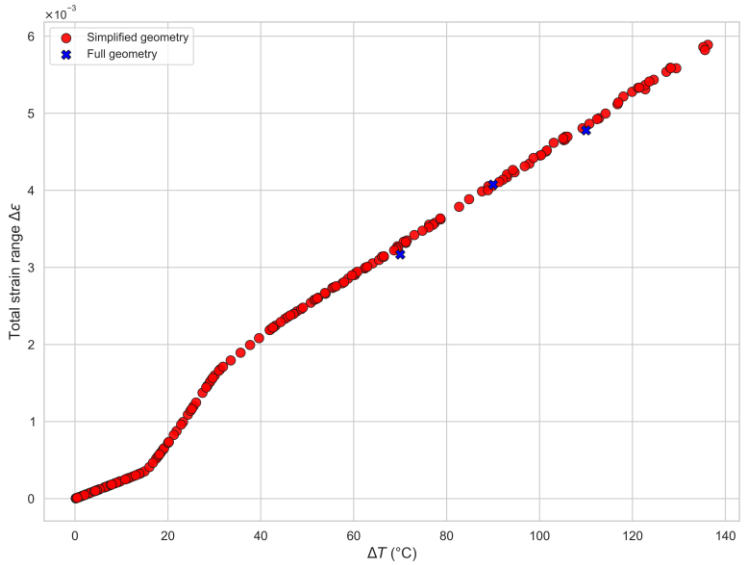
- Large radii: Caused undesirable smoothing, making the solution insensitive to refinement.
- Small radii: Led to statistical instability due to insufficient element counts within the averaging volume.
- Optimization Goal: Targeted a GCI < 5% and an order of convergence $P_{GCI} \approx 1 - 2$
- Final Mesh Triplet: $n_D = [16, 20, 32]$ (Coarse, Medium, Fine).
- Averaging Radius: $r_{wva} = 0.6 \times R_h$
- Achieved Metrics (for Equivalent Plastic Strain):
  - Average Refinement Ratio (ARR): 1.513 (with 9.3% deviation).
  - GCI: 3.935%
  - Order of Convergence P_{GCI}: 0.63.
  - Selection Rationale: This configuration offered the most robust compromise across all primary variables of interest for quantifying degradation.

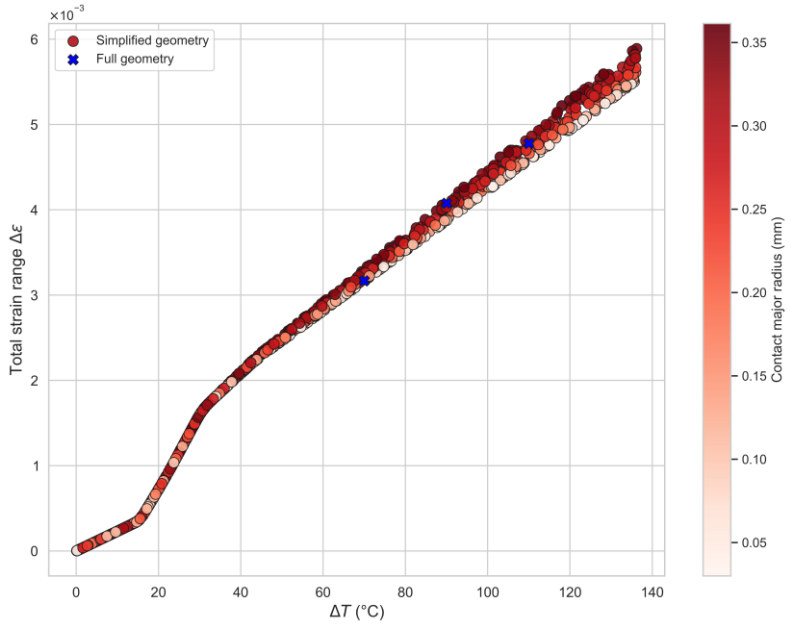
Appendix

	Quantity	Radius_Factor	f_fine	f_medium	f_coarse	n_fine_zone	n_medium_zone	n_coarse_zone	p	f_exact	gci_fine	asymptotic_ratio	r21	r32	r_diff_percent	r_consistent
0	Stress_Eqv	0.40	2.7801e+07	2.7765e+07	2.8262e+07	36	16	4	7.04	2.7808e+07	0.029	-0.4930	1.310	1.587	19.1	False
1	Stress_Eqv	0.45	2.7868e+07	2.7899e+07	2.8295e+07	42	19	5	7.13	2.7863e+07	0.025	0.5101	1.303	1.560	18.0	False
2	Stress_Eqv	0.50	2.7797e+07	2.7979e+07	2.8303e+07	74	23	7	1.47	2.7562e+07	1.054	0.9948	1.476	1.487	0.7	True
3	Stress_Eqv	0.55	2.7305e+07	2.8056e+07	2.8281e+07	101	27	9	-2.99	2.8333e+07	-4.706	0.8978	1.552	1.442	7.4	True
4	Stress_Eqv	0.60	2.7504e+07	2.8137e+07	2.8281e+07	119	30	10	-3.57	2.8289e+07	-3.569	0.8501	1.583	1.442	9.3	True
5	Stress_Eqv	0.65	2.7632e+07	2.7984e+07	2.8247e+07	139	34	18	-0.84	2.8713e+07	-4.887	0.9038	1.599	1.236	25.6	False
6	Stress_Eqv	0.70	2.7740e+07	2.8008e+07	2.8285e+07	159	38	24	0.10	2.2486e+07	23.675	1.0156	1.611	1.166	32.1	False
7	Stress_Eqv	0.75	2.7834e+07	2.8081e+07	2.8342e+07	194	54	33	0.18	2.4704e+07	14.056	1.0221	1.532	1.178	26.1	False
8	Stress_Eqv	0.80	2.7910e+07	2.8102e+07	2.8363e+07	219	76	41	1.10	2.7506e+07	1.811	1.0810	1.423	1.228	14.7	True
9	Stress_Eqv	0.85	2.7873e+07	2.8188e+07	2.8394e+07	268	90	46	-1.44	2.8645e+07	-3.461	0.9070	1.439	1.251	14.0	True
10	Stress_Eqv	0.90	2.7827e+07	2.8250e+07	2.8386e+07	311	103	54	-3.85	2.8385e+07	-2.510	0.7533	1.445	1.240	15.3	False
11	Stress_Eqv	0.95	2.7866e+07	2.8299e+07	2.8377e+07	346	113	57	-5.66	2.8359e+07	-2.210	0.6735	1.452	1.256	14.5	True
12	Stress_Eqv	1.00	2.7933e+07	2.8271e+07	2.8385e+07	399	124	64	-3.50	2.8387e+07	-2.031	0.7533	1.476	1.247	16.9	False
13	PI_Strain_Eqv	0.40	1.8973e-02	1.6733e-02	1.4846e-02	36	16	4	-0.46	-4.9988e-05	-125.329	1.0477	1.310	1.587	19.1	False
14	PI_Strain_Eqv	0.45	1.8567e-02	1.6437e-02	1.4043e-02	42	19	5	0.32	4.2310e-02	159.844	0.9698	1.303	1.560	18.0	False
15	PI_Strain_Eqv	0.50	1.8203e-02	1.5968e-02	1.4133e-02	74	23	7	-0.50	5.5861e-03	-86.640	1.0018	1.476	1.487	0.7	True
16	PI_Strain_Eqv	0.55	1.5753e-02	1.5697e-02	1.4468e-02	101	27	9	7.65	1.5755e-02	0.016	1.3182	1.552	1.442	7.4	True
17	PI_Strain_Eqv	0.60	1.4736e-02	1.4581e-02	1.4379e-02	119	30	10	0.63	1.5200e-02	3.935	1.0290	1.583	1.442	9.3	True
18	PI_Strain_Eqv	0.65	1.4286e-02	1.3994e-02	1.3595e-02	139	34	18	0.89	1.4851e-02	4.947	1.1128	1.599	1.236	25.6	False
19	PI_Strain_Eqv	0.70	1.3454e-02	1.3472e-02	1.2632e-02	159	38	24	11.65	1.3454e-02	0.001	-5.6648	1.611	1.166	32.1	False
20	PI_Strain_Eqv	0.75	1.3341e-02	1.3569e-02	1.2093e-02	194	54	33	6.16	1.3323e-02	0.167	-2.1255	1.532	1.178	26.1	False
21	PI_Strain_Eqv	0.80	1.3194e-02	1.3196e-02	1.1098e-02	219	76	41	24.67	1.3194e-02	0.000	-5.7401	1.423	1.228	14.7	True
22	PI_Strain_Eqv	0.85	1.3529e-02	1.2473e-02	1.0477e-02	268	90	46	2.15	1.4416e-02	8.202	1.1566	1.439	1.251	14.0	True
23	PI_Strain_Eqv	0.90	1.2743e-02	1.1682e-02	1.0001e-02	311	103	54	1.56	1.4104e-02	13.352	1.1220	1.445	1.240	15.3	False
24	PI_Strain_Eqv	0.95	1.2152e-02	1.1147e-02	1.0003e-02	346	113	57	0.43	1.7959e-02	59.738	1.0303	1.452	1.256	14.5	True
25	PI_Strain_Eqv	1.00	1.1912e-02	1.1054e-02	9.7020e-03	399	124	64	1.47	1.3021e-02	11.634	1.1266	1.476	1.247	16.9	False

Appendix

- Compared a dense point cloud from 1,000 simplified simulations against sparse reference data from healthy full-geometry models.
- .Reference points aligned with the upper region of the simulation cloud (maximal contact length), confirming the model accurately reproduces healthy-state mechanics.
- Verified alignment by highlighting simulations with minimal crack length ($< 10\%$ degradation), showing a clear overlap between simplified and full-geometry models
- Employed the Maximum Mean Discrepancy (MMD) two-sample test to quantitatively determine if simplified and full-geometry datasets share the same underlying distribution.

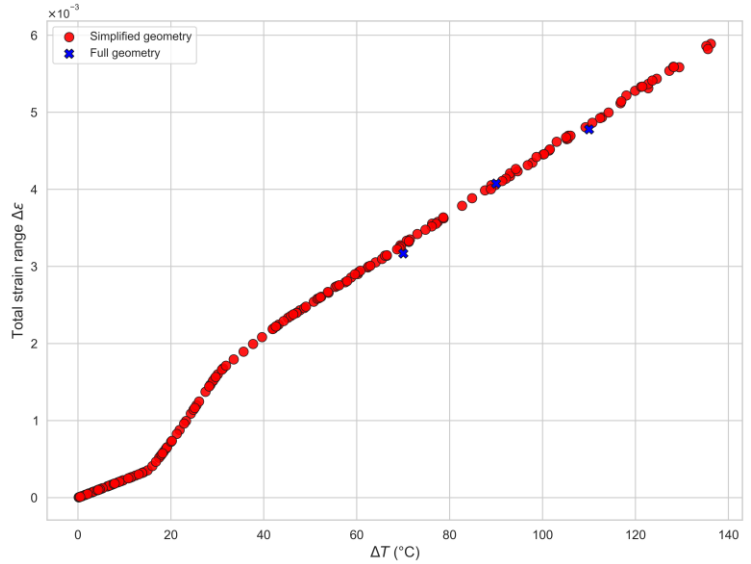


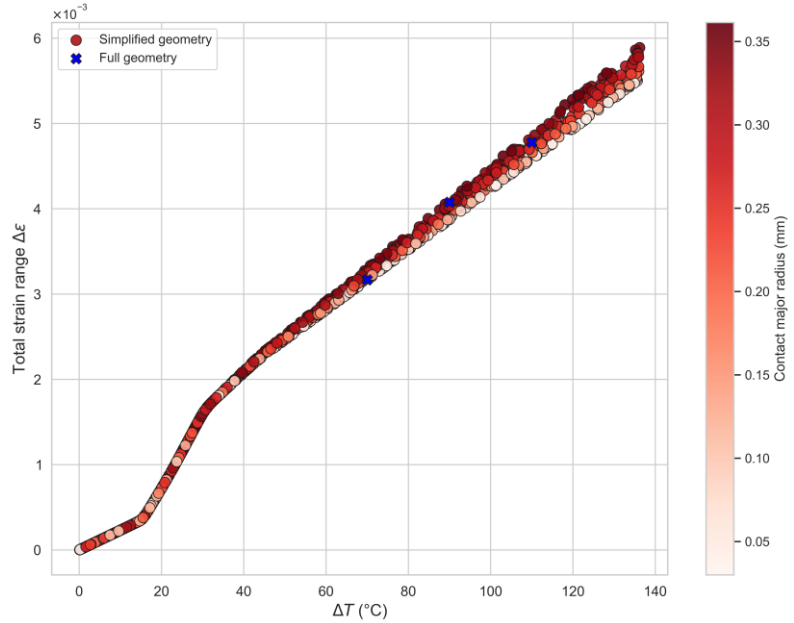


Appendix

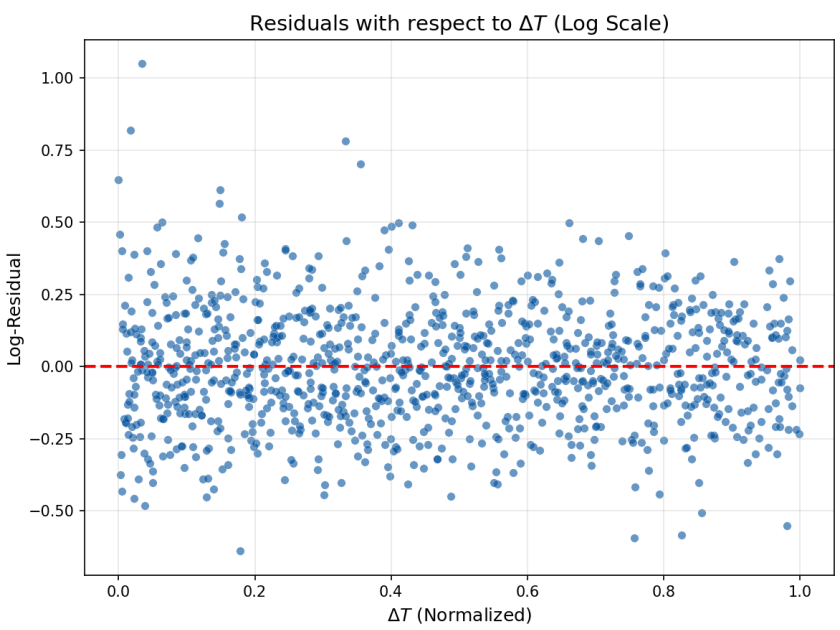
- MMD projects distributions into a Reproducing Kernel Hilbert Space (RKHS) to measure dissimilarity as the distance between mean projections
- $\widehat{\mathrm{MMD}}^2(X,Y) = \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i, y_j) + \frac{1}{m^2}\sum_{i,j=1}^{m} k(y_i, y_j)$

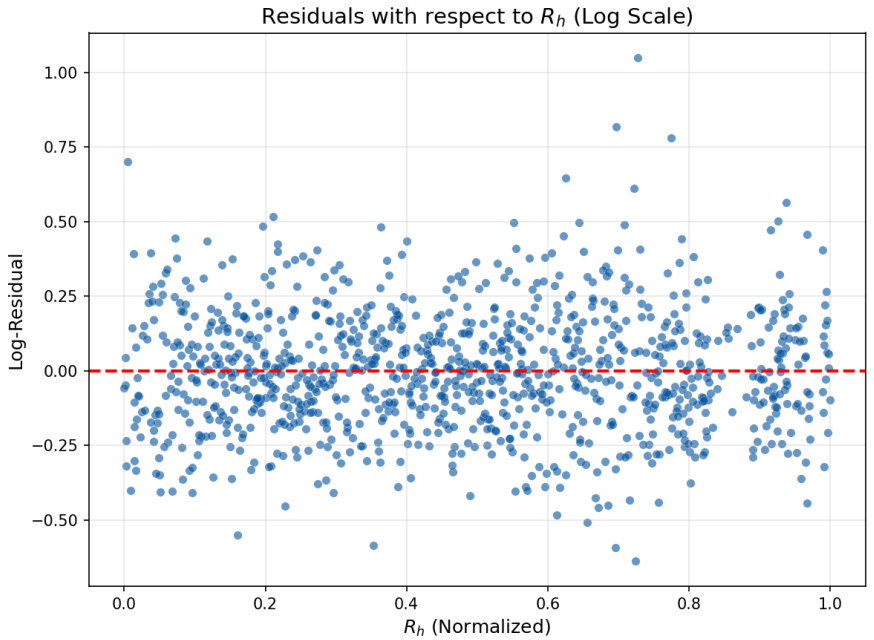
Max l_c allowed	Min l_c allowed	MMD^2	p-value
100%	75%	0.090	0.57
75%	50%	0.084	0.60
50%	25%	0.090	0.57
25%	0%	0.080	0.61





Appendix





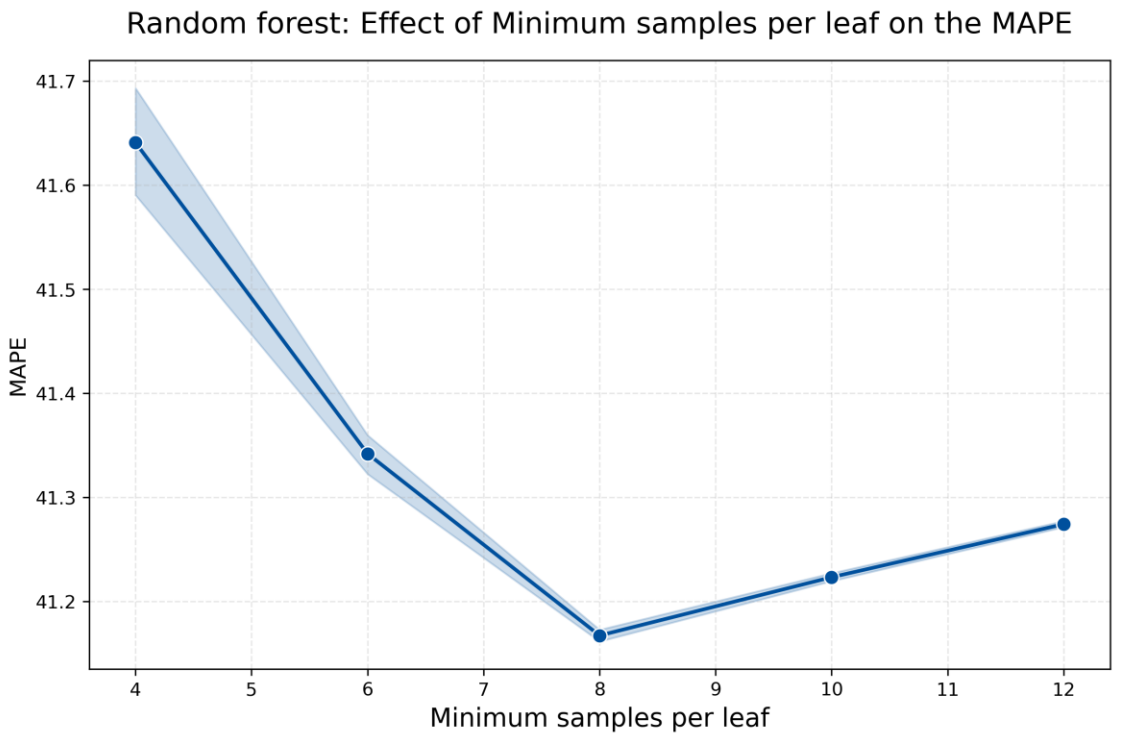
Minimal bias in the residuals and uniform variance distribution

