Reducing Machine Learning Model Emissions by Integrating Physics Knowledge

Daisy R Bradley[1], Elizabeth J Cross[1]

[1]Dynamics Research Group, School of Mechanical, Aerospace and Civil Engineering, University of Sheffield

Contents

Engineering Computing

Almost all engineers/engineering companies are using artificial intelligence (AI) or machine learning (ML)

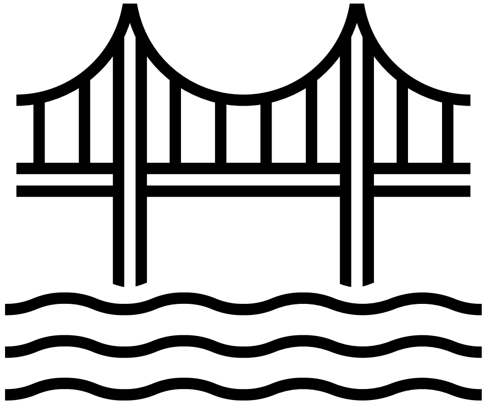
- Data analysis
- Generative design
- Project planning
- Code writing
- Predictive modelling

Energy consumption

Engineering research focusses on achieving sustainability related goals

Engineering Machine Learning

Data
- Temperatures
- Weathers
- Wind speeds
- Wind directions
- Loading from cars
- Loading from waves
- Tide levels



Physical knowledge
- Materials
- Dimensions
- Manufacturing techniques
- Structural mechanics theory
- Wave force models
- Design constraints

Physics-Informed Machine Learning

Increased interpretability, increased extrapolation performance →

← Less understanding required, superior flexibility to represent data

Black-box modelling:
- Gaussian processes
- Neural networks
- Random forests
- Support vector machines

Grey-box modelling
- PINNs

White-box modelling:
- Finite element methods
- Computational fluid dynamics
- Differential equations
- Conservation laws

Our Research Objective

Increase physics in model → Increased model complexity → Increased runtime and emissions?

↓

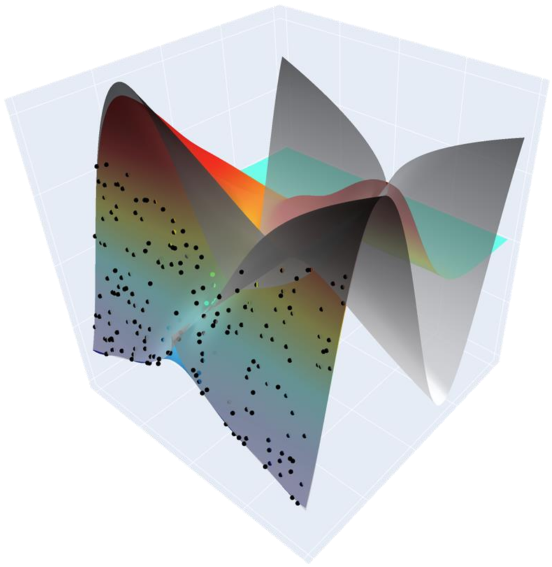
Increase extrapolative performance

↓

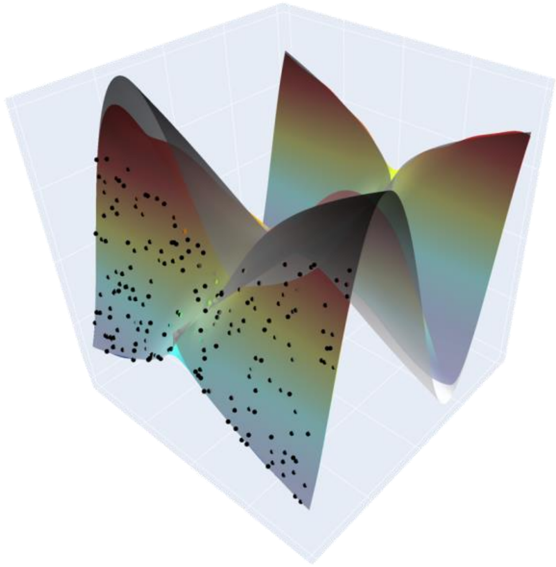
Decrease training data requirement

↓

Decrease runtime and emissions?



Black-Box Model



Grey-Box Model

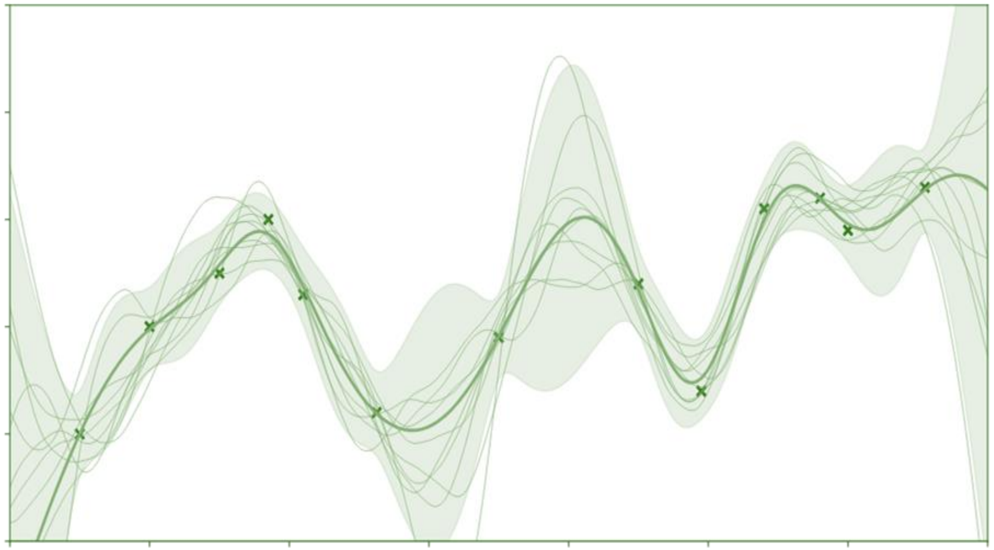
Gaussian Processes

Non-parametric Bayesian approach to regression and classification
Provide a predictive distribution over a family of functions whose properties are defined by hyperparameters

$$f \sim GP(m, k)$$

Our kernel function:

- Tells the model how two inputs correlate
- Defines properties of the function (smoothness, periodicity)
- Can be combined with other kernel functions
- Contains hyperparameters to be learnt during training
- Creates a large matrix that must be inverted during training