Appendix

- Metallization layer and wire (Al) : Bilinear Kinematic Hardening :
  - Yield stress σ = 29MPa
  - Tangent modulus E = 35.5MPa
- Copper : Chaboche Kinematic Hardening : 2 modes :
  - T = 20◦C : C1 = 54.041MPa, γ1 = 962, C2 = 721MPa, γ2 = 1.1
  - T = 50◦C : C1 = 52.880MPa, γ1 = 1000, C2 = 700MPa, γ2 = 1.1
  - T = 150◦C : C1 = 45.760MPa, γ1 = 1100, C2 = 600MPa, γ2 = 1.1

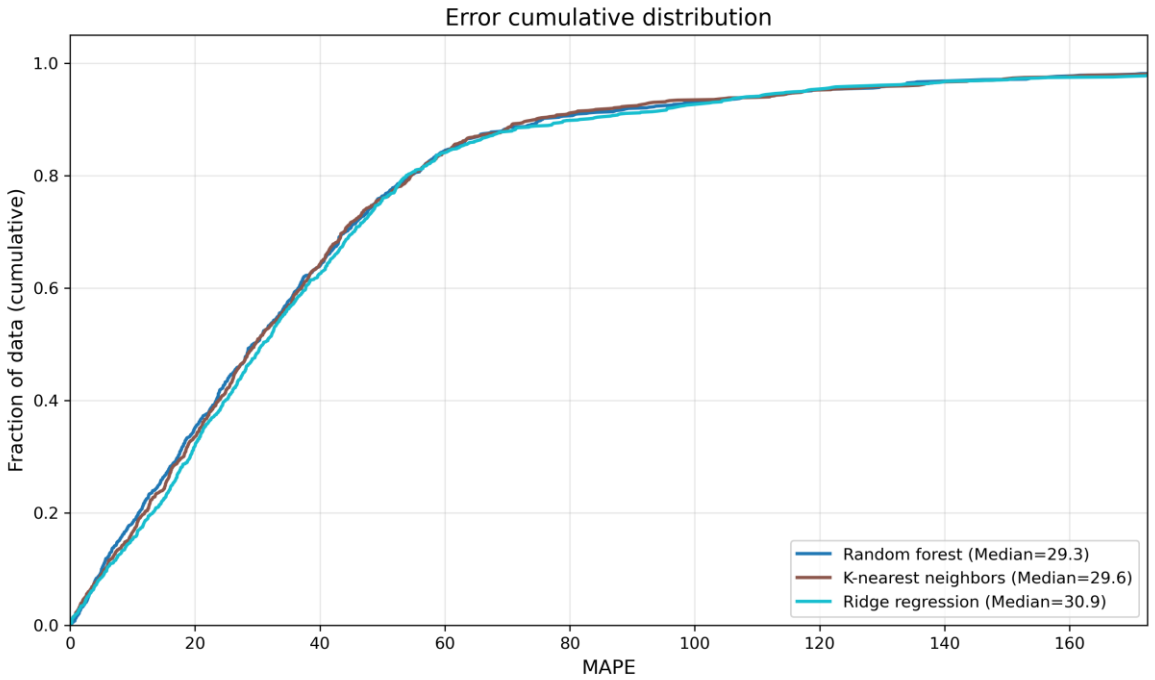
Appendix

- Hyperparameters estimated using grid search within nested cross validation:
  - Outer loop to calculate prediction metrics (5 folds) : In each iteration, one fold was held out as the test set to serve as unseen data, while the remaining four folds constituted the training set.
  - Inner loop to identify the optimal hyperparameters (5 folds) : Within the training set, a secondary 5- fold cross-validation was performed to identify the optimal hyperparameters within the predefined grid



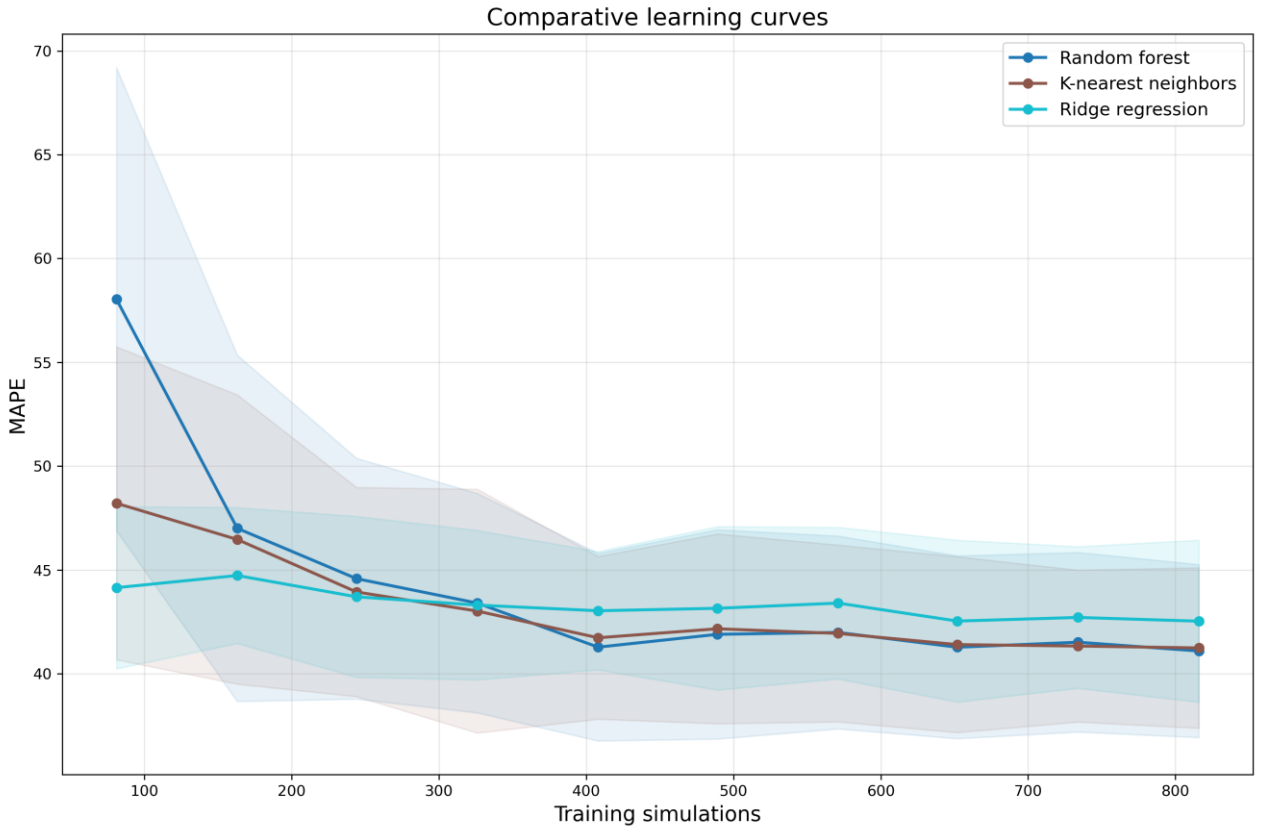
Appendix

- Skewness in MAPE observed in the cumulative distribution function of the errors.
- The curve rises steeply indicating a concentration of low errors, before flattening out due to the high-error tail.



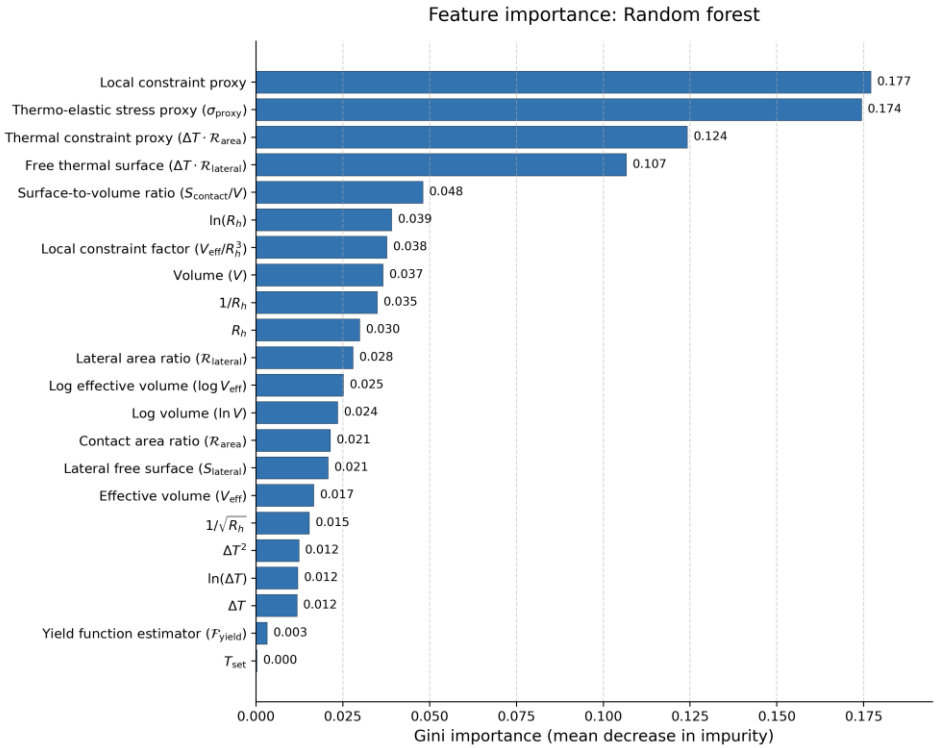
Appendix

- Learning curve showcasing saturation: adding more simulations would not substantially decrease the MAPE ⇒ Error not due to lack in data

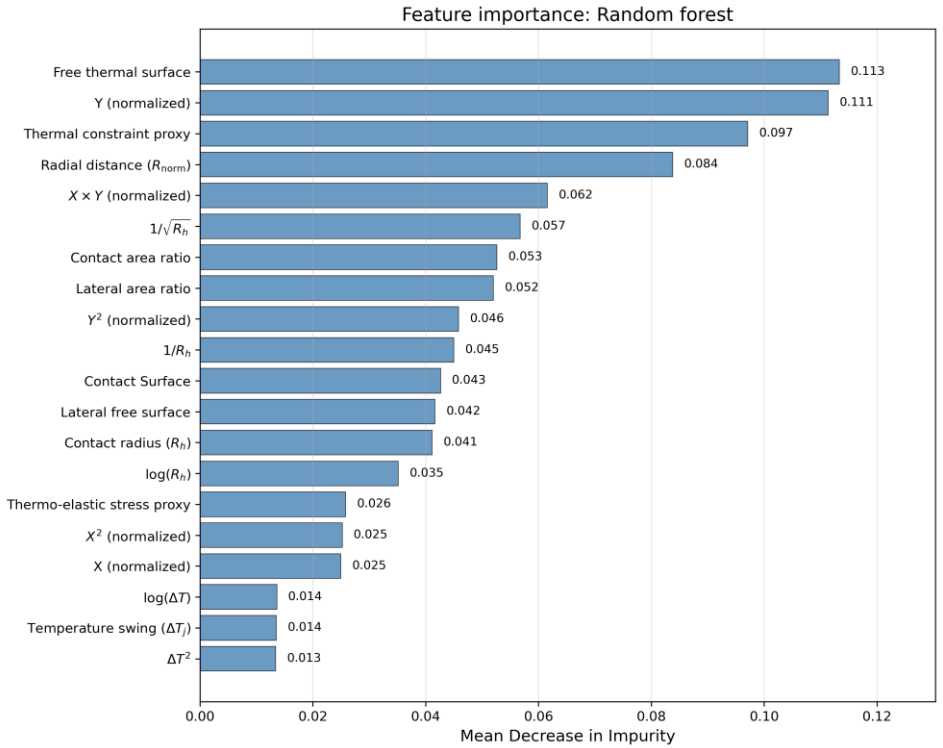


Appendix

- VWA predictions

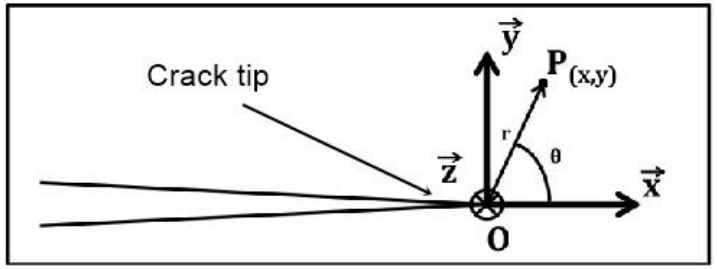


- Spatial predictions



Appendix

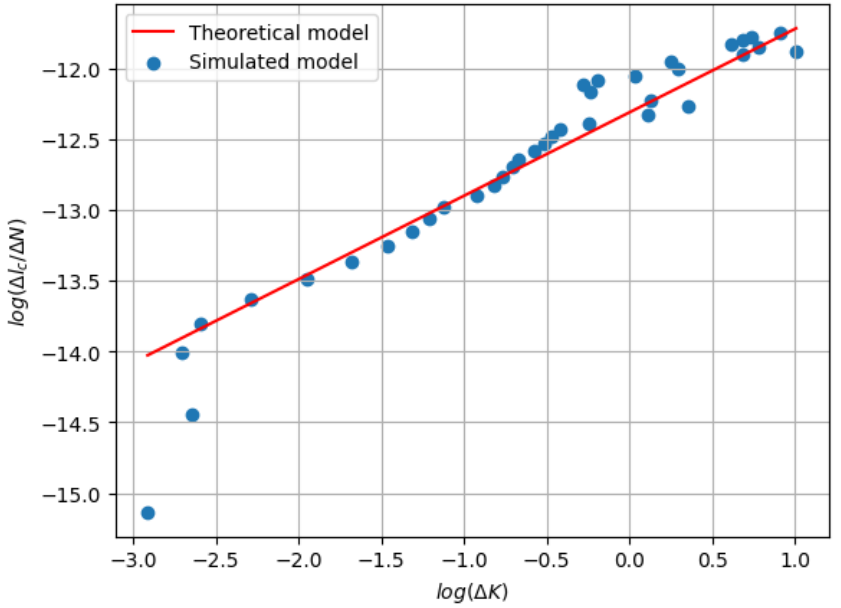
- Determination of Paris law's validity (Legacy simulations)



$$\sigma_{ij} = \frac{K}{\sqrt{2\pi r}} f_{ij}(r,\theta)$$

- $C = 4.499 \times 10^{-6}; m = 0.589$
- $\Delta K_1 = 0.075 MPa\, m^{\frac{1}{2}}$; $\Delta K_2 = 1.429 MPa\, m^{\frac{1}{2}}$

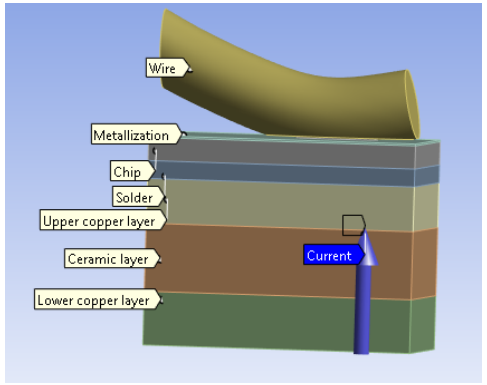
Specimen No.	C	m
Aluminium alloy (CAL combined)	$2.44E^{-07}$	3.06
CCT-SBL-AL-01	$1.28E^{-06}$	2.44
CCT-SBL-AL-02	$1.10E^{-06}$	2.52
Aluminium alloy (Constant R) – Major stress range test only	$2.83E^{-08}$	3.49

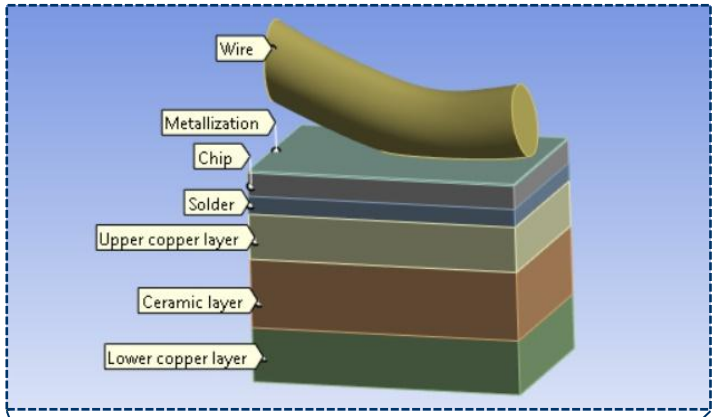


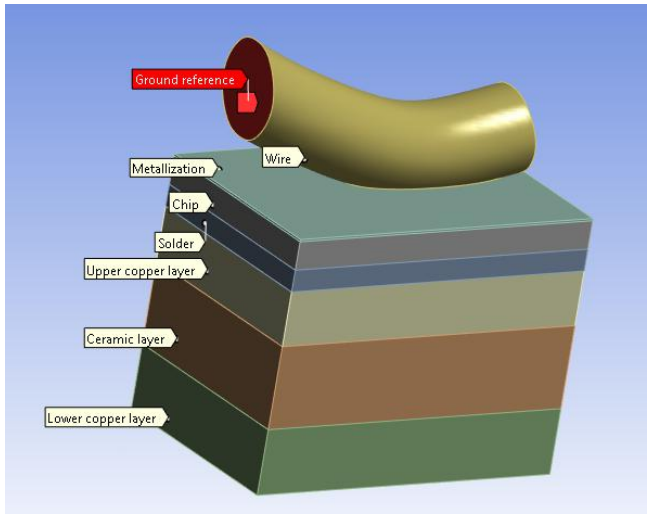
Appendix

A voltage $0V$ is imposed on the surface $\delta_W\Omega$ of the wire connecting the module to the rest of the circuit.

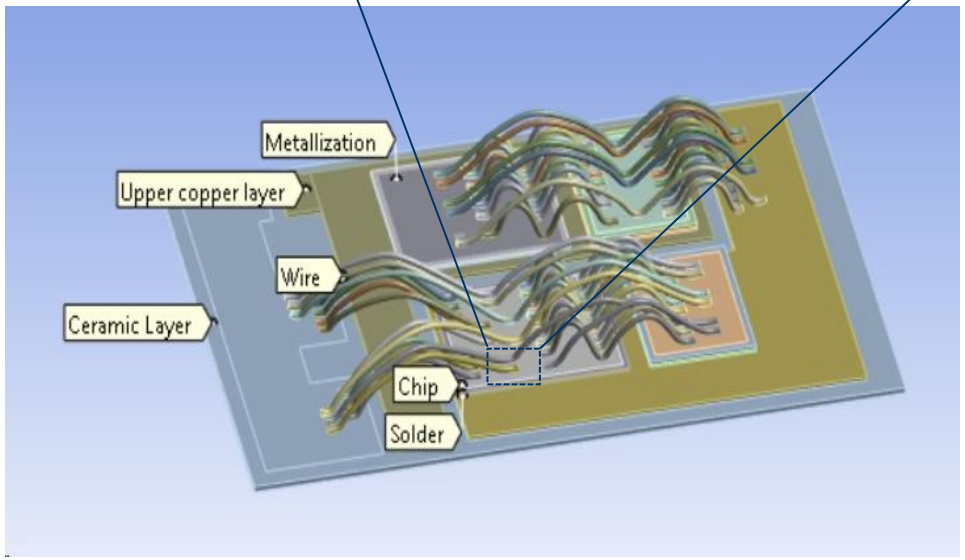
A current I is applied to the lower face of the upper copper layer Ω_{UCB}.





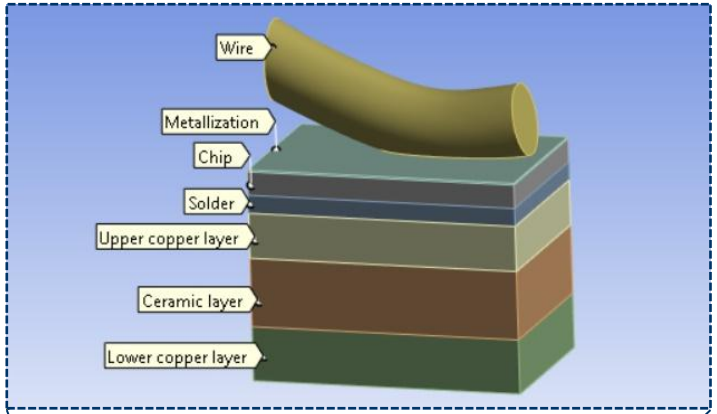


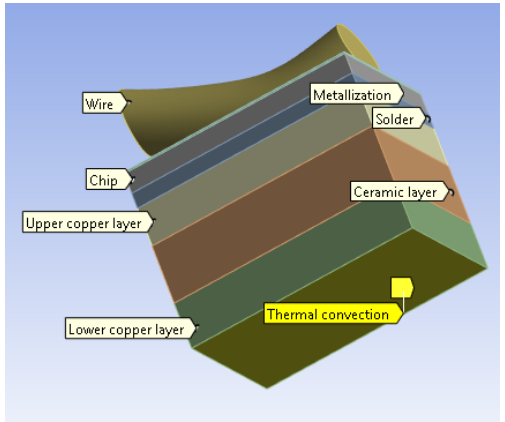
Physics	Electric
Unknowns	Potential V Current density j
Equilibrium	$div(\boldsymbol{j}) = 0$
Constitutive law	$\boldsymbol{j} + \sigma(T) \cdot \boldsymbol{grad}(V) = \boldsymbol{0}$
Boundary conditions	$V = 0$ on $\delta_w\Omega$ $\boldsymbol{j} \cdot \boldsymbol{z} = \frac{I_{ref}}{S_{UCB}}$ on $\delta_{UCB}\Omega$



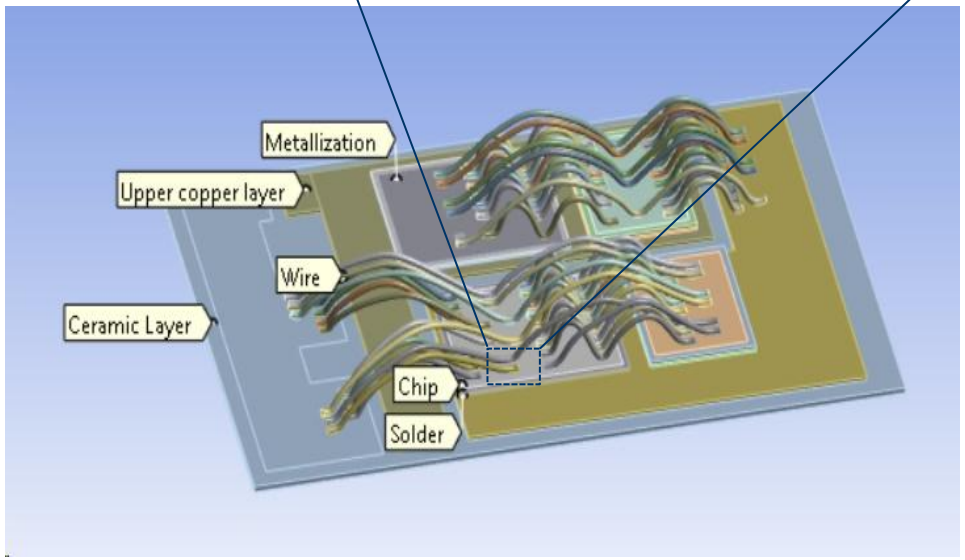
Appendix

Thermal boundary conditions on the bottom face of the lower copper layer $\delta_{LCB}\Omega$ to simulate cooling using a coefficient of thermal convection h and a cooling temperature $T_0 = 55\ °C$.





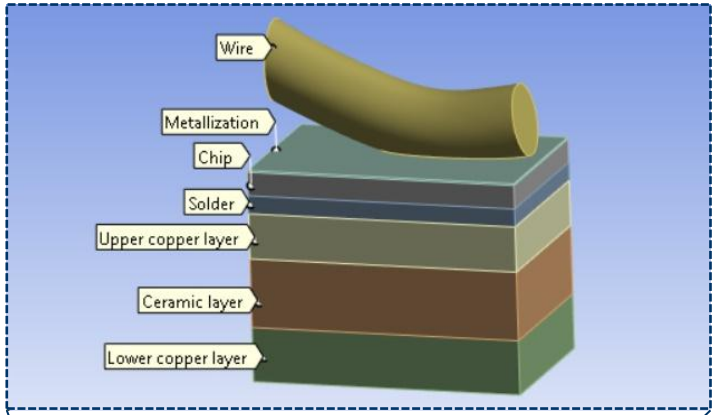
Physics	Thermal
Unknowns	Temperature T Heat flux q
Equilibrium	$-div(\boldsymbol{q}) + p_J = 0$ $p_J = \sigma(T)\|\|\boldsymbol{grad}(V)\|\|^2$
Constitutive law	$\boldsymbol{q} + k \cdot \boldsymbol{grad}(T) = 0$
Boundary conditions	$T = T_0$ on $\delta_{LCB}\Omega$ $\boldsymbol{q} \cdot \boldsymbol{z} + hT = 0$ on $\delta_{LCB}\Omega$

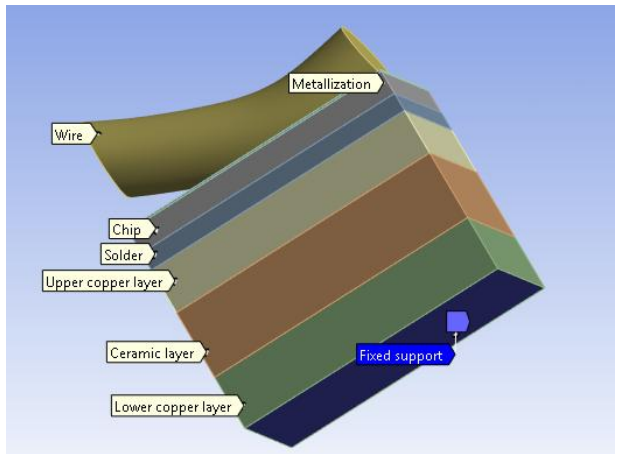


Appendix

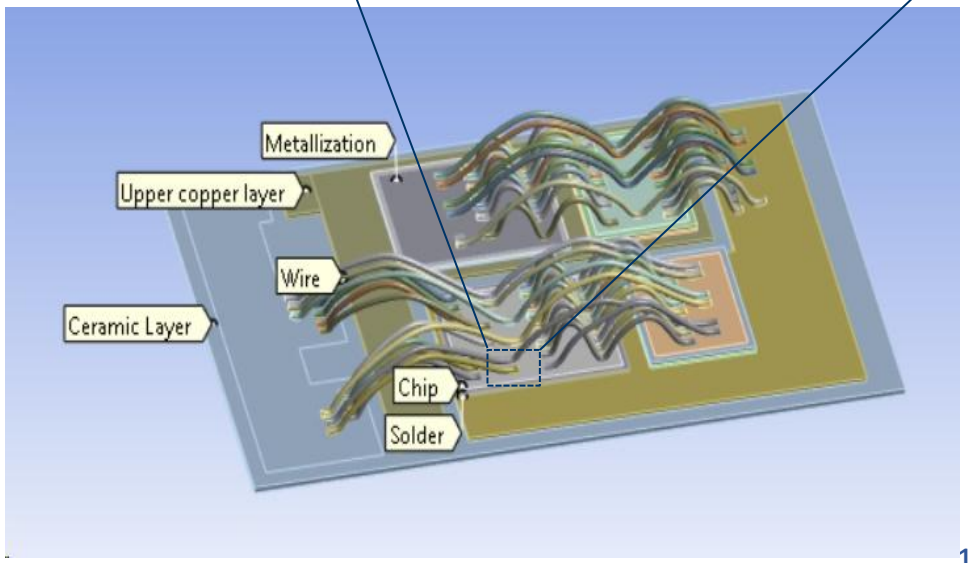
A Fixed support boundary condition is imposed on the bottom face of the lower copper layer$\delta_{LCB}\Omega$ as the module's basis du module de puissance.

Plasticity is modeled in copper Ω_C and aluminum Ω_A using Chaboche kinematic hardening (Dong 2009) and bilinear kinematic hardening (Rajaguru 2014), respectively.



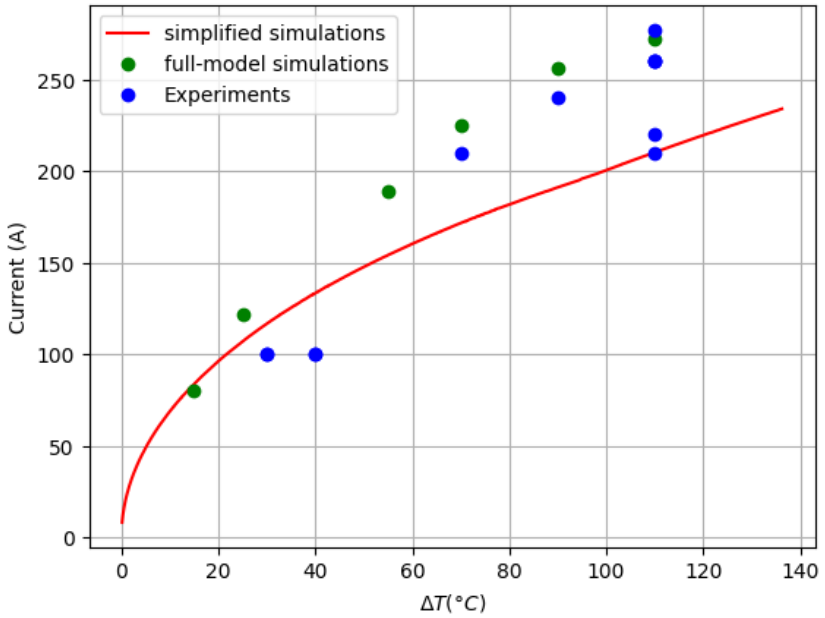


Physics	Mechanical
Unknowns	Displacement u Stress $\underline{\sigma}$ Strain $\underline{\varepsilon}$
Equilibrium	$\boldsymbol{div}\left(\underline{\sigma}\right) = 0$
Constitutive law	$\underline{\sigma} = \mathcal{K}\colon(\underline{\underline{\varepsilon^e}}(\boldsymbol{u}))$ in Ω_L $\underline{\sigma} = f_A(\underline{\dot{\varepsilon}})$ in Ω_A $\underline{\sigma} = f_C(\underline{\dot{\varepsilon}})$ in Ω_C $\underline{\underline{\varepsilon^e}}(\boldsymbol{u}) = grad_s(\boldsymbol{u}) - \alpha_{th}(T - T_0)\underline{I_d} - \underline{\underline{\varepsilon^p}}(\boldsymbol{u})$
Boundary conditions	$\boldsymbol{u} = \boldsymbol{0}$ on $\delta_{LCB}\Omega$ $\underline{\sigma} \cdot \boldsymbol{n} = \boldsymbol{0}$ on $\delta\Omega \backslash \delta_{LCB}\Omega$

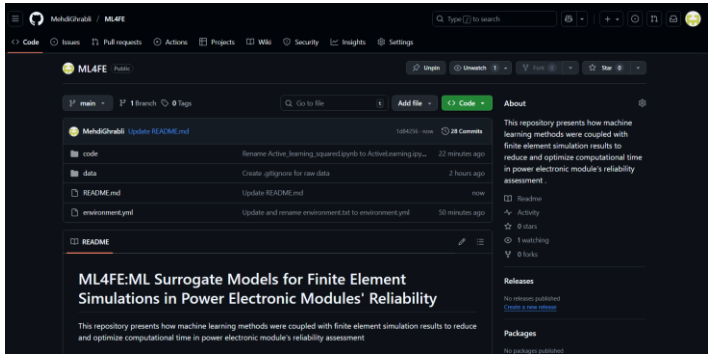


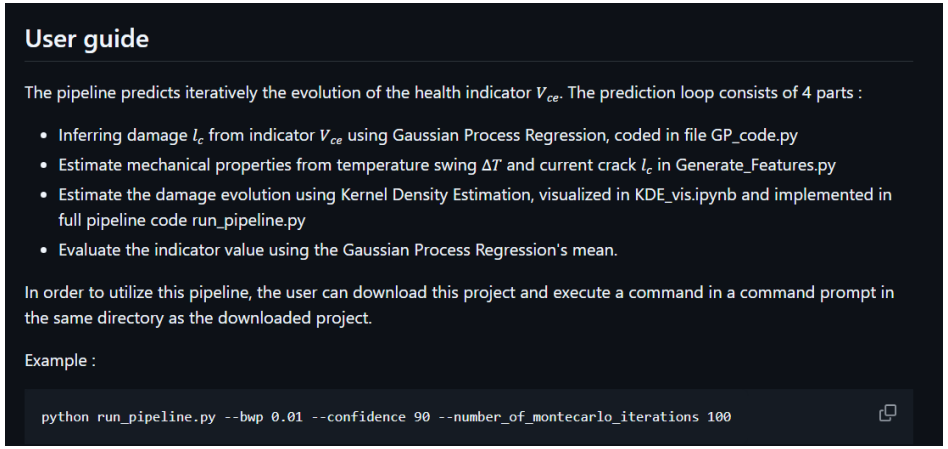
Appendix

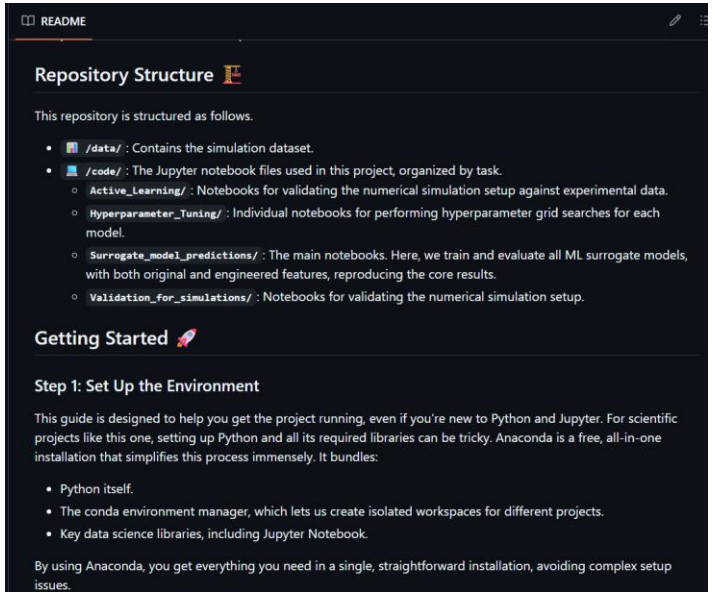
- The simulation current I_{sim} was chosen to generate a predefined range of temperature variations ($\left[I_{sim}^{(1)}, I_{sim}^{(2)}\right] \Leftrightarrow [\Delta T^{(1)}, \Delta T^{(2)}]$). This approach ensures the resulting data effectively spans the necessary thermal conditions for training our surrogate models.
- Due to the model's reduced geometry, a direct current comparison is misleading. We established an equivalent full-model current $I_{eq} = \alpha I_{sim}$ by scaling simulation current with the geometric reduction factor α.
- On average, our equivalent current I_{eq} is slightly lower than reference values. This is an expected outcome attributed to two main factors:
  - **Boundary Conditions:** A higher cooling base temperature was used in our simulations, which requires less self-heating (and thus lower current) to achieve the same ΔT.
  - **Geometric range:** The model focuses exclusively on the primary hot-spot (wire-metallization contact), neglecting cooler peripheral regions. This concentrates heat, reducing the power required to reach the target temperature compared to a full model where heat dissipates over a larger area.