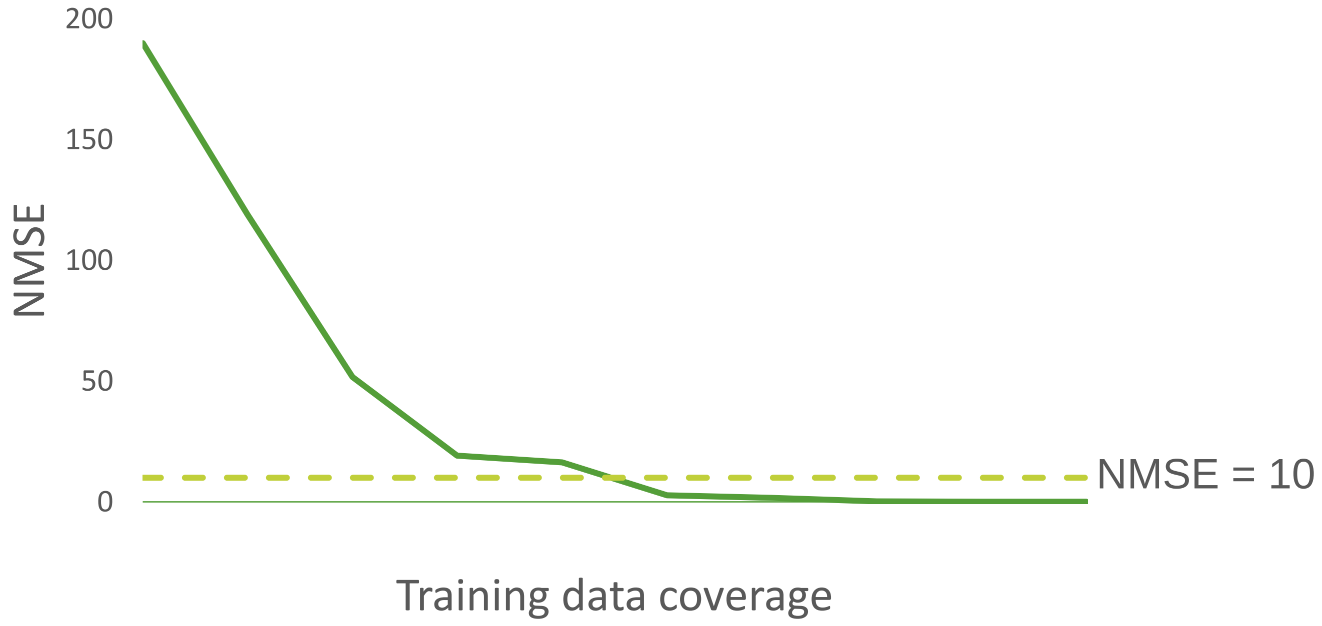


Confidence intervals

Performance Thresholds

Performance targets

$$NMSE = \frac{100\sum(y - y_*)^2}{n\sigma}$$

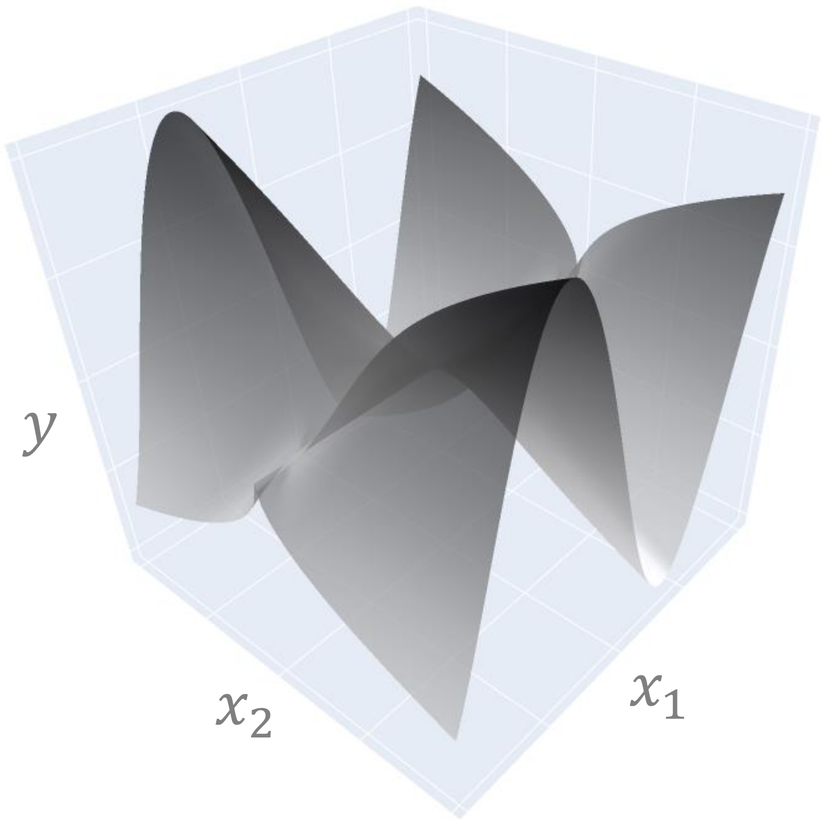


Toy-Box Test Case

$$y = \sqrt{|x_2|}\sin(4x_1)$$

Predict y given x_1 and x_2

NMSE < 10 for 9 runs

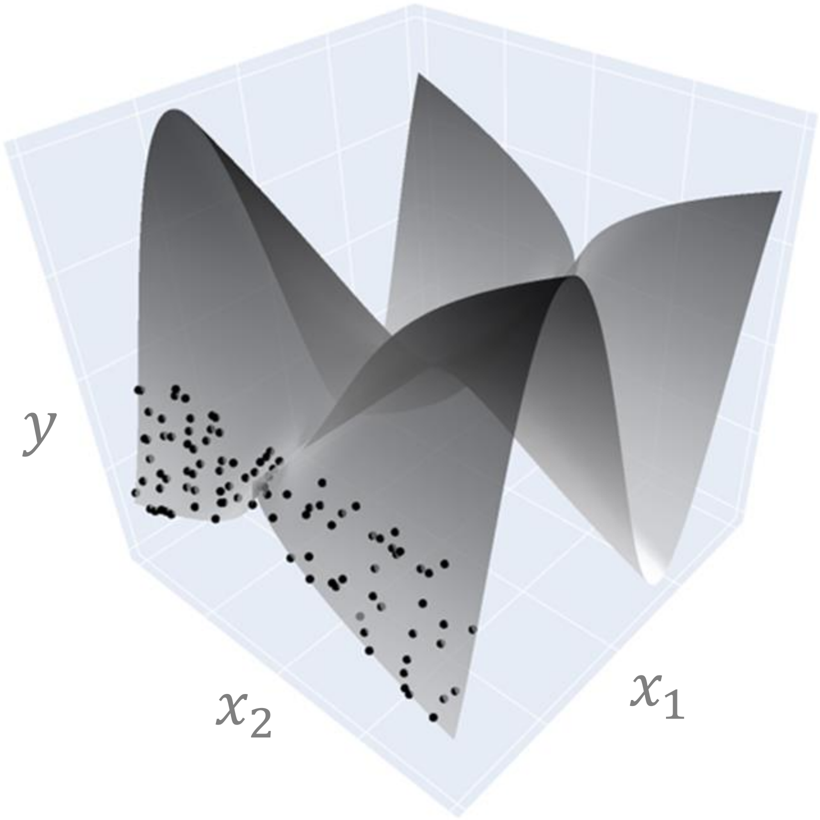


Toy-Box Test Case

$$y = \sqrt{|x_2|}\sin(4x_1)$$

Predict y given x_1 and x_2

NMSE < 10 for 9 runs

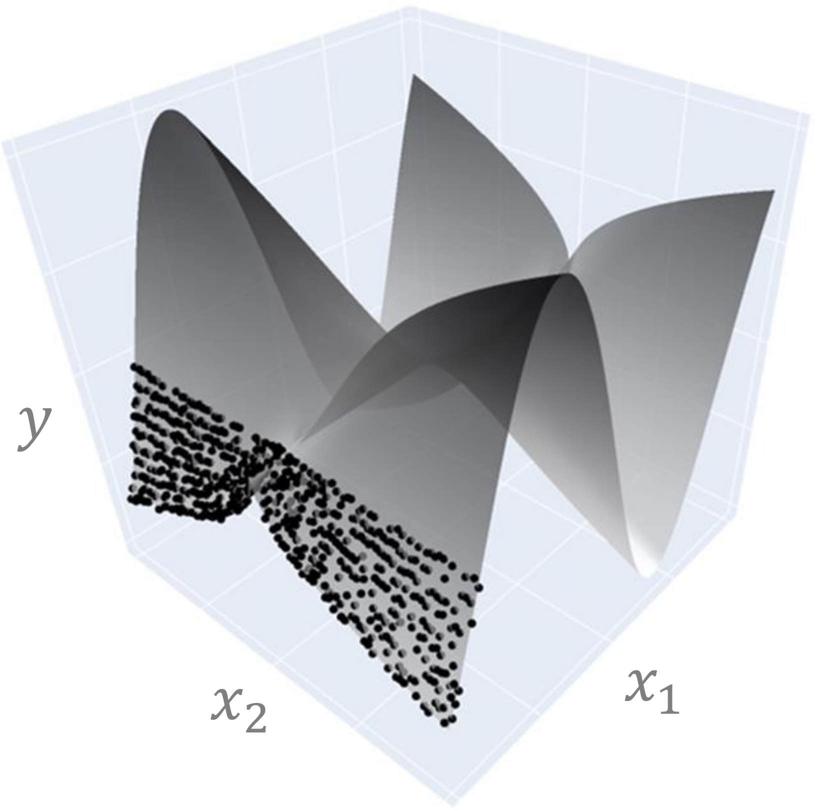


Toy-Box Test Case

$$y = \sqrt{|x_2|}\sin(4x_1)$$

Predict y given x_1 and x_2