Appendix







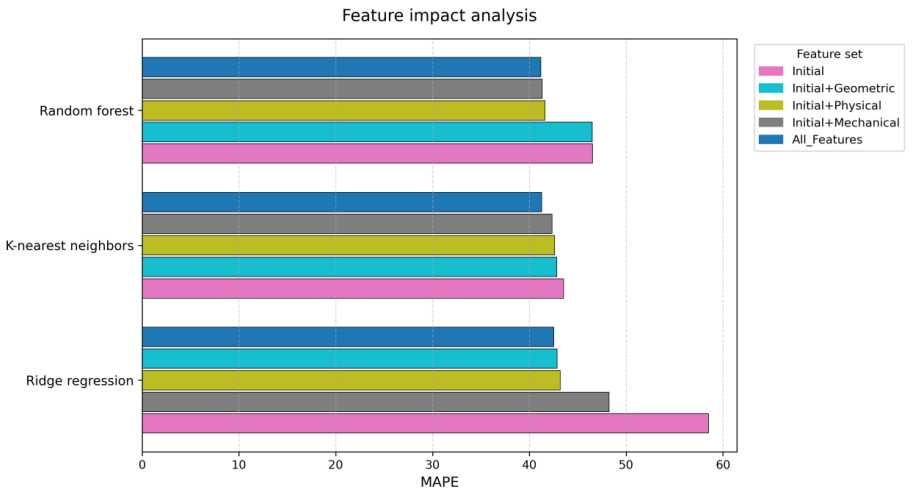
- Code + simulation data open-sourced via GitHub repositories

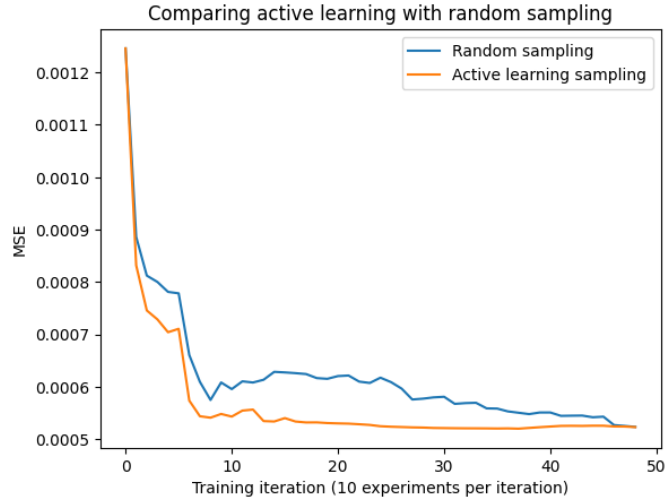
ML surrogate models: github.com/MehdiGhrabli/ML4FE

RUL estimation model: github.com/MehdiGhrabli/PIMC4RUL

Appendix

- For efficient application of active learning, ensemble models should be competitive, otherwise aggregation would have a detrimental effect:
  - The best performing model is treated the same as the worst performing model
  - Active learning may converge to an error higher than the best performing model
  - Poor models contribute negatively and hinder sampling
- Further validation through cross validation (“coarse” error plots), computing costs, comparing schemes (Sobol, LHS, etc…), Hyperparameter study (optimization, effect, etc…)
- Active learning comes with a hidden cost of repeated forward calls





Appendix

Lifecycle reliability & maintenance strategies

Corrective maintenance

- Replace component only after failure
  - Maximizes lifetime
  - Costs due to unexpected downtime
  - Potential cascading effects leading to expensive system failure

Predictive maintenance

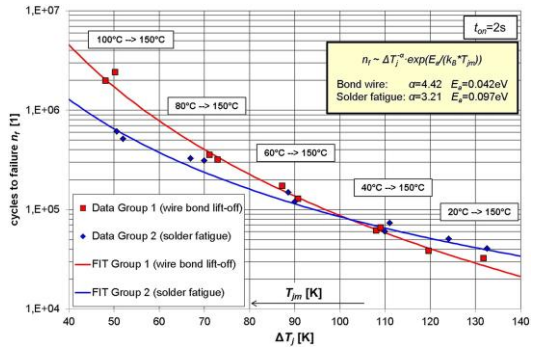
- Replace component directly before failure
  - Downtime planning
  - No cascading effects
  - Efficient
  - Requires accurate lifetime modeling

Preventive maintenance

- Replace component periodically prior to failure
  - Downtime planning
  - No cascading effects
  - Wasteful: inefficient workload and material exploitation

Empirical models

- Temperature based Coffin-Manson
  - $N_f = A(\Delta T)^B$
- Bayerer model
  - $N_f = K \times \Delta T^{\beta_1} \times e^{\frac{\beta_2}{T+273}} \times t_{on}^{\beta_3} \dots$
- Simple
- Poor out-of-distribution (OOD) prediction due to calibration issues



Physics-of-Failure

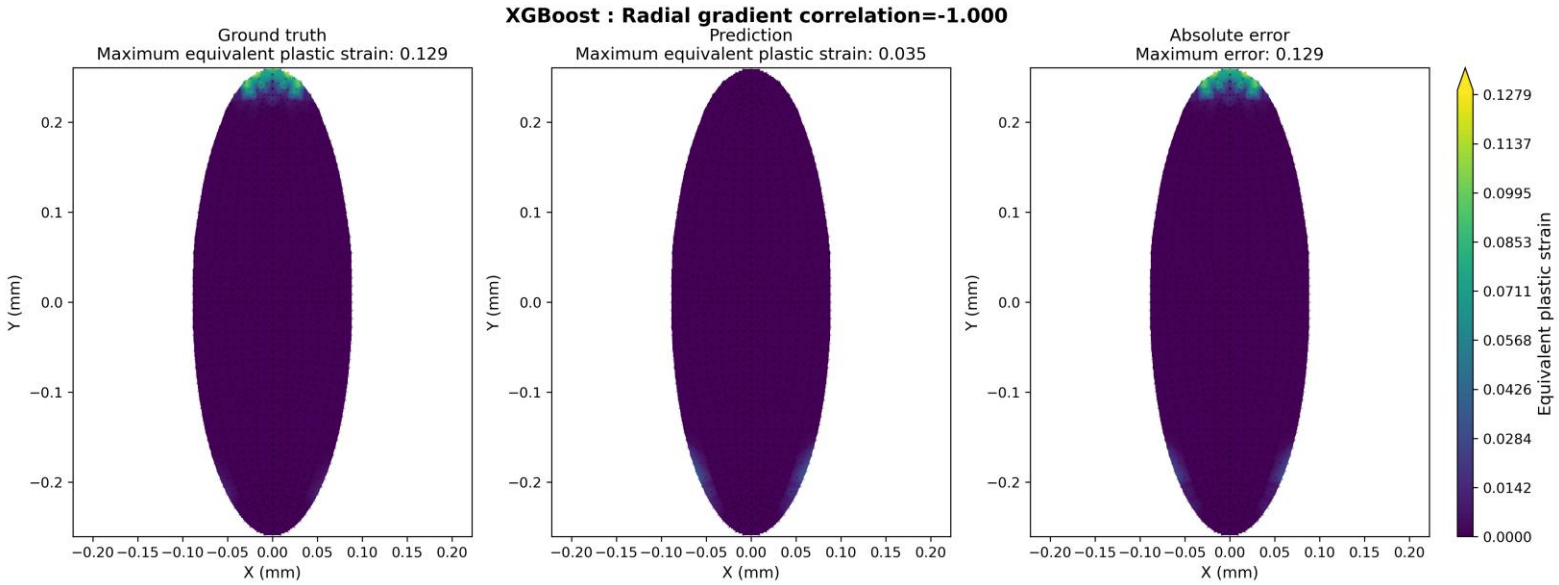
- Coffin-Manson
  - $\Delta\varepsilon = \frac{\sigma_f'}{E}(2N_f)^b + \varepsilon_f'(2N_f)^c$
  - Physics-informed
  - Static indicator measurement
- Paris' law
  - $\frac{da}{dN} = C(\Delta K)^m$
  - Dynamic
  - ΔK intractable
- Modified Paris' law
  - $\frac{da}{dN} = C(\Delta\varepsilon)^m$
  - $\Delta\varepsilon$ calculated through numerical simulations
  - Impractical (computational costs due to frequent updates)

Data-driven models

- Deep learning architectures
  - LSTMs + CNN
- Stochastic models
- Poor OOD predictions
- Hight data dependence
- PINNs
- Better convergence
- Limited to shape-based constraints
- Foundation models remain unexplored

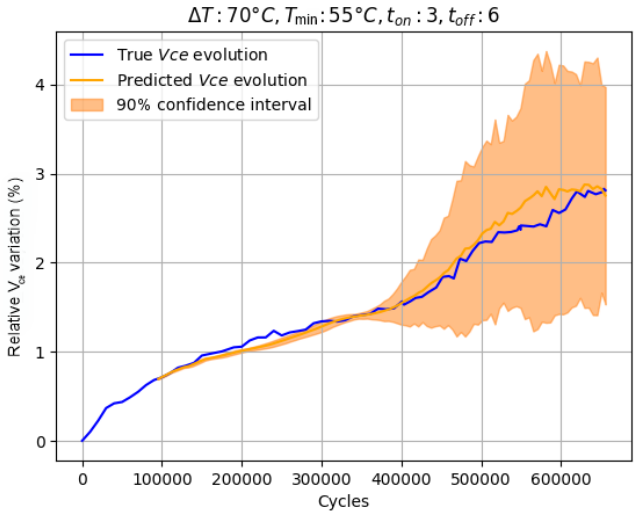
Appendix

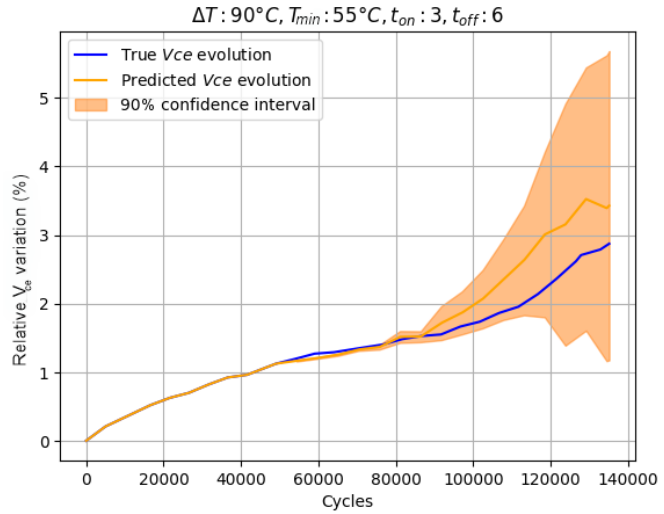
- Outlier behavior reported in the simulation divergence region $(R_h \approx 0{,}26mm)$ characterized by an inversion of the plastic strain's distribution around the center leading to an increase in prediction error.

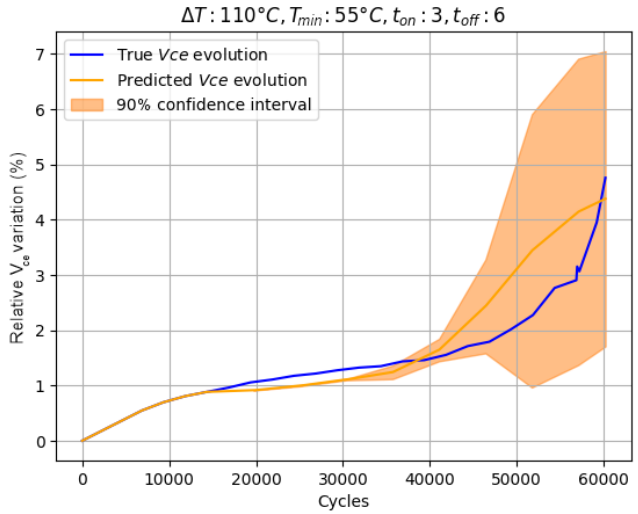


Appendix

- In-distribution prediction results showcasing consistent behavior: tight confidence intervals across the life of the module until reaching a high variance regime near failure



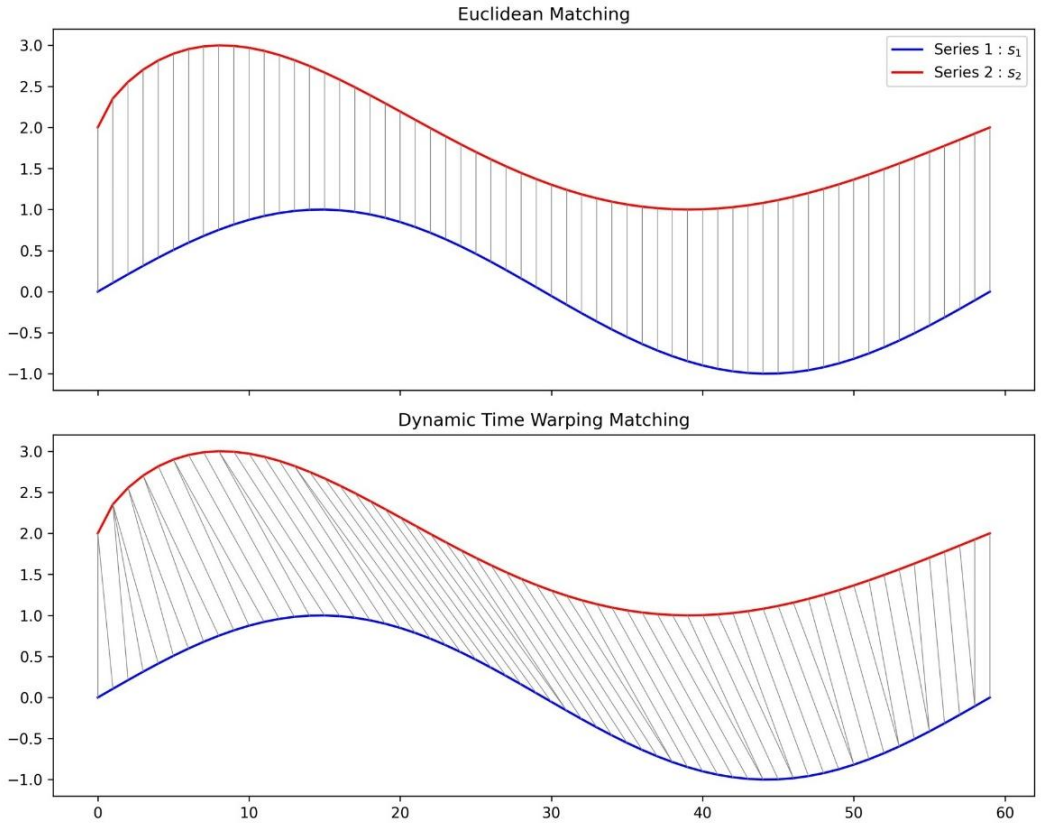




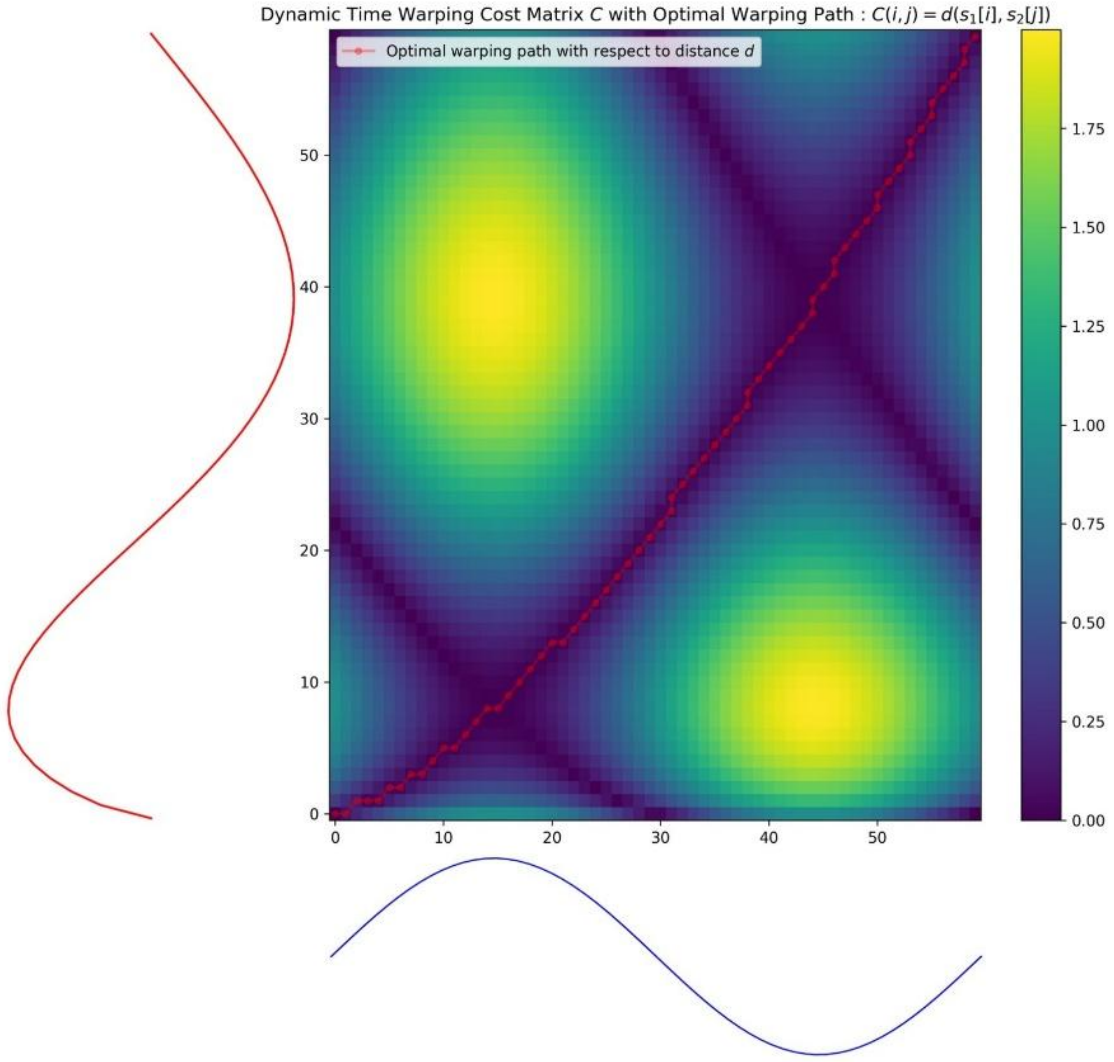
Test ΔT (°C)	Train ΔT (°C)	Coverage @90	Strict coverage @90	Bandwidth
70	{70,90,110}	0,702	0,637	0,01
90	{70,90,110}	0,802	0,802	0,01
110	{70,90,110}	0,509	0,509	0,01

Appendix

- Dynamic warping (DTW): a time series matching algorithm robust to time warping (compression and stretching)
- DTW aligns peaks and troughs appropriately despite temporal offset (shortcomings of Euclidean matching)



Appendix



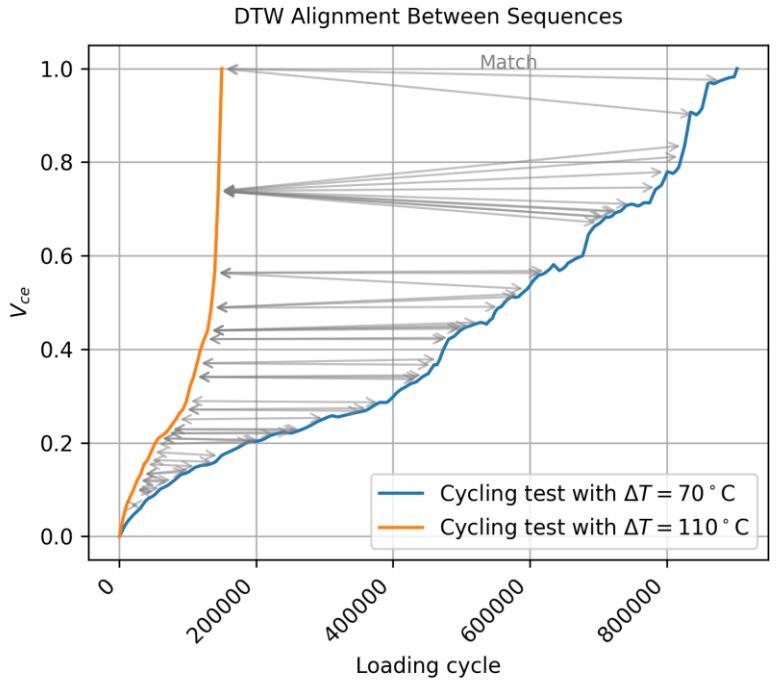
- DTW defines a cost matrix $\mathcal{C}$ using the signal values and a distance measure:

$$\mathcal{C}[i,j] = d(s_1[i], s_2[j])$$

- Matching is determined through the warping path: path of the lowest cost in the matrix $\mathcal{C}$ calculated using dynamic programming
- Holds information on the relative speed between both signals

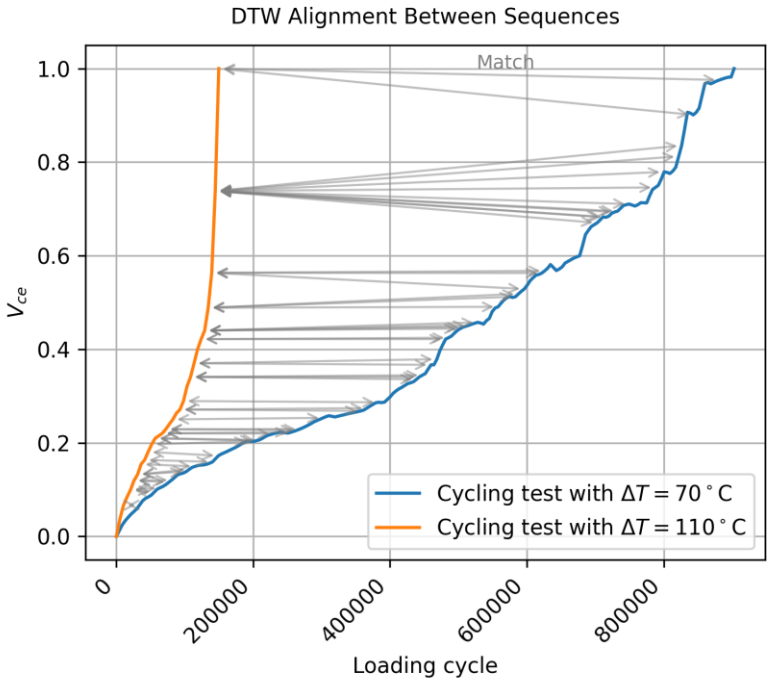
Appendix

- Warping effect seen in health indicator data as high temperature loading causes early failure (compression) compared to slower low temperature tests (stretching)
- Health indicator curves share the same characteristic shape of steady evolution followed by a sharp increase signaling failure



Appendix

- Use dynamic time warping to compute the relative speed between a fast reference (High ΔT, low t_{on}) and a slow incomplete test (Low ΔT, high t_{on})
  - ⇒ Information represented in the warping path
- Forecast the warping path in the future using time series prediction models
- Use the predicted warping path and the reference signal to complete the slow test ("Dynamic Time Unwarping")



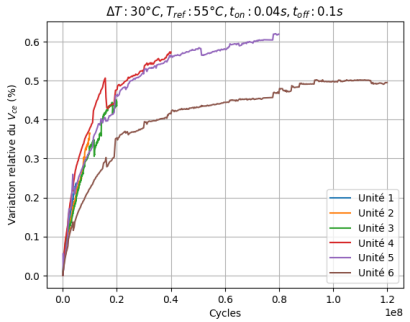
Appendix

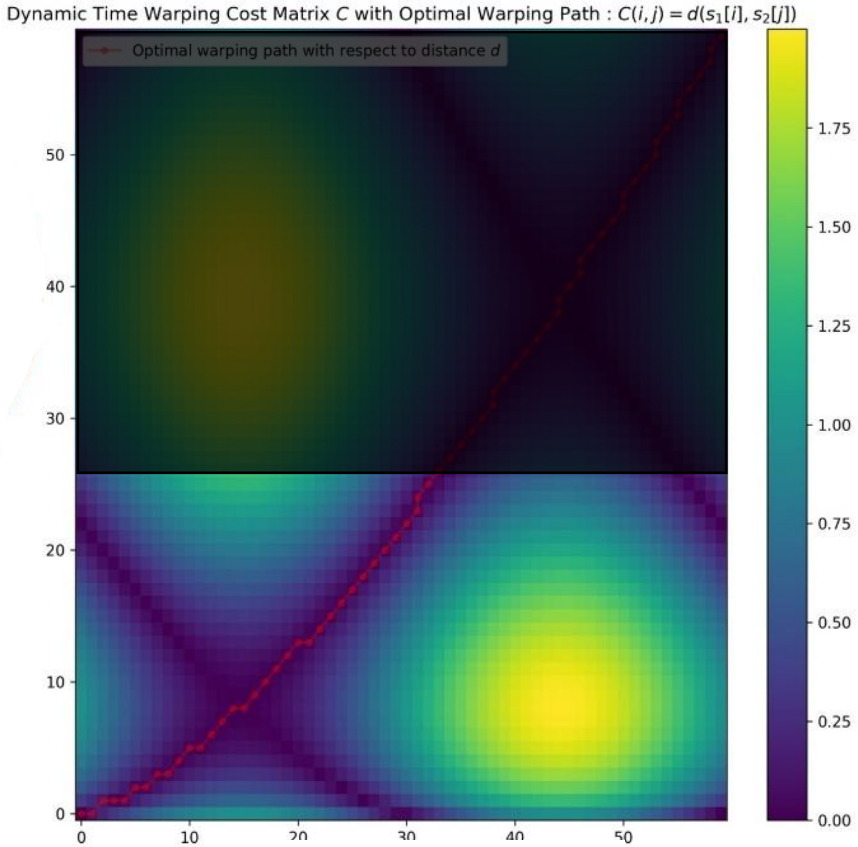
- Partial observation

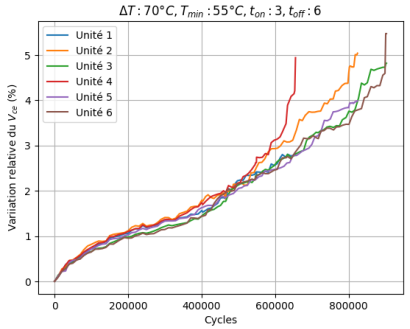
  of the signal ⇔

  Partial observation

  of the cost matrix



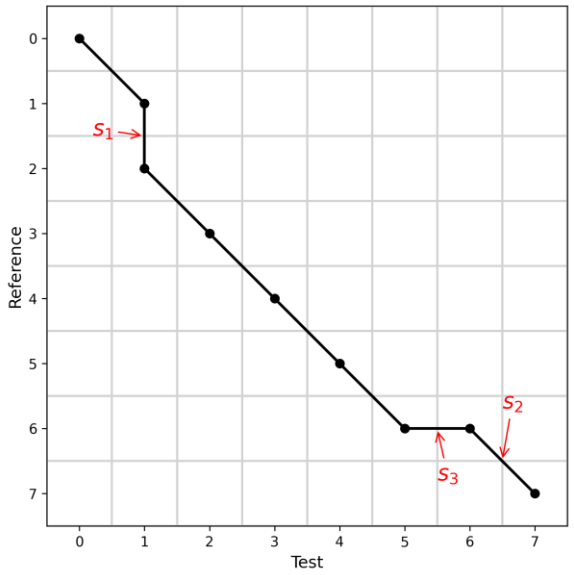


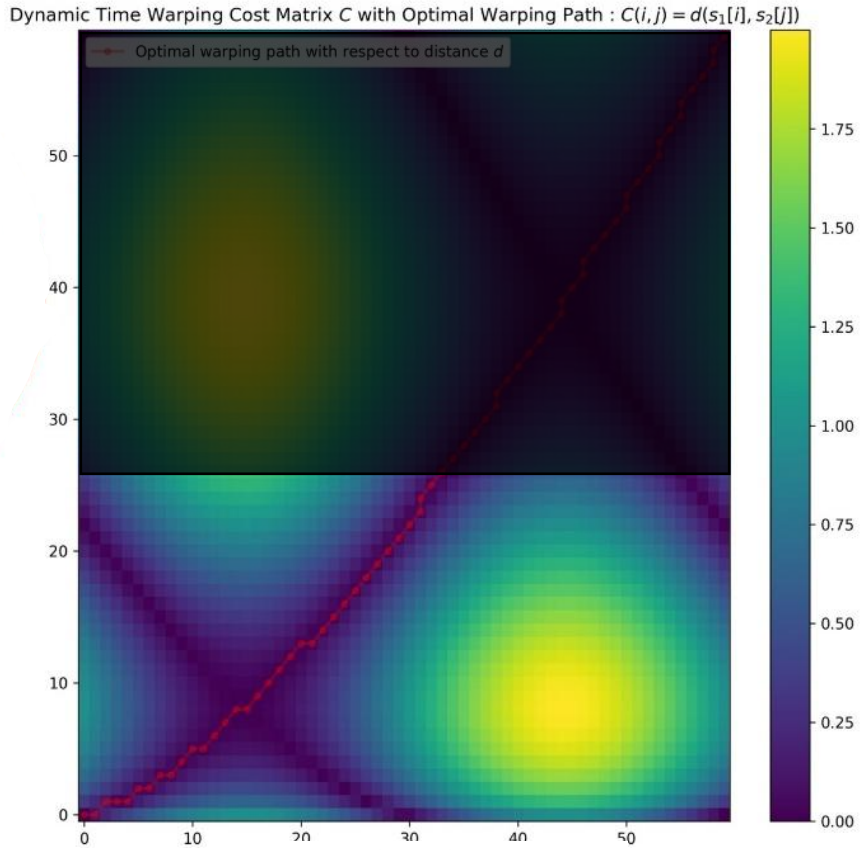


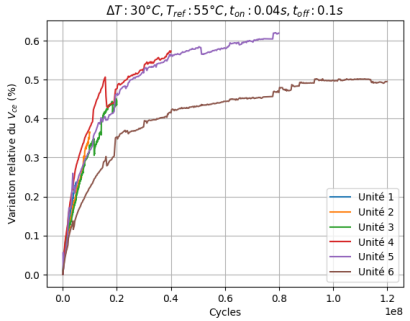
- Forecasting the warping path in the hidden area ⇔ **B**lindfolded **E**xploration using **D**ynamic **time** warping
- The warping path is a realization of a Random Walk (Time in-homogenous) with 3 states

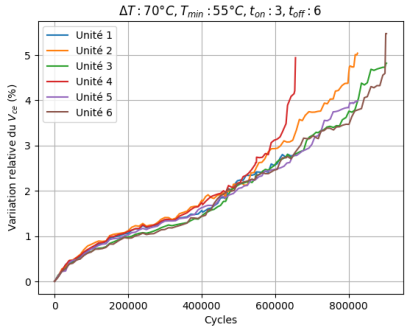
$$M(t) = \begin{bmatrix} & \text{Up} & \text{Diagonal} & \text{Down} \\ \text{Up} & \phi_{0,0}(t) & \phi_{0,1}(t) & \phi_{0,2}(t) \\ \text{Diagonal} & \phi_{1,0}(t) & \phi_{1,1}(t) & \phi_{1,2}(t) \\ \text{Down} & \phi_{2,0}(t) & \phi_{2,1}(t) & \phi_{2,2}(t) \end{bmatrix}$$

Appendix





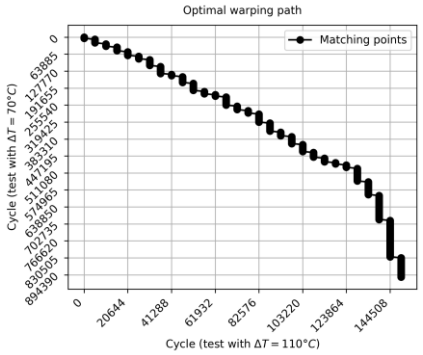


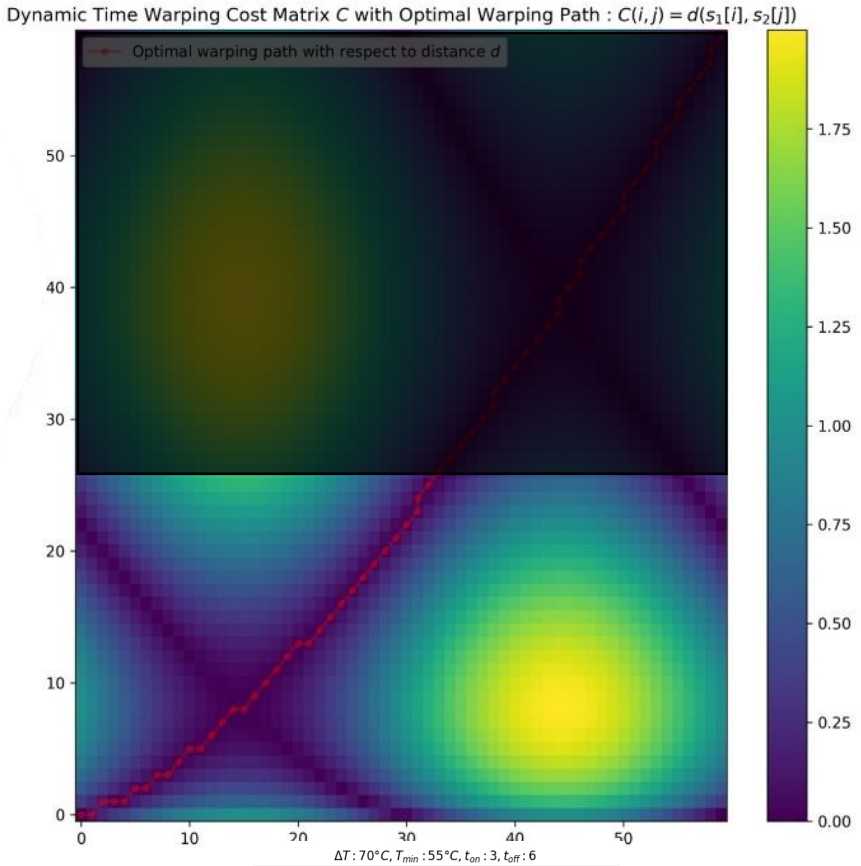


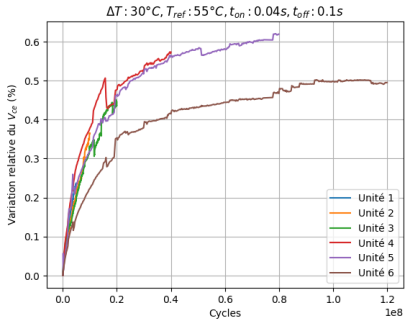
- Forecasting the warping path in the hidden area ⇔ **B**lindfolded **E**xploration using **D**ynamic **time** warping
- The warping path is a realization of a Random Walk (Time in-homogenous) with 3 states

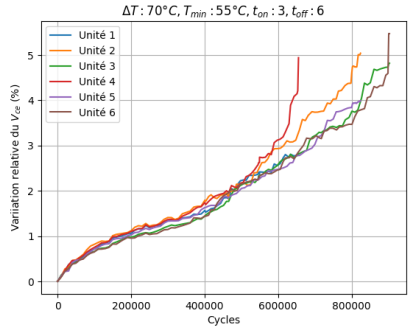
$$M(t) = \begin{bmatrix} & \text{Up} & \text{Diagonal} & \text{Down} \\ \text{Up} & \phi_{0,0}(t) & \phi_{0,1}(t) & \phi_{0,2}(t) \\ \text{Diagonal} & \phi_{1,0}(t) & \phi_{1,1}(t) & \phi_{1,2}(t) \\ \text{Down} & \phi_{2,0}(t) & \phi_{2,1}(t) & \phi_{2,2}(t) \end{bmatrix}$$

Appendix





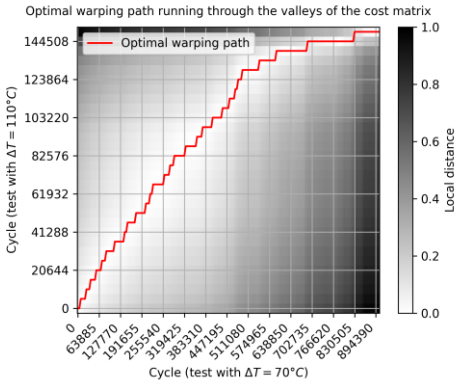


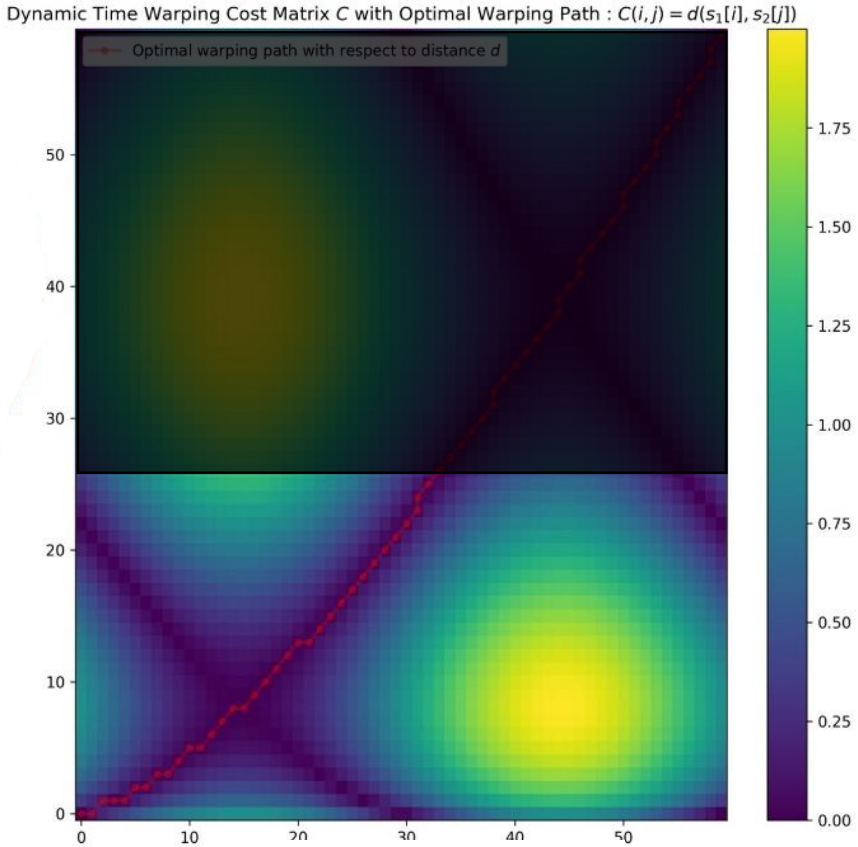


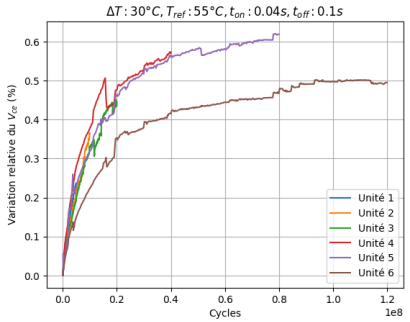
- Forecasting the warping path in the hidden area ⇔ **B**lindfolded **E**xploration using **D**ynamic **time** warping
- The warping path is a realization of a Random Walk (Time in-homogenous) with 3 states