NMSE < 10 for 9 runs



Toy-Box Test Case

Black-1

$$k_{SE}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)$$

+ mean and observation noise

4 hyperparameters

Grey-1

$$k_{SE\times Per}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2l_{SE}^2}\right)\exp\left(-\frac{2sin^2\left(\frac{\pi|x-x'|}{p}\right)}{l_{Per}^2}\right)$$

6 hyperparameters

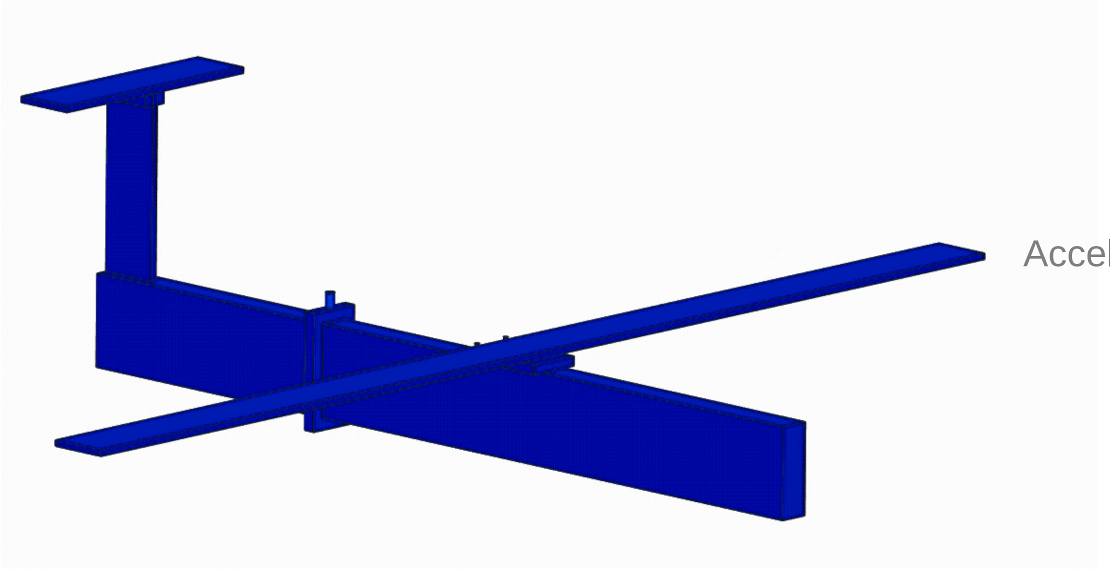
Grey-2

$$k_{SE\times Per}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2l_{SE}^2}\right)\exp\left(-\frac{2sin^2\left(\frac{\pi|x-x'|}{p}\right)}{l_{Per}^2}\right)$$