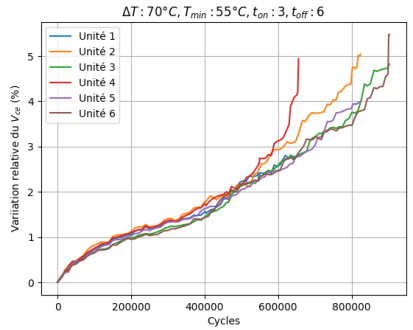
$$M(t) = \left[\begin{array}{c|ccc} & \text{Up} & \text{Diagonal} & \text{Down} \\ \hline \text{Up} & \phi_{0,0}(t) & \phi_{0,1}(t) & \phi_{0,2}(t) \\ \text{Diagonal} & \phi_{1,0}(t) & \phi_{1,1}(t) & \phi_{1,2}(t) \\ \text{Down} & \phi_{2,0}(t) & \phi_{2,1}(t) & \phi_{2,2}(t) \end{array}\right]$$

Appendix





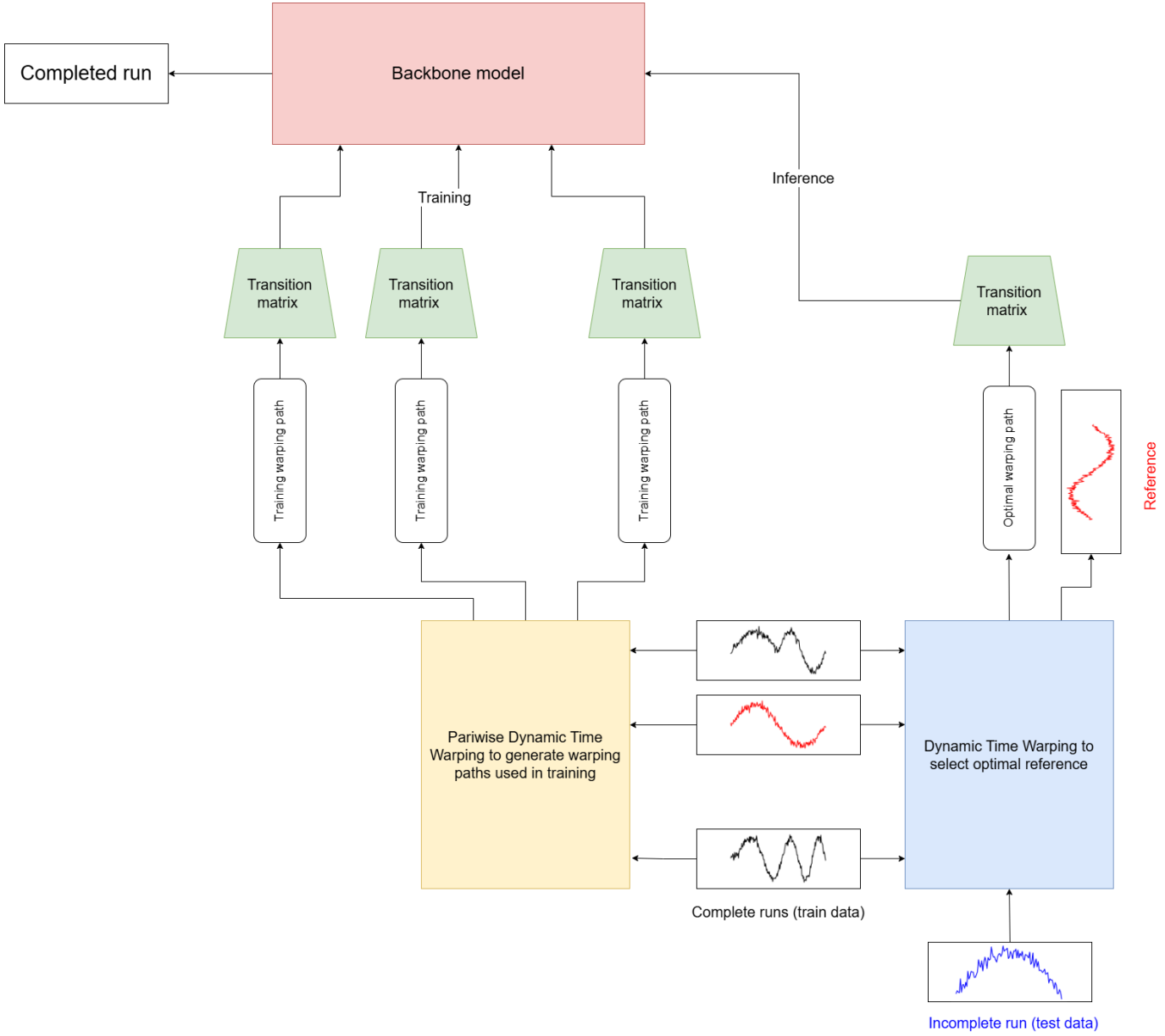




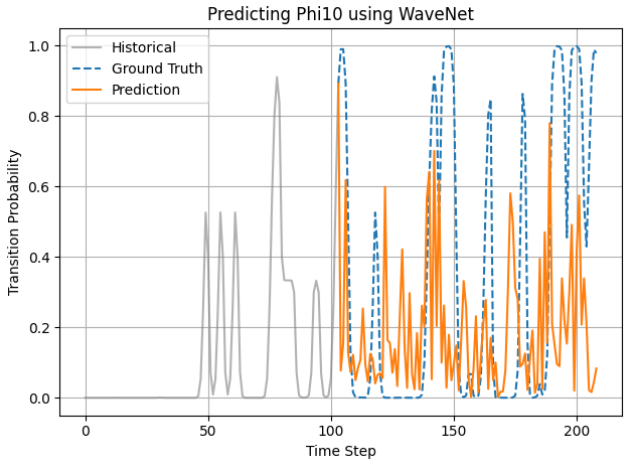
- Forecasting the warping path in the hidden area ⇔ **B**lindfolded **E**xploration using **D**ynamic **time** warping
- The warping path is a realization of a Random Walk (Time in-homogenous) with 3 states

$$M(t) = \left[\begin{array}{c|ccc} & \text{Up} & \text{Diagonal} & \text{Down} \\ \hline \text{Up} & \phi_{0,0}(t) & \phi_{0,1}(t) & \phi_{0,2}(t) \\ \text{Diagonal} & \phi_{1,0}(t) & \phi_{1,1}(t) & \phi_{1,2}(t) \\ \text{Down} & \phi_{2,0}(t) & \phi_{2,1}(t) & \phi_{2,2}(t) \end{array}\right]$$

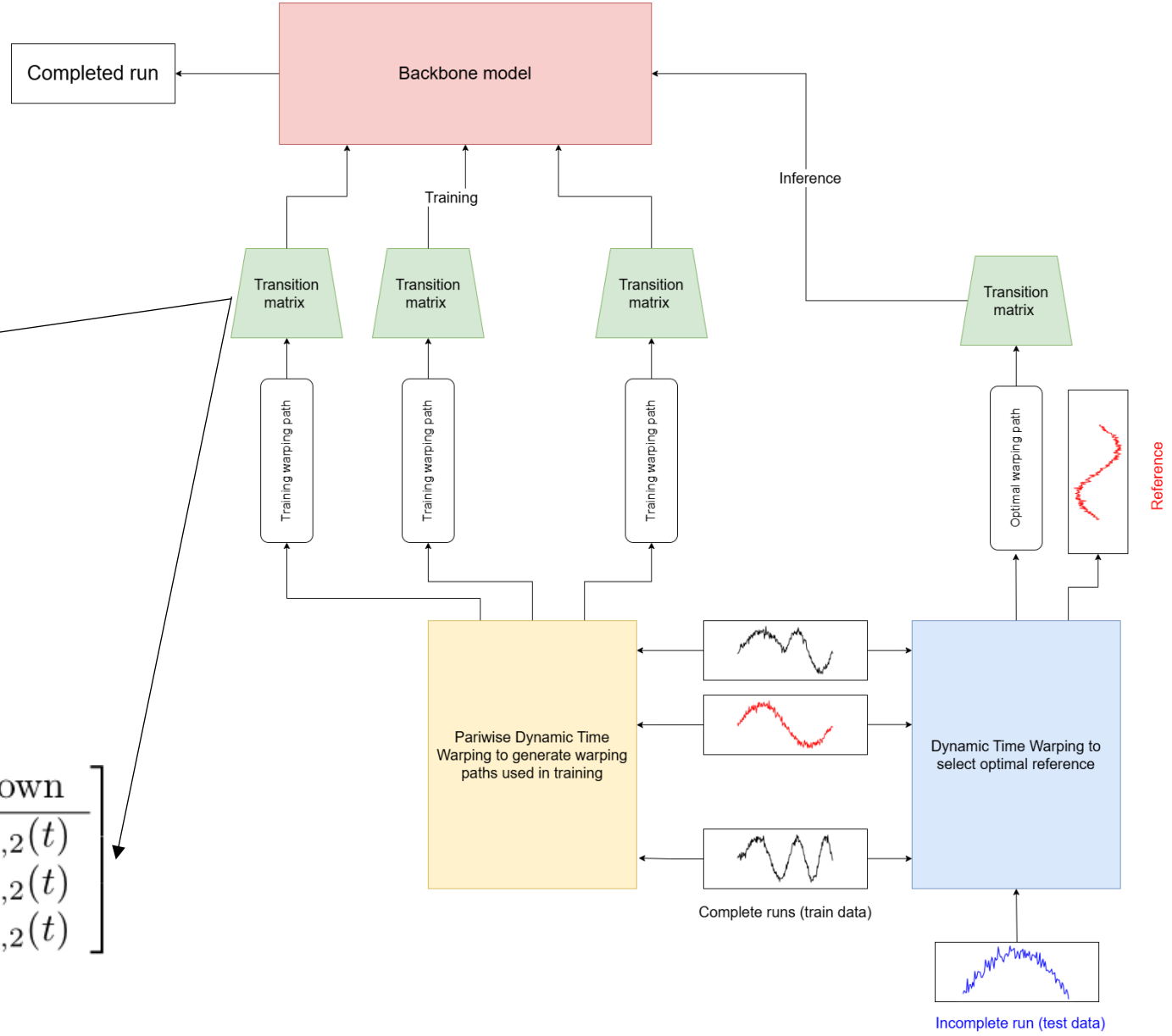
Appendix


Completed run
Backbone model
Inference
Training
Transition matrix
Transition matrix
Transition matrix
Transition matrix
Training warping path
Training warping path
Training warping path
Optimal warping path
Reference
Pariwise Dynamic Time Warping to generate warping paths used in training
Dynamic Time Warping to select optimal reference
Complete runs (train data)
Incomplete run (test data)

Appendix



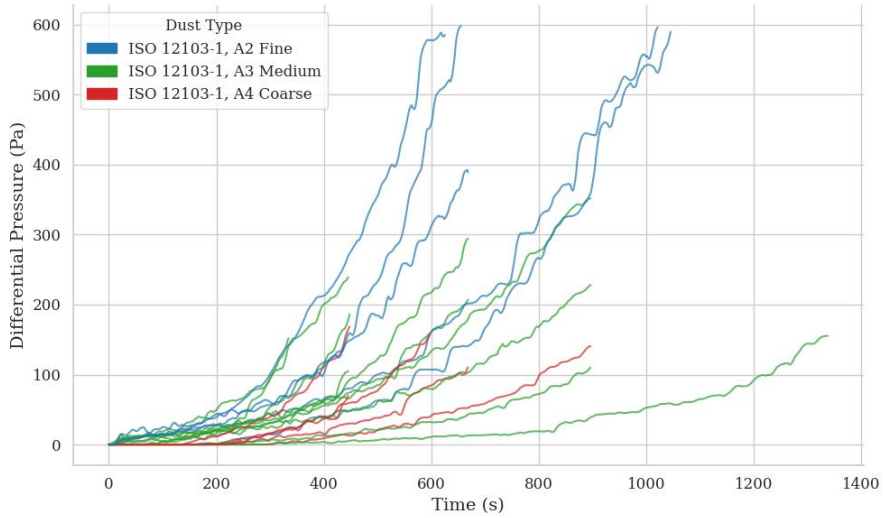
$$M(t) = \begin{bmatrix} & \text{Up} & \text{Diagonal} & \text{Down} \\ \text{Up} & \phi_{0,0}(t) & \phi_{0,1}(t) & \phi_{0,2}(t) \\ \text{Diagonal} & \phi_{1,0}(t) & \phi_{1,1}(t) & \phi_{1,2}(t) \\ \text{Down} & \phi_{2,0}(t) & \phi_{2,1}(t) & \phi_{2,2}(t) \end{bmatrix}$$



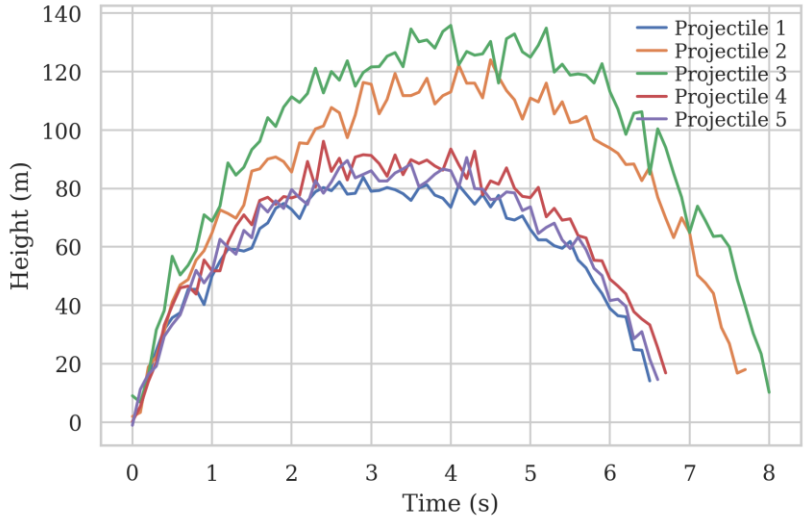
Appendix

Example of data showcasing warping effects

- Filter dataset representing the evolution of filter clogging through observation of differential pressure (Mauthe 2021) (50 total runs)



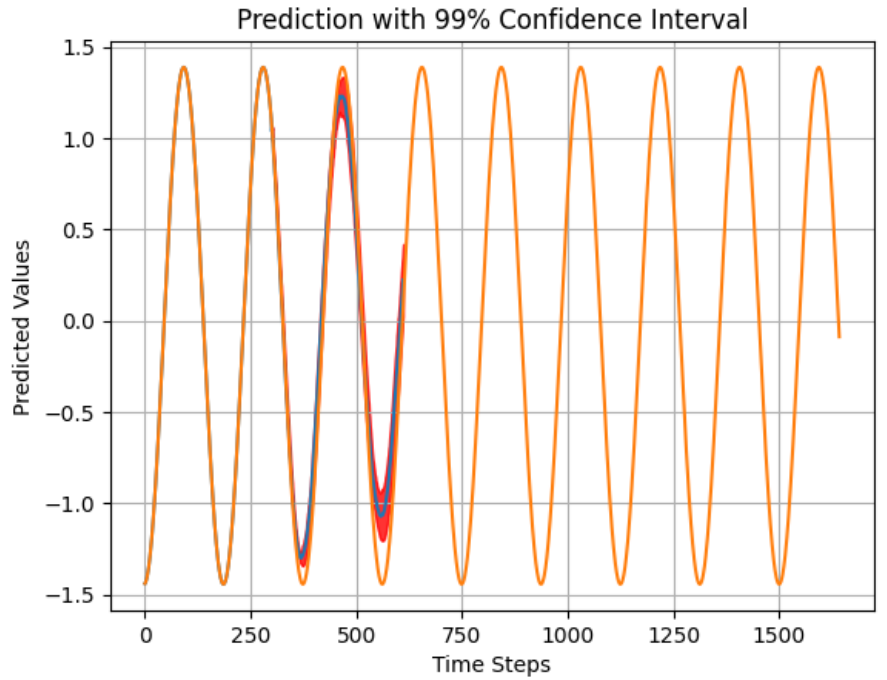
- Synthetic dataset (GDS Education Community of Practice, 2023) of 101 simulations representing the parabolic movement of launching different projectiles with different properties (mass, drag, etc…)



Appendix

- Taking the mean across all samples gives an estimate of the expected value at each future time step.
- Samples may present relative shifts due to different warpings

  ⇒ the extremal values in the forecast tend to diminish in magnitude.

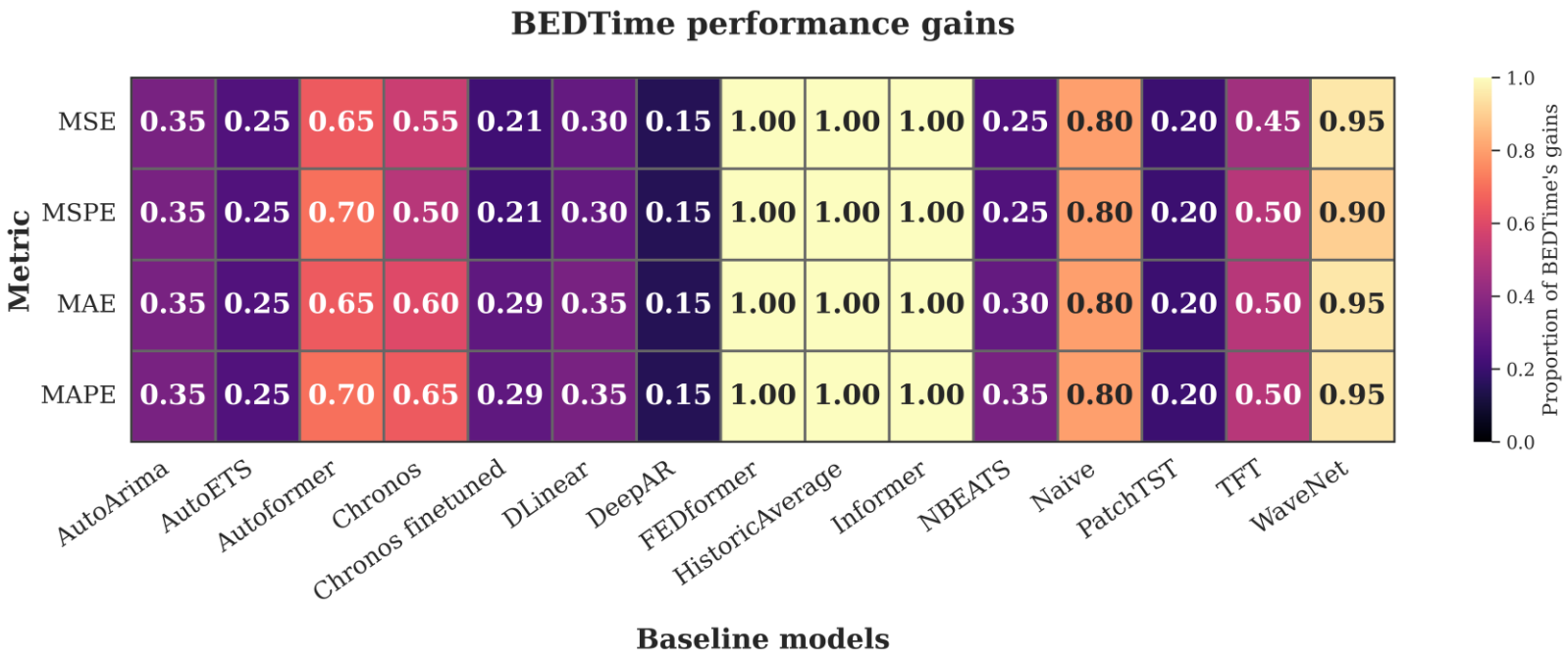


- This results from averaging with temporally shifted values near the extrema and this softening becomes more apparent when the prediction horizon is large and when the time series presents high fluctuation
- ⇒ Median-based aggregation (using DTW distance) is robust to extrema softening