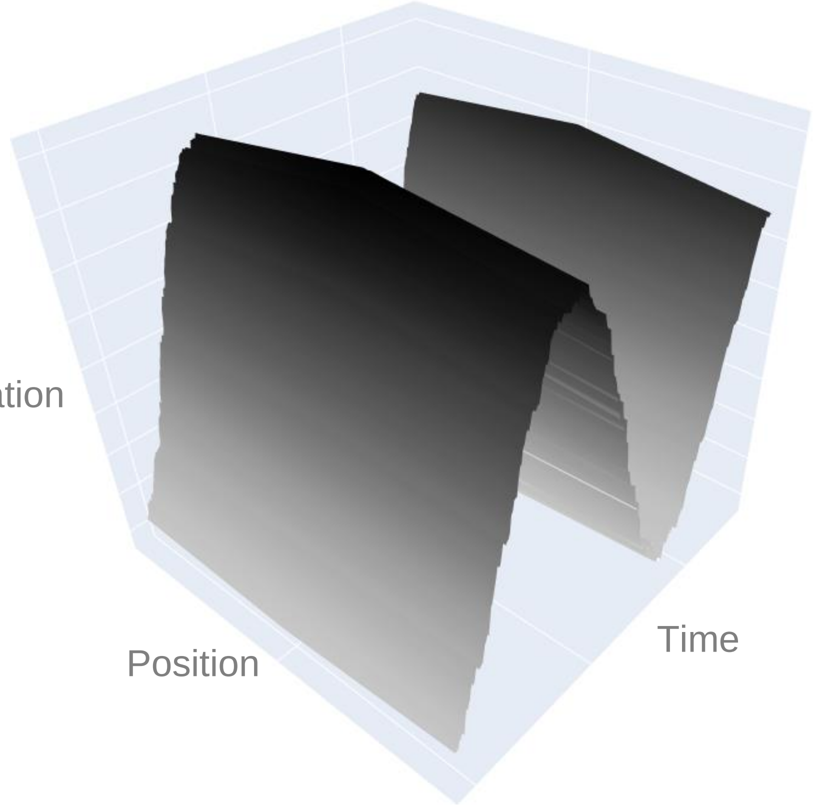
5 hyperparameters

An Engineering Case Study

Structural health monitoring



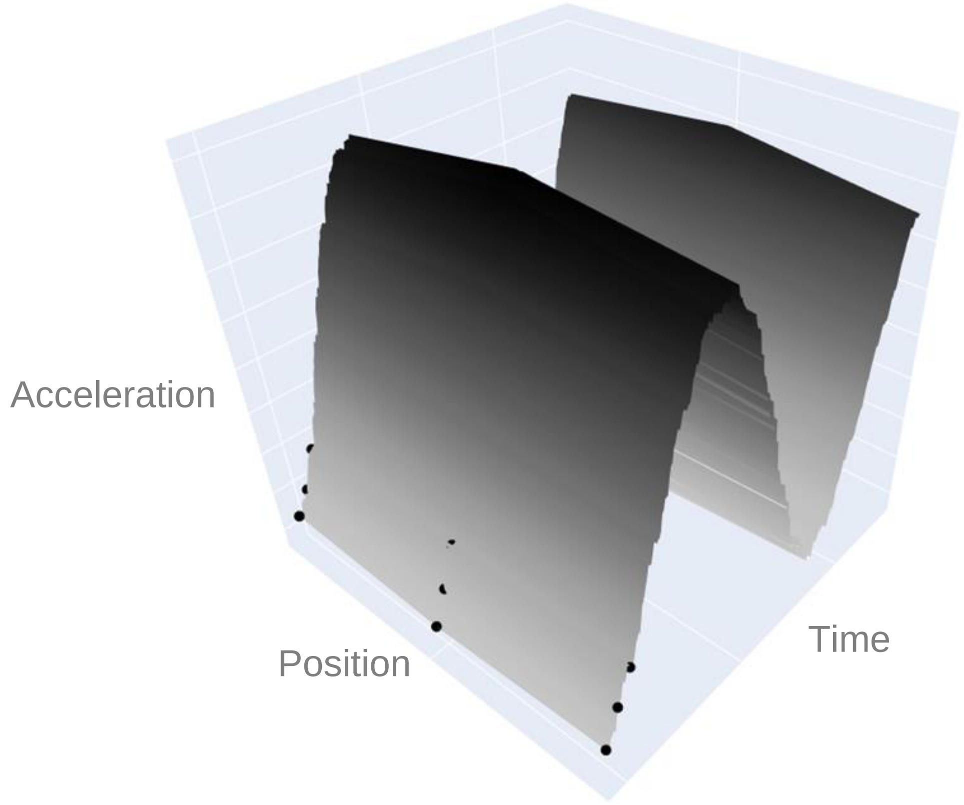
The GARTEUR SM-AG19



An Engineering Case Study

Structural health monitoring

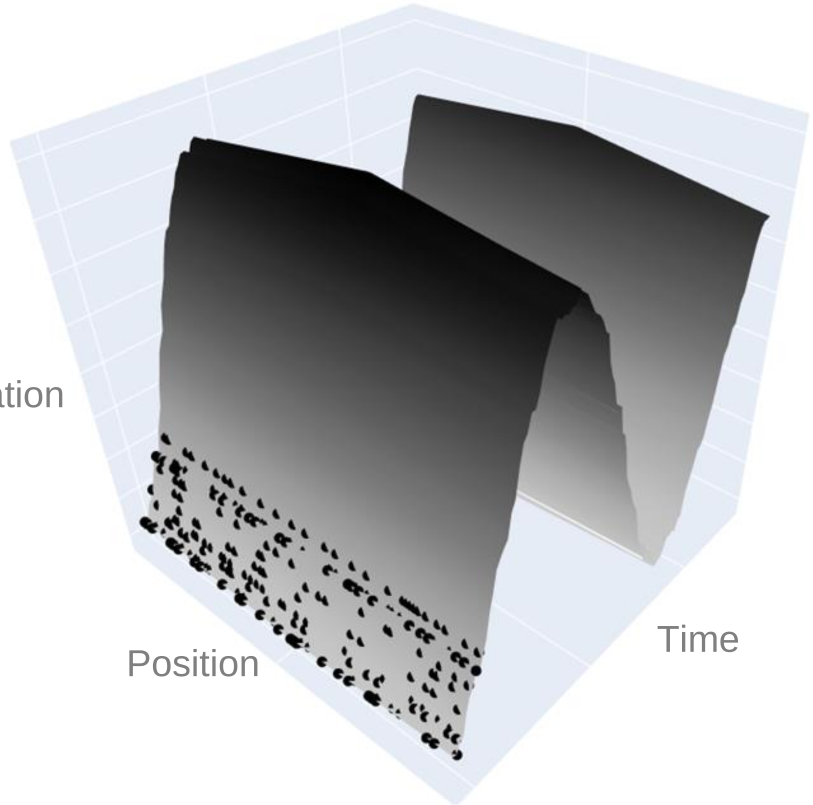
The GARTEUR SM-AG19



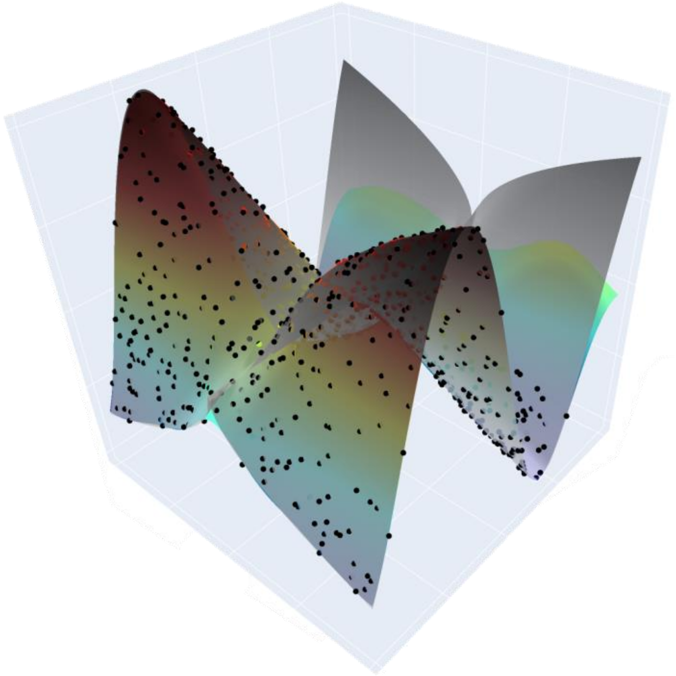
An Engineering Case Study

Structural health monitoring

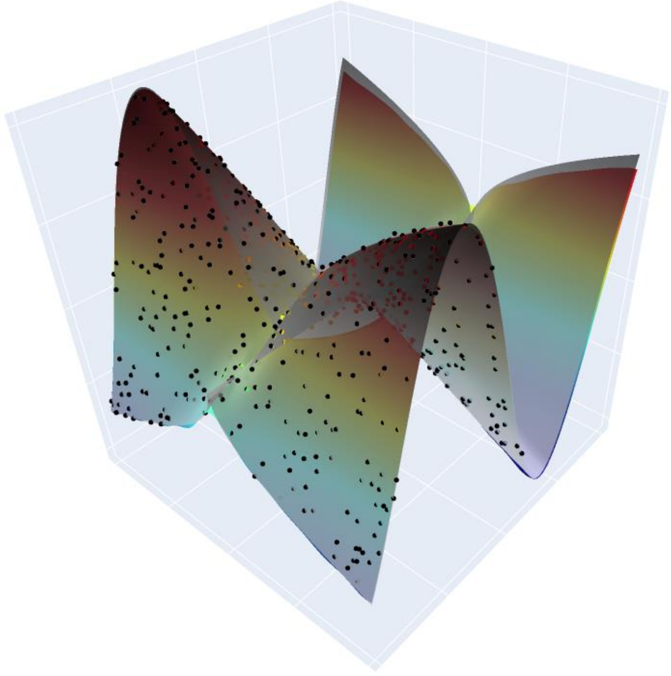
The GARTEUR SM-AG19



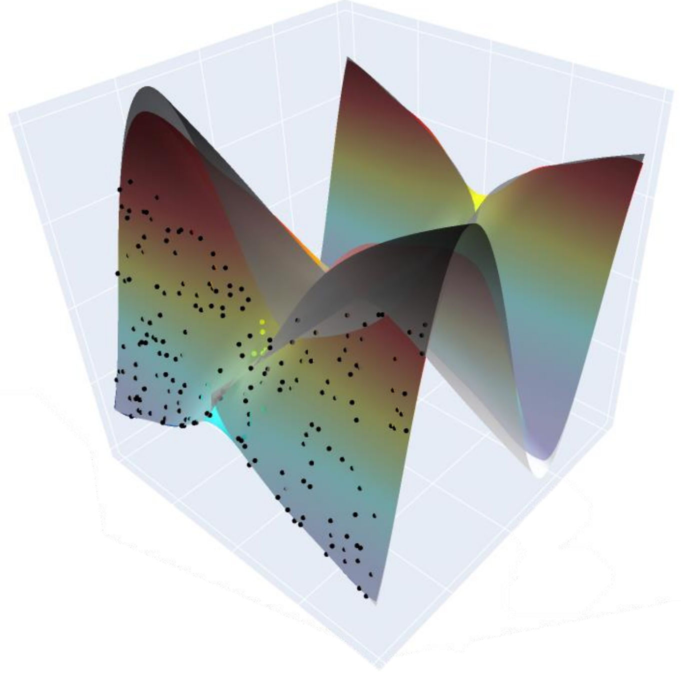
Training Data Coverage Results



Black-1
80%



Grey-1
60%

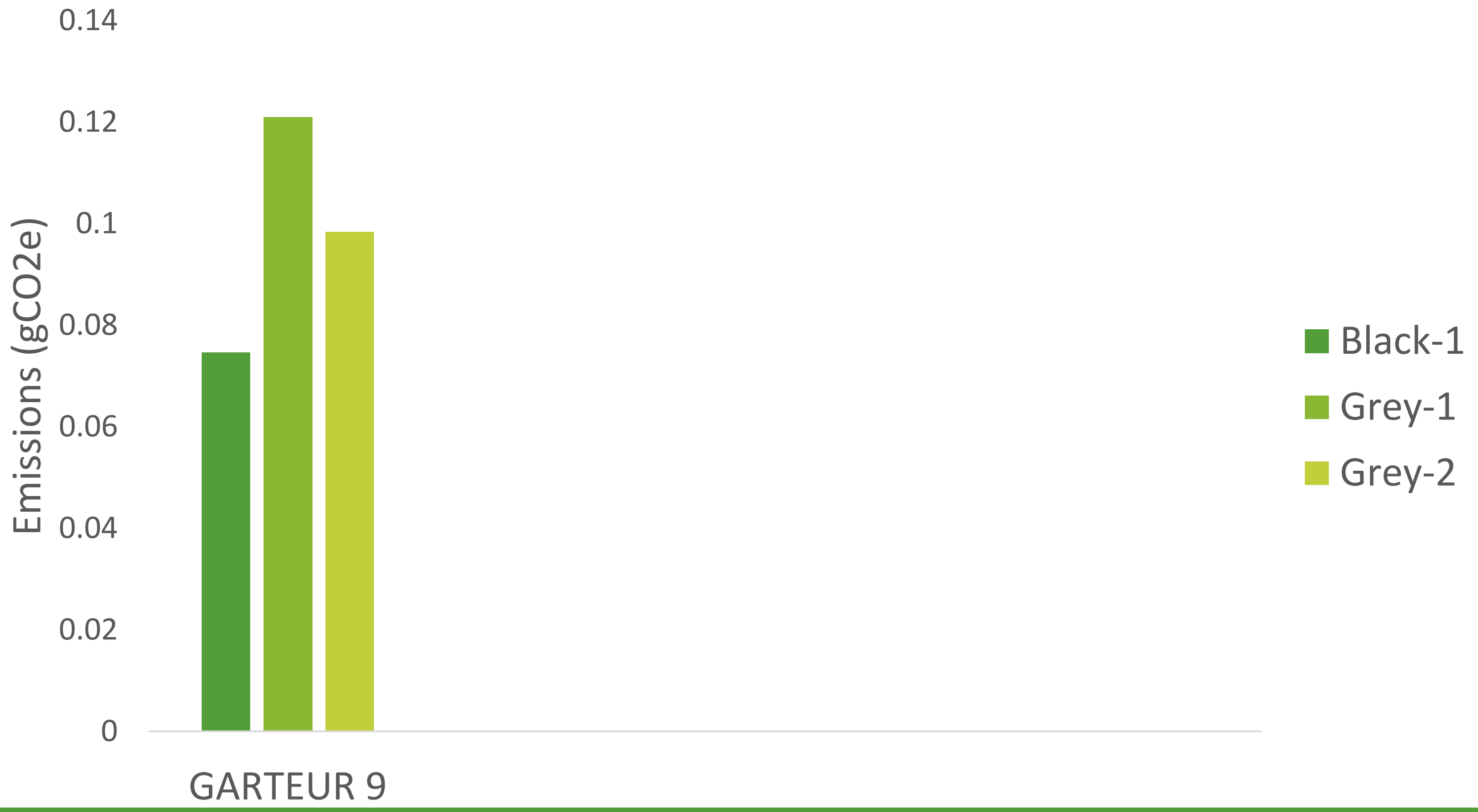


Grey-2
20%

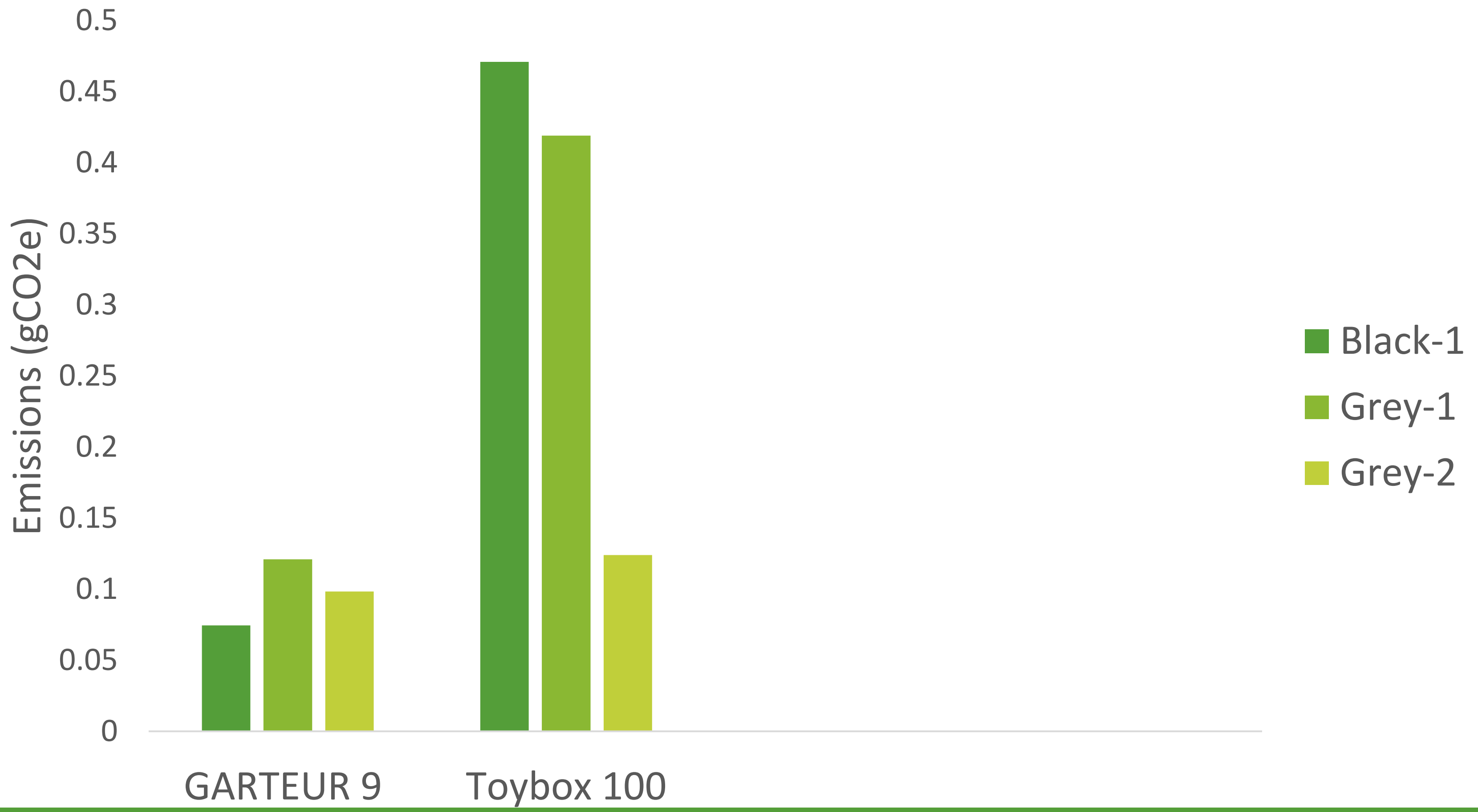
Training Data Coverage Results

	Black-1	Grey-1	Grey-2
Toy-box 100 coverage (%)	80	60 (-25%)	20 (-75%)
Toy-box 600 coverage (%)	80	70 (-13%)	20 (-75%)
GARTEUR 9 coverage (%)	70	60 (-14%)	30 (-57%)
GARTEUR 200 coverage (%)	80	60 (-25%)	20 (-75%)