Appendix

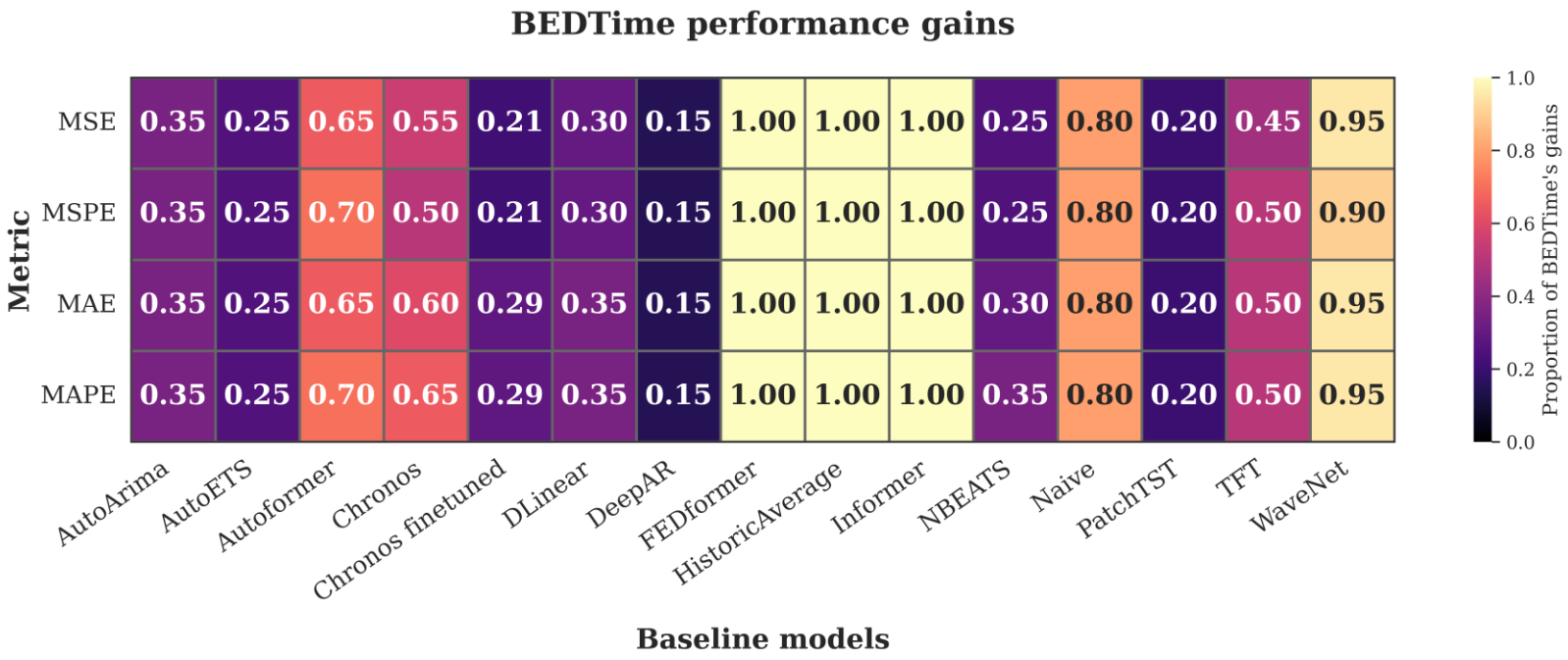
- The choice of the experimental setup was based on a trade-off between data quality, computational
- time, and applicability.
- Using 40/60 and 60/40 observation/prediction splits is useful to test the models' capabilities of forecasting on medium and long horizons, which is the main goal of forecasting in physical applications.
- Using training sizes of 1 and 6 allows testing in one-shot and few-shot settings.
- A training size of 6 was used due to the scarcity of meaningful data, especially in the Filter dataset (only 6 runs where the differential pressure exceeds 500Pa).
- These choices were applied to all datasets to keep a consistent benchmark.
- The baseline models were chosen to cover a variety of solutions in time series forecasting.

Appendix



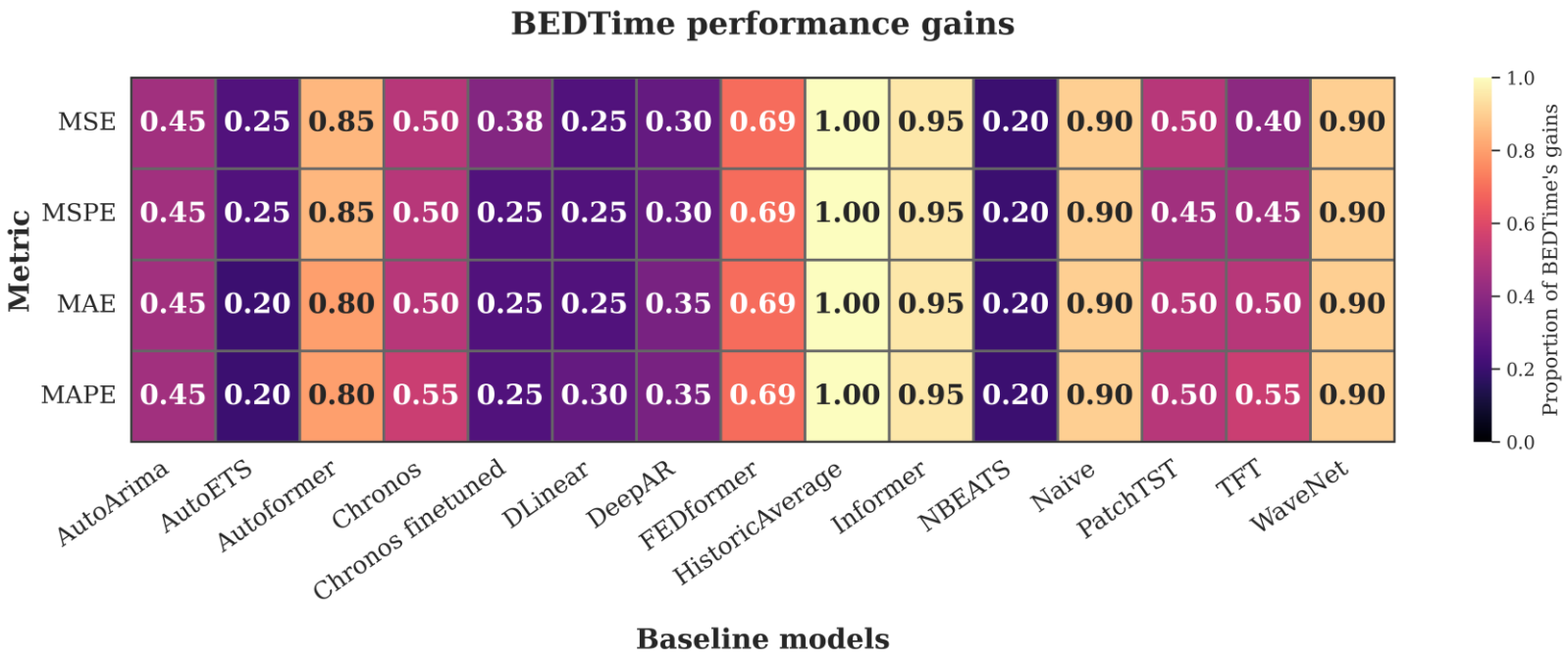
- Proportion of BEDTime wins with respect to the backbone and its corresponding time series prediction model on the PEM dataset (16 total timeseries)
- 240s training budget per model, split across backbones for BEDtime
- Results averaged across 5 seeds, 2 splits and regimes (one-shot and few-shot)

Appendix



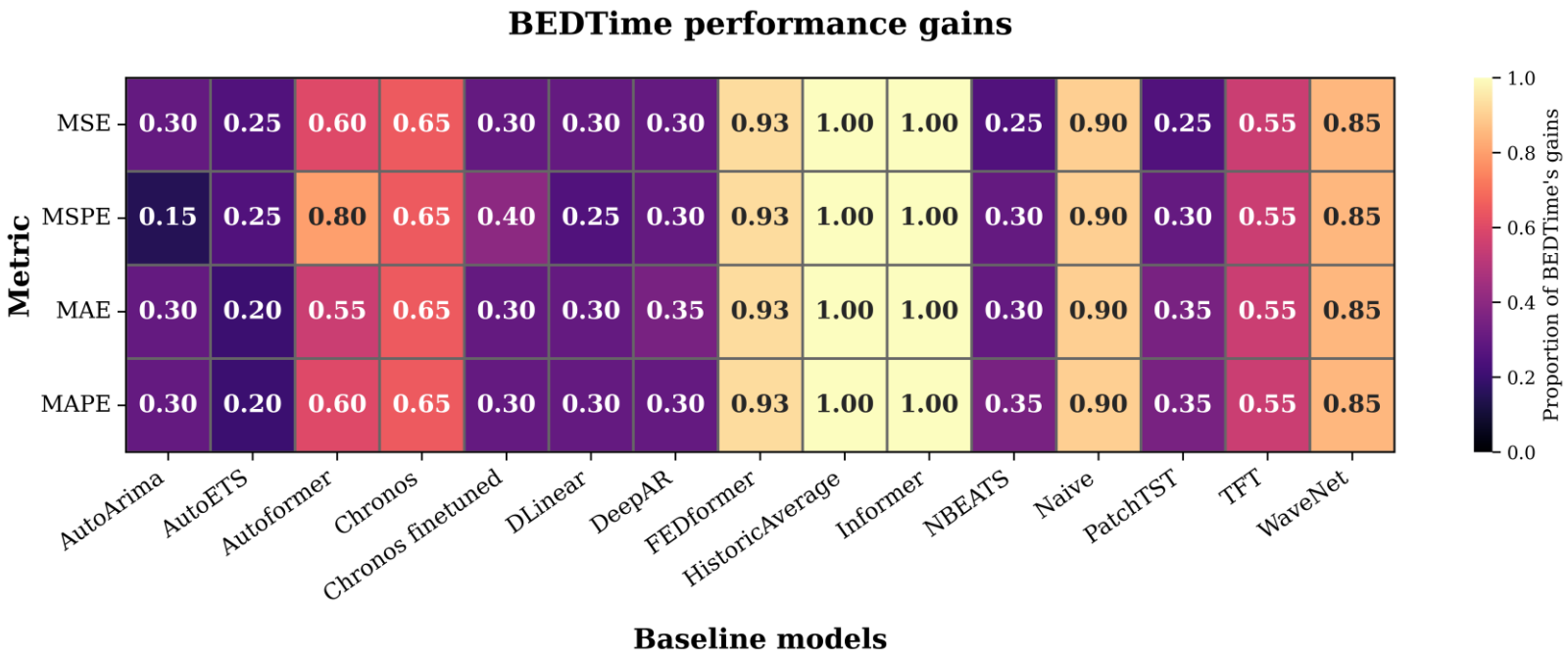
- Proportion of BEDTime wins with respect to the backbone and its corresponding time series prediction model on the PEM dataset (16 total timeseries)
- 60s training budget per model and per backbone
- Results averaged across 5 seeds, 2 splits and regimes (one-shot and few-shot)

Appendix



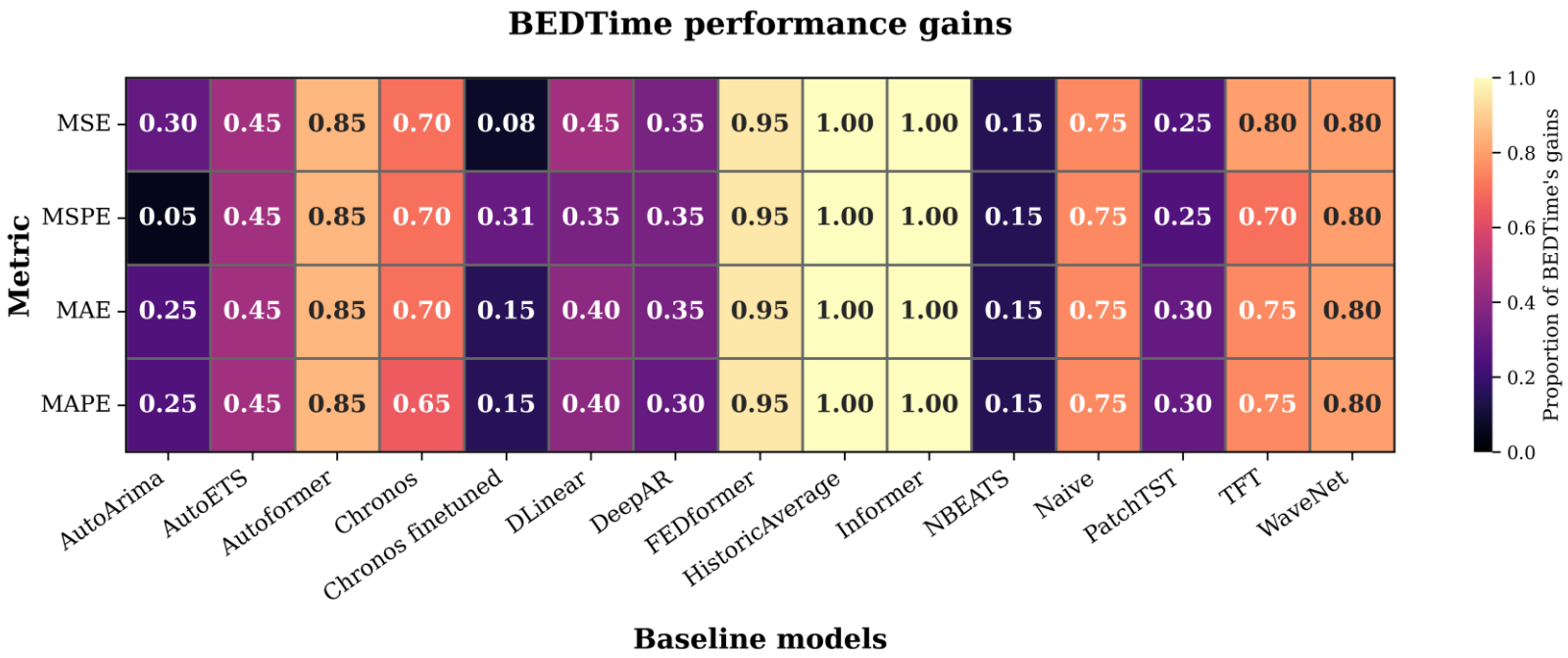
- Proportion of BEDTime wins with respect to the backbone and its corresponding time series prediction model on the PEM dataset (16 total timeseries)
- 30s training budget per model and per backbone
- Results averaged across 5 seeds, 2 splits and regimes (one-shot and few-shot)

Appendix



- Proportion of BEDTime wins with respect to the backbone and its corresponding time series prediction model on the Filter dataset (50 total timeseries)
- 60s training budget per model and per backbone
- Results averaged across 5 seeds, 2 splits and regimes (one-shot and few-shot)

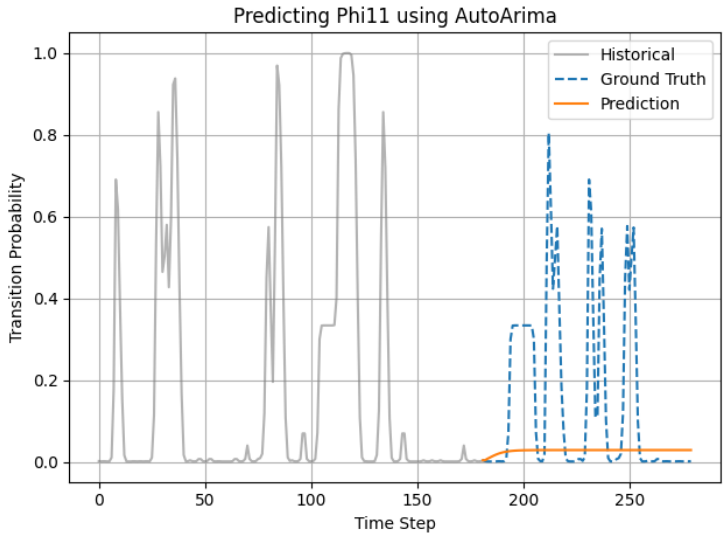
Appendix

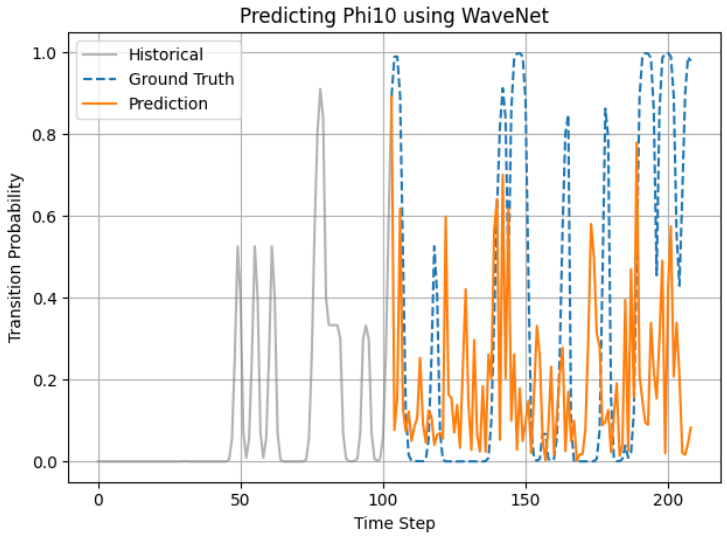


- Proportion of BEDTime wins with respect to the backbone and its corresponding time series prediction model on the Filter dataset (50 total timeseries)
- 30s training budget per model and per backbone
- Results averaged across 5 seeds, 2 splits and regimes (one-shot and few-shot)

Appendix

- BEDTime relies on backbone models to forecast transition probabilities, and its performance is influenced by the backbone's ability to model temporal dynamics in this space.
- Transformer-based models such as Autoformer, FEDformer and Informer yields noticeable improvements when integrated into BEDTime: architectures are well-suited to capture long-range temporal dependencies
- Classical statistical models such as ARIMA and ETS, see their performance decrease when coupled with BEDTime. This suggests that the transition dynamics require a more complex modeling.
- A similar behavior is seen with deep learning models with lower complexity such as DLinear, DeepAR, and NBEATS.
- BEDTime exhibits particularly strong performance when combined with WaveNet: architecture uses dilated convolutions to model wave-like structures, resembling the oscillatory patterns seen in the transition probability graphs



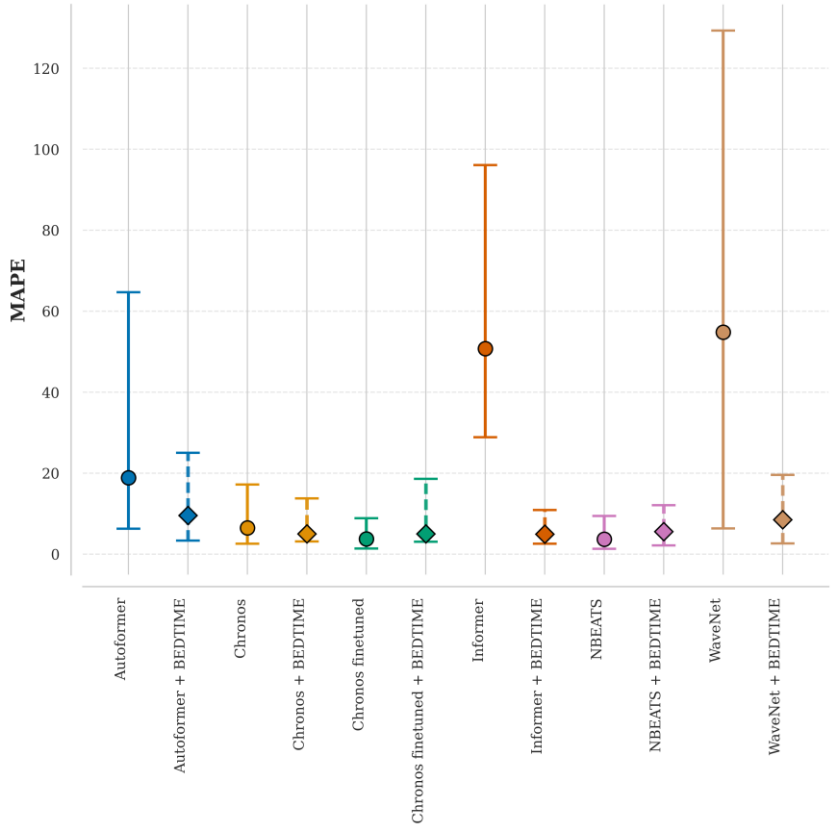


Appendix

- Analysis focused on key models: Autoformer, Chronos (zero-shot and fine-tuned), Informer, NBeats, and WaveNet.
- Utilized MAPE quantile range plots ($25\% - 75\%$ and $10\% - 90\%$) with median overlays.
- BEDTime significantly reduced MAPE for Transformer-based models (Autoformer, Informer) and WaveNet, shifting median errors toward minimal regions.
- A notable decrease in error variance was observed, demonstrating improved model consistency.
- For NBeats and fine-tuned Chronos, BEDTime resulted in slightly increased errors, though distributions remained comparable and accuracy losses were deemed minimal.

- Quantiles were calculated from a sample size of 20, derived from a 60-second training budget across two training sizes, two test splits, and five random seeds.

10-90 Quantile range analysis on the PEM dataset

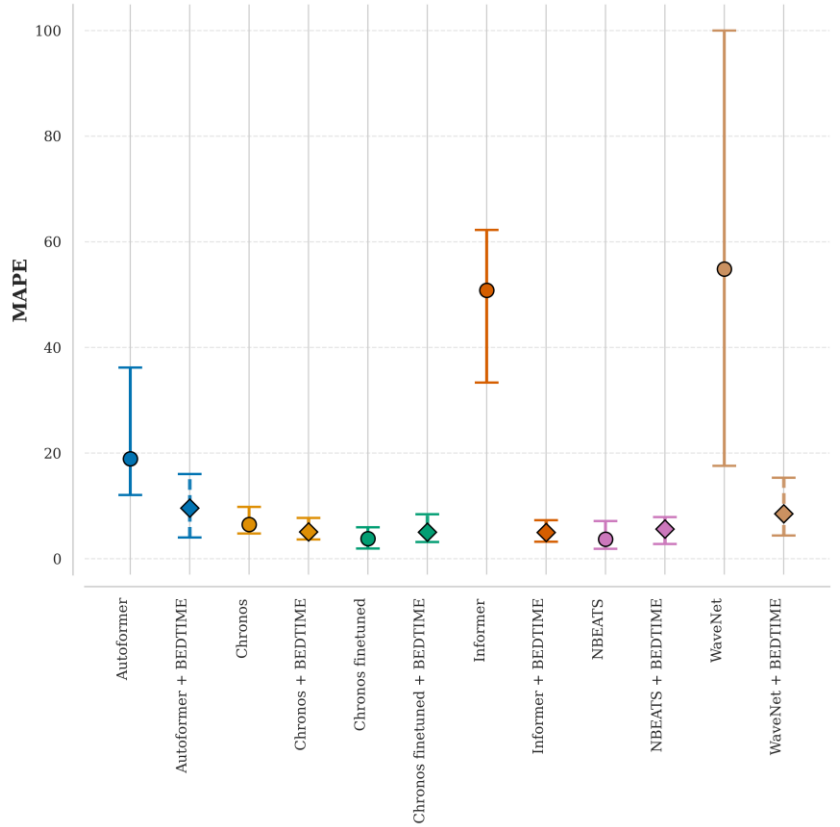


Appendix

- Analysis focused on key models: Autoformer, Chronos (zero-shot and fine-tuned), Informer, NBeats, and WaveNet.
- Utilized MAPE quantile range plots ($25\% - 75\%$ and $10\% - 90\%$) with median overlays.
- BEDTime significantly reduced MAPE for Transformer-based models (Autoformer, Informer) and WaveNet, shifting median errors toward minimal regions.
- A notable decrease in error variance was observed, demonstrating improved model consistency.
- For NBeats and fine-tuned Chronos, BEDTime resulted in slightly increased errors, though distributions remained comparable and accuracy losses were deemed minimal.

- Quantiles were calculated from a sample size of 20, derived from a 60-second training budget across two training sizes, two test splits, and five random seeds.

25-75 Quantile range analysis on the PEM dataset

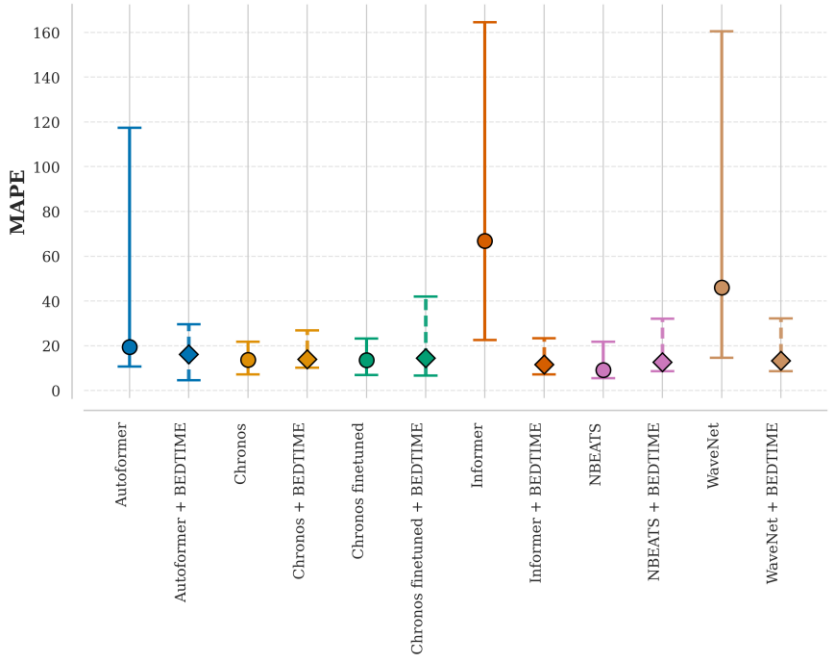


Appendix

- Analysis focused on key models: Autoformer, Chronos (zero-shot and fine-tuned), Informer, NBeats, and WaveNet.
- Utilized MAPE quantile range plots ($25\% - 75\%$ and $10\% - 90\%$) with median overlays.
- BEDTime significantly reduced MAPE for Transformer-based models (Autoformer, Informer) and WaveNet, shifting median errors toward minimal regions.
- A notable decrease in error variance was observed, demonstrating improved model consistency.
- For NBeats and fine-tuned Chronos, BEDTime resulted in slightly increased errors, though distributions remained comparable and accuracy losses were deemed minimal.

- Quantiles were calculated from a sample size of 20, derived from a 60-second training budget across two training sizes, two test splits, and five random seeds.

10-90 Quantile range analysis on the Fitler dataset

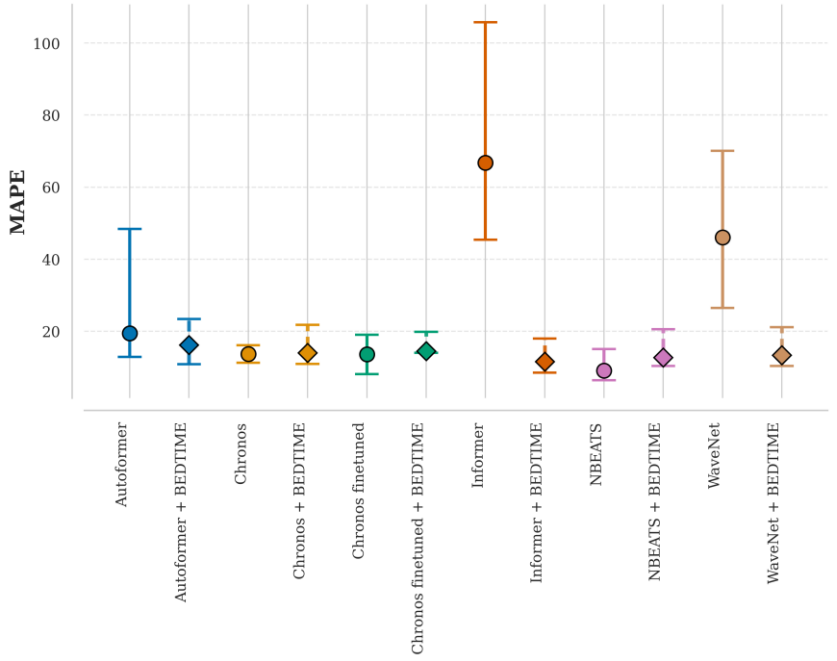


Appendix

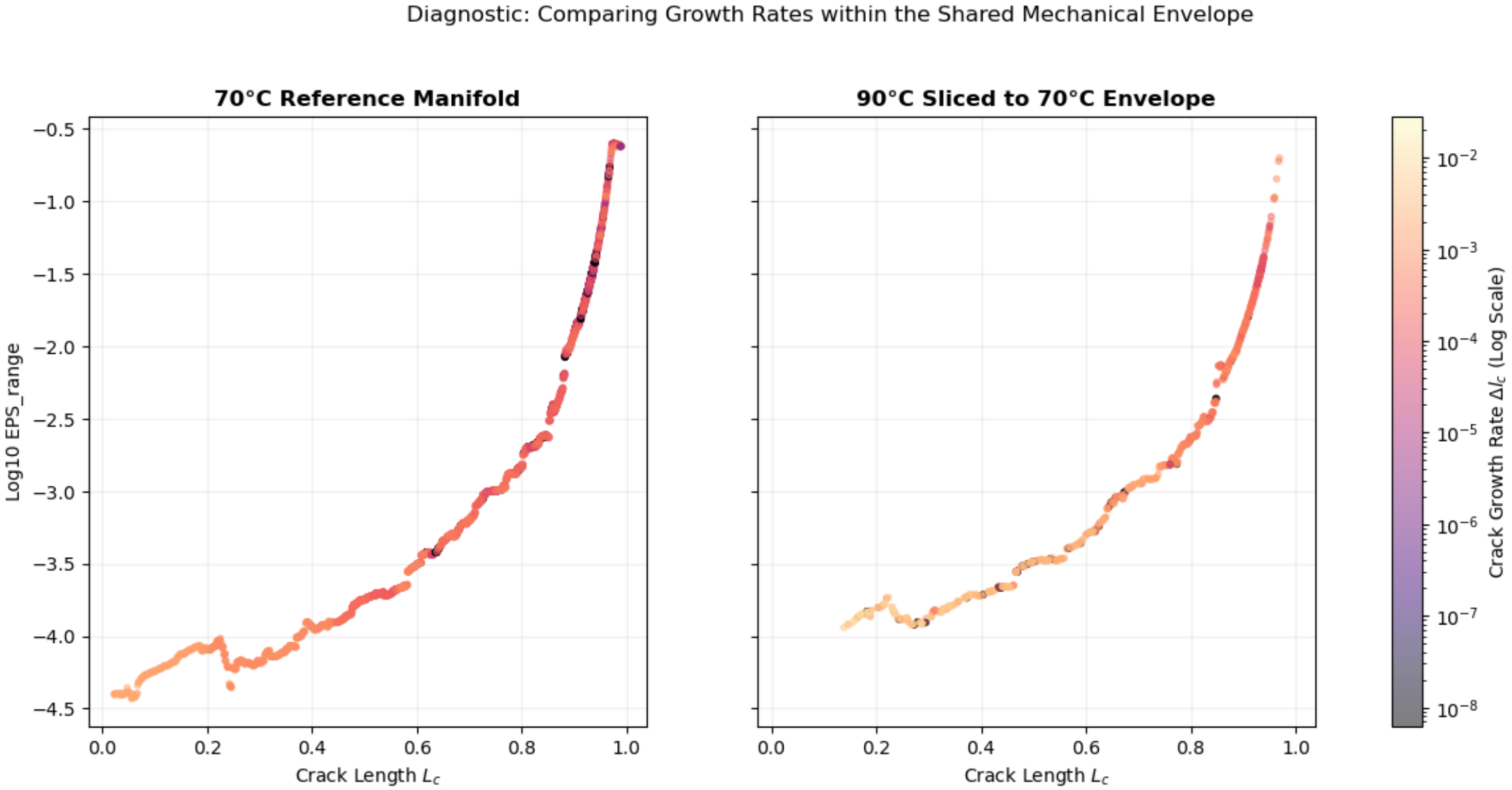
- Analysis focused on key models: Autoformer, Chronos (zero-shot and fine-tuned), Informer, NBeats, and WaveNet.
- Utilized MAPE quantile range plots ($25\% - 75\%$ and $10\% - 90\%$) with median overlays.
- BEDTime significantly reduced MAPE for Transformer-based models (Autoformer, Informer) and WaveNet, shifting median errors toward minimal regions.
- A notable decrease in error variance was observed, demonstrating improved model consistency.
- For NBeats and fine-tuned Chronos, BEDTime resulted in slightly increased errors, though distributions remained comparable and accuracy losses were deemed minimal.

- Quantiles were calculated from a sample size of 20, derived from a 60-second training budget across two training sizes, two test splits, and five random seeds.

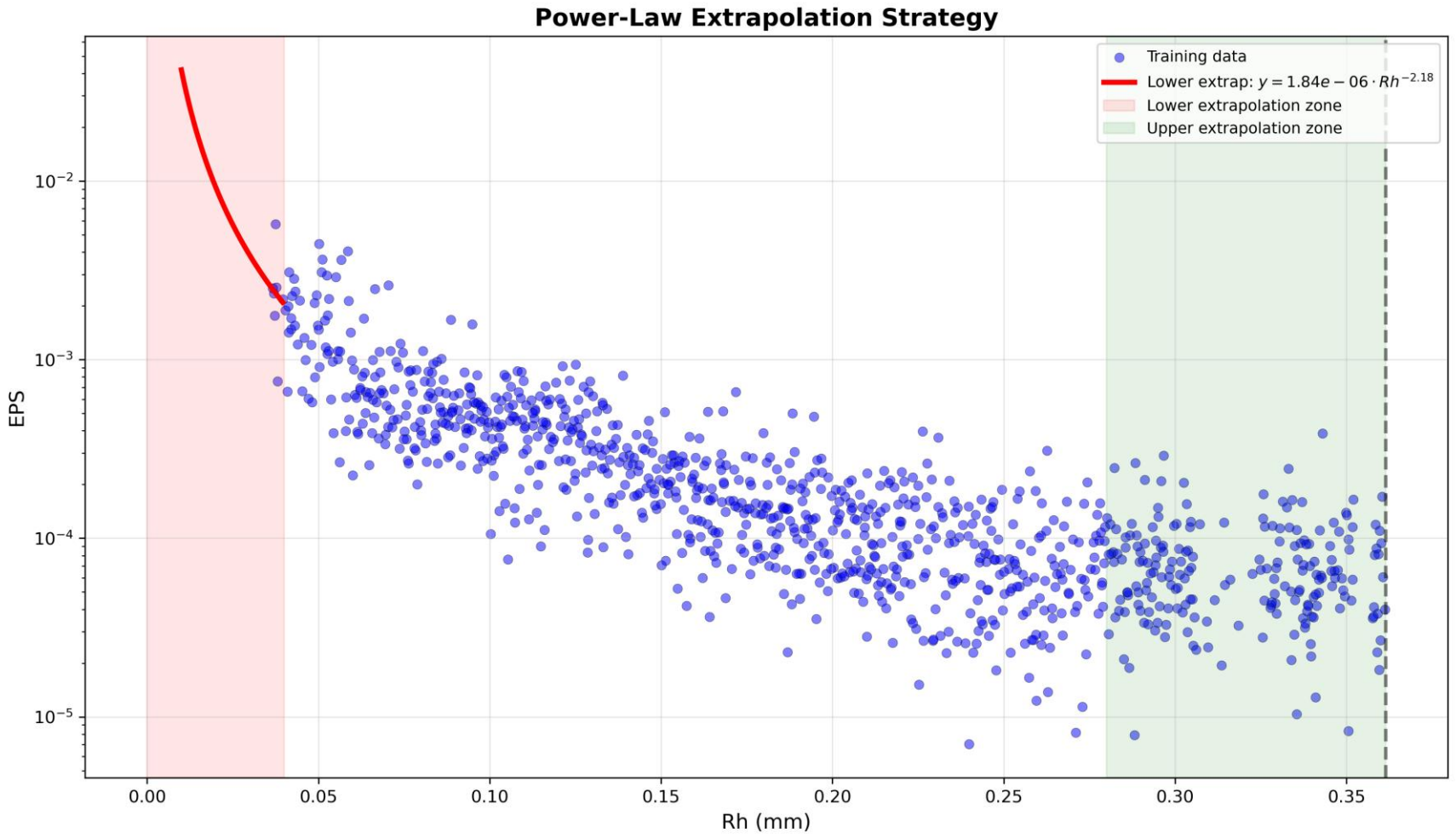
25-75 Quantile range analysis on the Fitler dataset



Appendix


Diagnostic: Comparing Growth Rates within the Shared Mechanical Envelope
70°C Reference Manifold
90°C Sliced to 70°C Envelope
Log10 EPS_range
Crack Length L_c
Crack Growth Rate Δl_c (Log Scale)

Appendix


Power-Law Extrapolation Strategy
Training data
Lower extrap: $y = 1.84e-06 \cdot Rh^{-2.18}$
Lower extrapolation zone
Upper extrapolation zone
EPS
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Rh (mm)