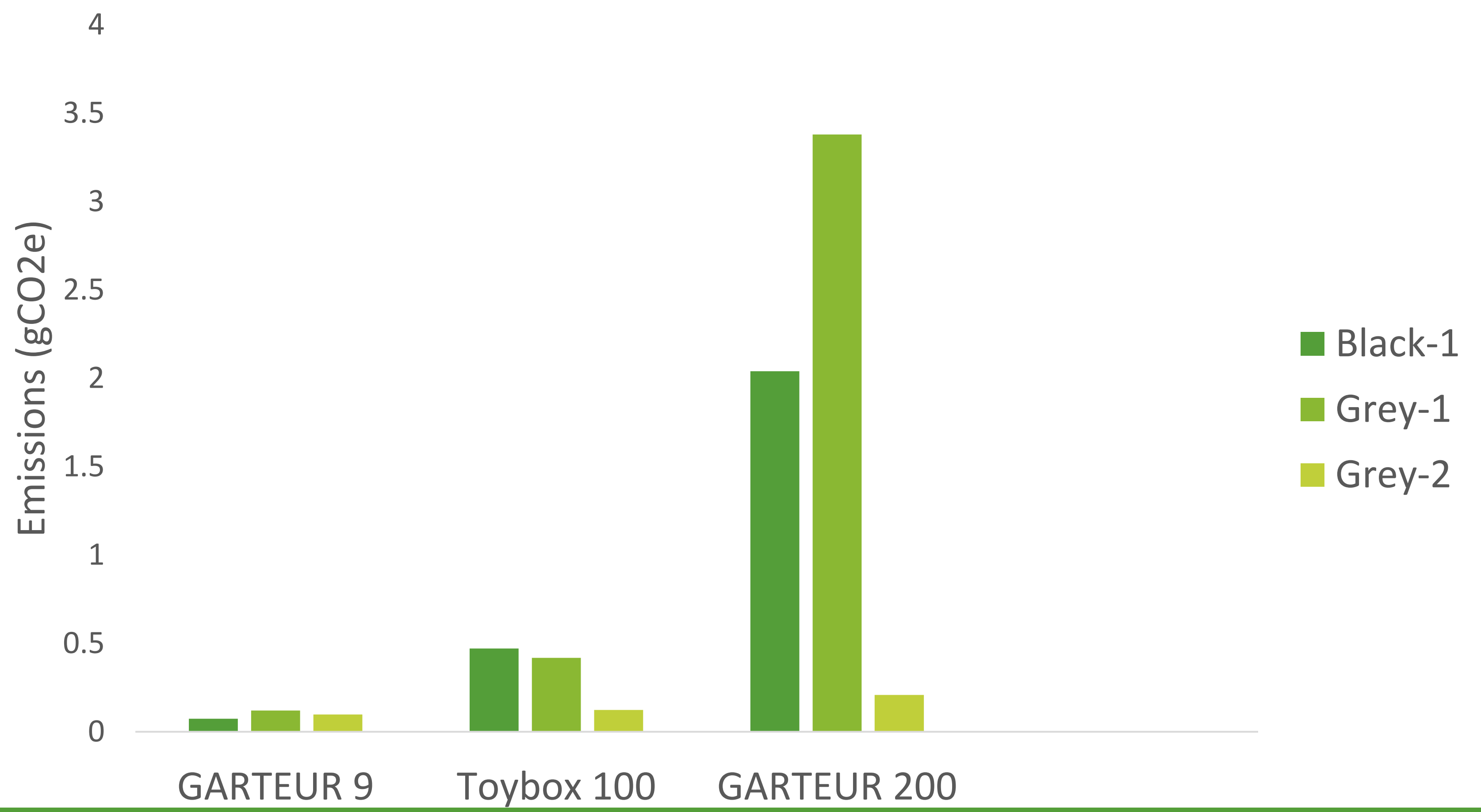
CodeCarbon Results



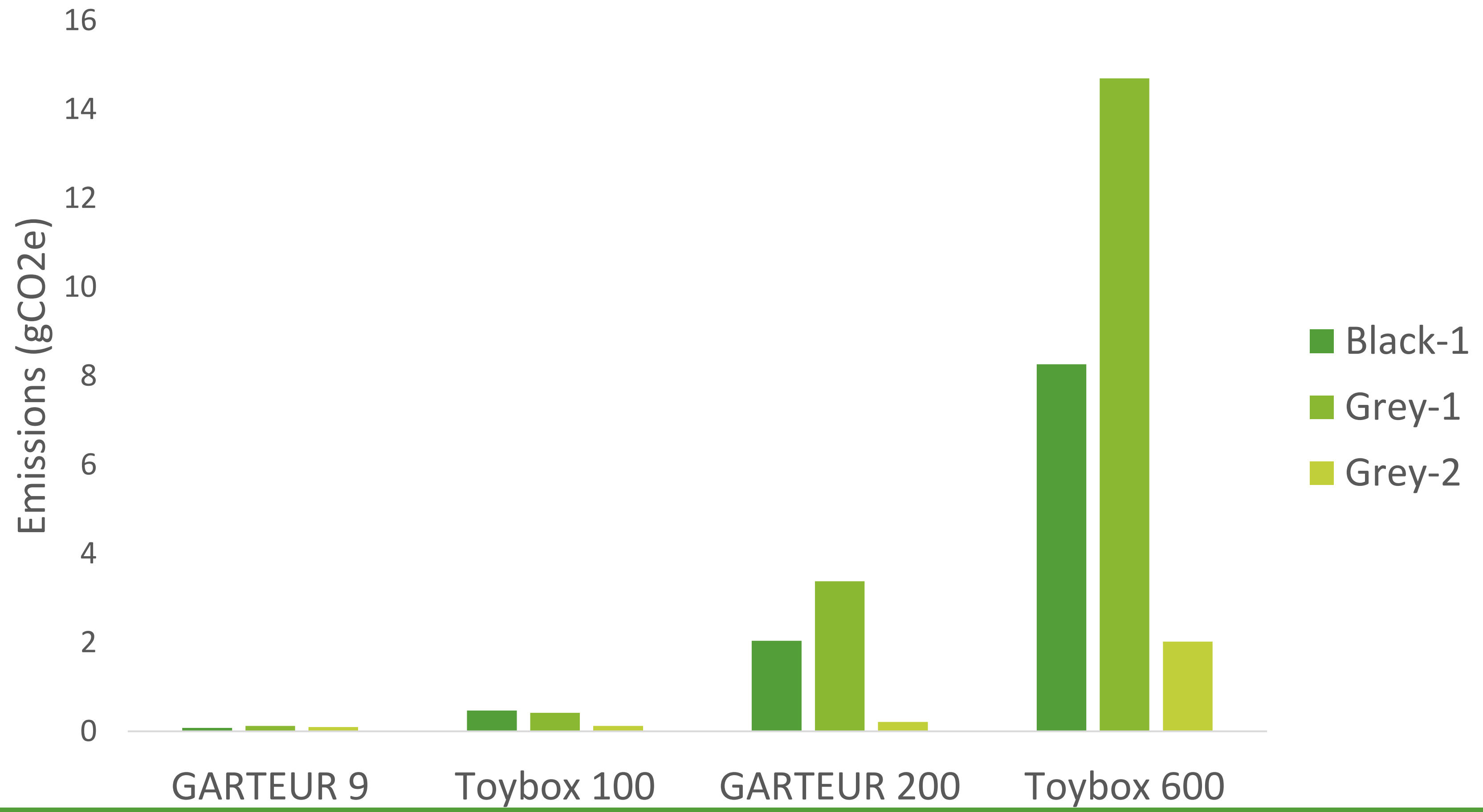
CodeCarbon Results



CodeCarbon Results



CodeCarbon Results



Emissions Discussion

CodeCarbon ~ Runtime

$$Runtime \sim O(N^3) + O(HN^2)$$

From covariance matrix inversion

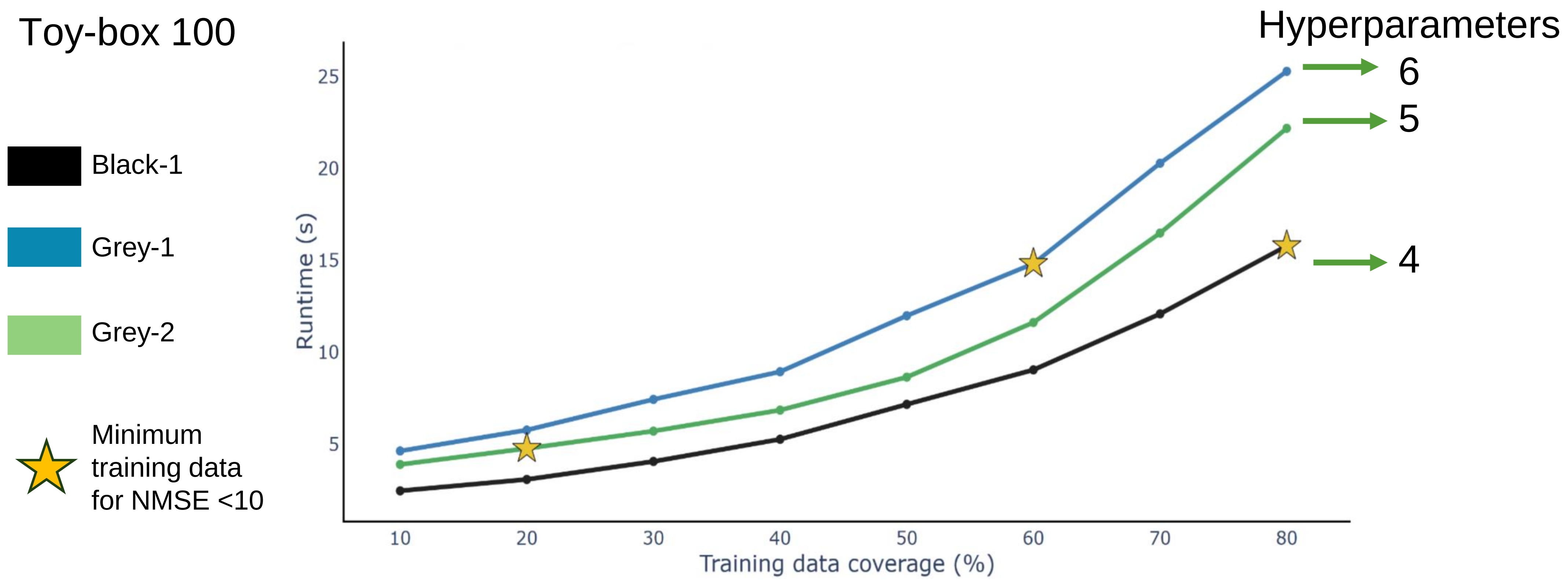
From hyperparameter optimisation

N = number of training data points
H = number of hyperparameters

In this work, N and H are both very small

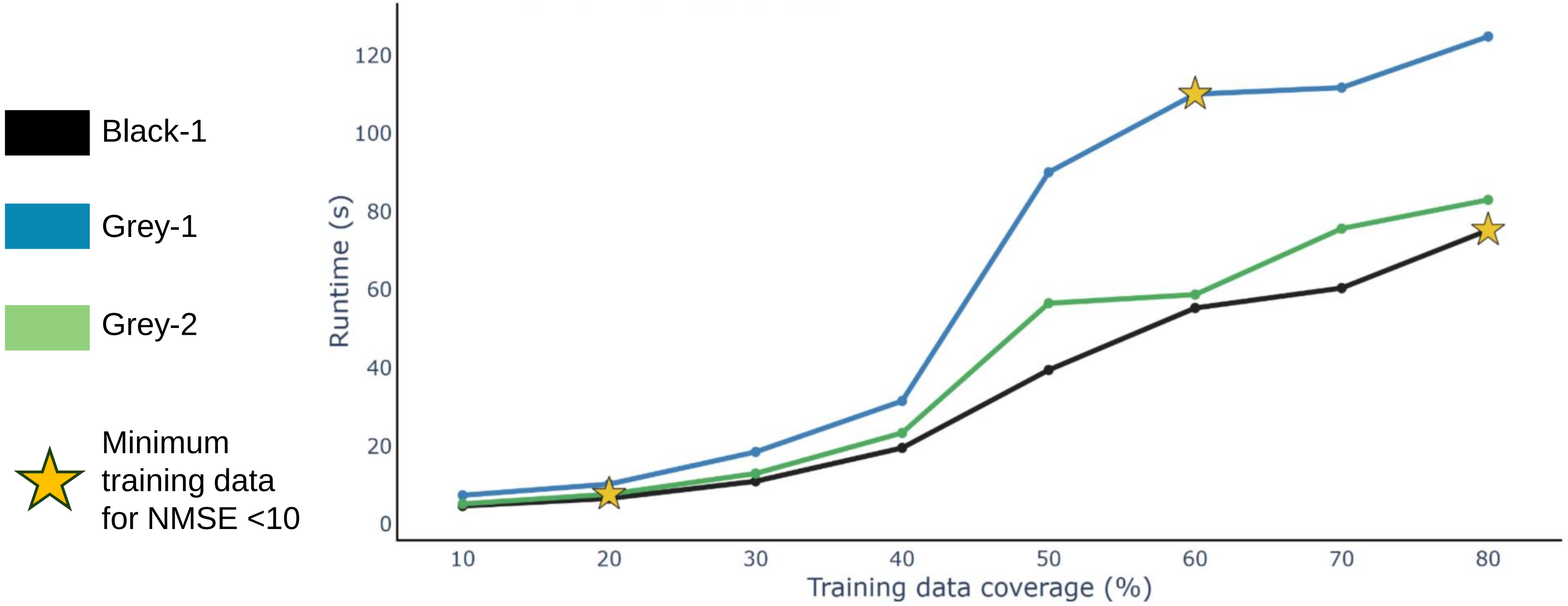
Emissions Discussion

Toy-box 100



Emissions Discussion

GARTEUR 200



Future Work

This is a preliminary study

Training data and hyperparameter counts are small

Model and test cases are very simple

Future work will:

- Upscale training data sizes
- Expand into more complex Gaussian Processes
- Investigate other models e.g. PINNs (linear scaling laws)

Conclusions

- Preliminary investigation
- When physics integration reduces training data, do we reduce our ML emissions?
- In some cases, yes!
- Training data vs hyperparameter trade-off
- More investigation needed with larger data sets and complex models

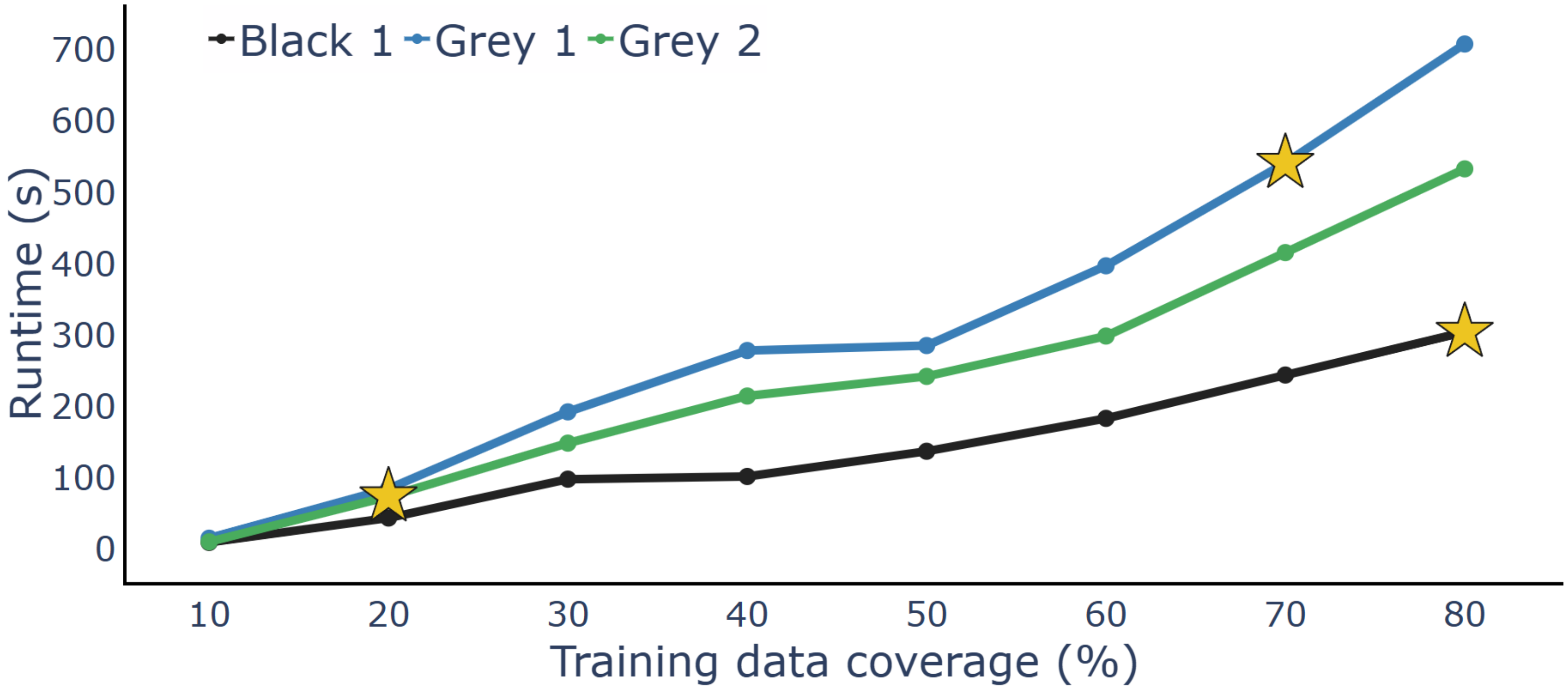
Read the paper here!

Thank you for listening!

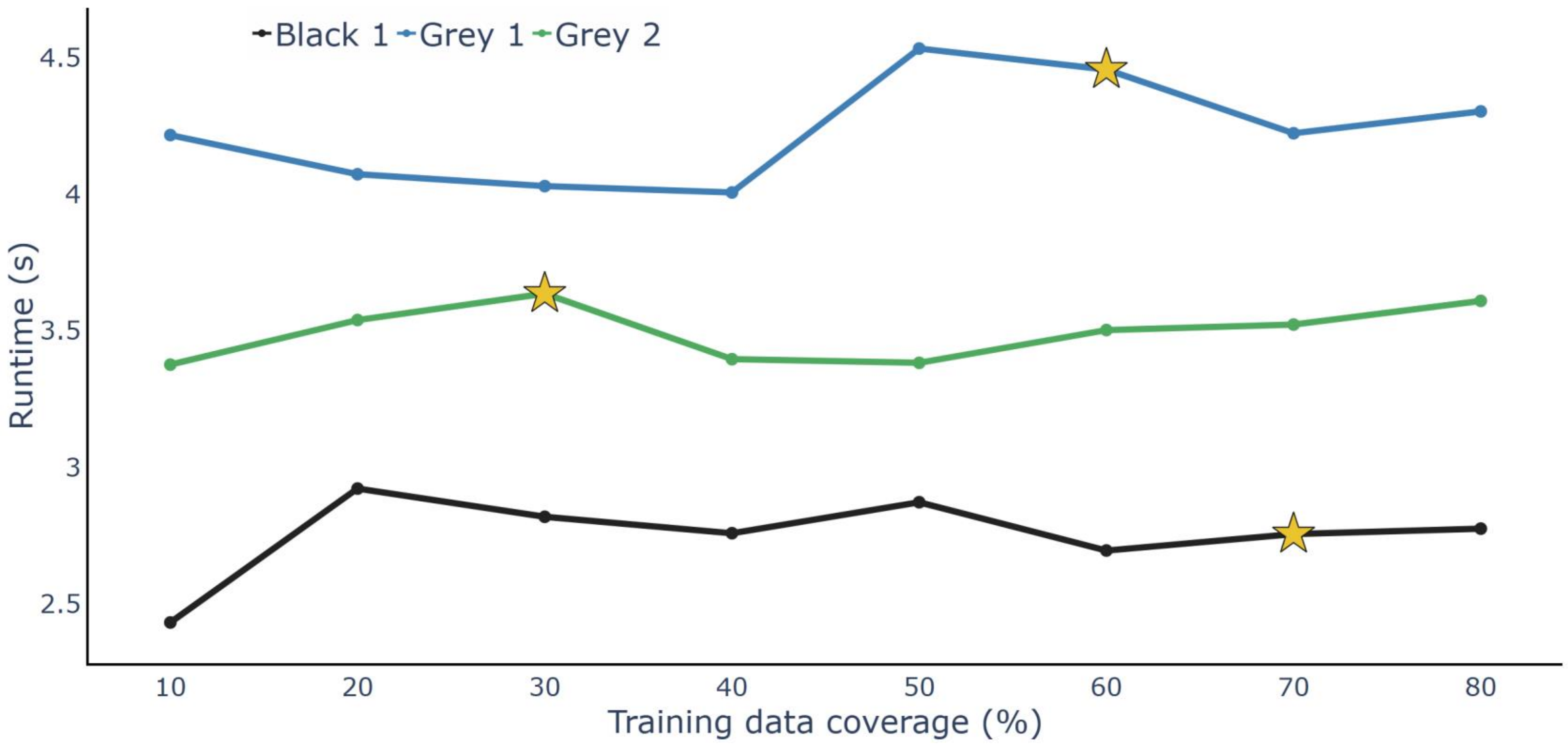
Email me: dbradley2@sheffield.ac.uk

Connect on LinkedIn

	Black-1	Grey-1	Grey-2
GARTEUR original emissions (GCO2e)	0.0746	0.121 (+62%)	0.0984 (+32%)
Toy-box 100 emissions (GCO2e)	0.471	0.419 (-11%)	0.124 (-74%)
GARTEUR 200 emissions (GCO2e)	2.04	3.38 (+66%)	0.209 (-90%)
Toy-box 600 emissions (GCO2e)	8.26	14.7 (+78%)	2.02 (-76%)



Black 1
Grey 1
Grey 2
Runtime (s)
0
100
200
300
400
500
600
700
10
20
30
40
50
60
70
80
Training data coverage (%)



Black 1
Grey 1
Grey 2
4.5
4
3.5
3
2.5
Runtime (s)
10
20
30
40
50
60
70
80
Training data coverage (%)