Physics-Informed Neural Networks

using Curriculum Learning

Background

What are PINNs?

PINNs are a class of neural networks which respect in their output the governing laws of physics described through differential equations. This can be achieved by introducing physical domain knowledge by PDEs into the network using the loss function.

What is curriculum learning in the context of PINNs?

Within this context, curriculum learning is a training strategy that gradually increases the complexity of the training data, allowing the network to train on simpler examples first and then gradually increasing the complexity.

Motivation

Neural networks are data-driven models, learning from data. However, noisy and sparse data can have significant impact on the performance of neural networks and is a research topic in itself. In the context of PINNs, most studies assume mostly noise-free data.

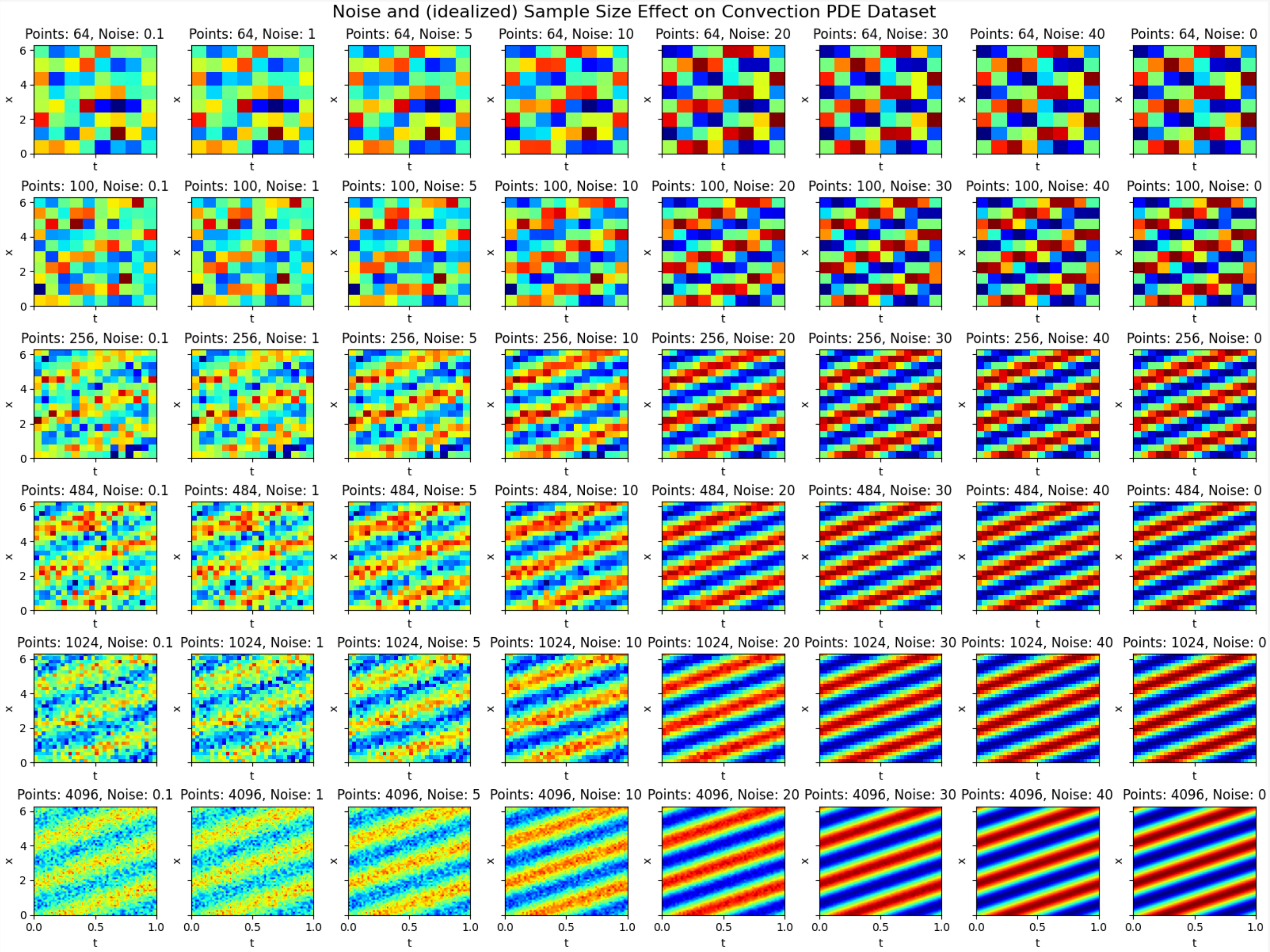
Research Question

Krishnapriyan, Aditi, et al. (2021) analyzed the performance of PINNs and introduced and showed improvements through curriculum learning.

Our aim is to build on their work, specifically on the curriculum learning approach, and investigate the effects of sampling size and noise in the training data on the performance of PINNs.

Visualizing the effect of sampling size and noise

(Note that the sample are idealized, as we assume that the samples are equidistantly distributed over the domain.)



The Classroom

Our first goal is to reproduce the results of Krishnapriyan, Aditi, et al. (2021) and verify against the baseline results, which does not use curriculum learning. Furthermore, we aim to find optimal hyperparameters for the optimizers used in the following experiments.

This is done by learning the Convection-Diffusion Equation and comparing the results of the loss function for a convection coefficient of $c = 30$.

Setup

Convection-Diffusion Equation

The PDE in the experiments is the Convection–Diffusion Equation with no diffusion term and a scalar convection term with a continuous boundary as well as a sine wave as the initial condition, given by

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

with c being the convection coefficient and $u(x, t)$ is the primitive function of the CDE.

Setup

Data Generation

The data is generated by random sampling 100 points from the domain $\Omega = [0, 1] \times [0, 2\pi]$ consisting of 10000 equidistant points. The corresponding labels are the analytical solution of the CDE. Additionally, the data is augmented with Gaussian noise to achieve a signal-to-noise ratio (SNR) of 50.

Setup

Model

Fully connected neural network with 3 hidden layers and 50 neurons per layer. The activation function is the hyperbolic tangent function. The input of the network is the spatial coordinate x and temporal coordinate t. The output of the network is $u(x, t)$.

Setup

Loss Function

The loss function is the mean squared error (MSE) between the predicted and the analytical solution of the CDE and the PDE of the learned function. The loss function is given by

$$\mathcal{L}(\hat{u}, u \mid \theta) = \mathcal{L}_{\text{PDE}}(\hat{u} \mid \theta) + \mathcal{L}_{\text{MSE}}(\hat{u}, u \mid \theta)$$

$$\mathcal{L}_{\text{PDE}}(\hat{u} \mid \theta) = \frac{\partial \hat{u}}{\partial t} + c\frac{\partial \hat{u}}{\partial x}$$

$$\mathcal{L}_{\text{MSE}}(\hat{u}, u \mid \theta) = \frac{1}{n}\sum_{i=1}^{n}(\hat{u}(x_i, t_i) - u(x_i, t_i))^2$$

where $\hat{u}$ is the predicted value of the CDE, u is the analytical solution of the CDE, θ are the learned weights and biases of the network.

Setup

Optimizer

Optimizer	Learning Rate	Weight Decay	Momentum	Nesterov	Max Iterations	History Size
SGD	[.0001, .1]	[0, .1]	[.1, .9]	Yes, No	-	-
Adam	[.00001, .1]	[0, .1]	-	-	-	-
L-BFGS	[.01, 2]	-	-	-	5, 10, 20	25, 50, 100

The hyperparameters for the optimizers are found using sweeps with Bayesian optimization on the hyperparameter search space.

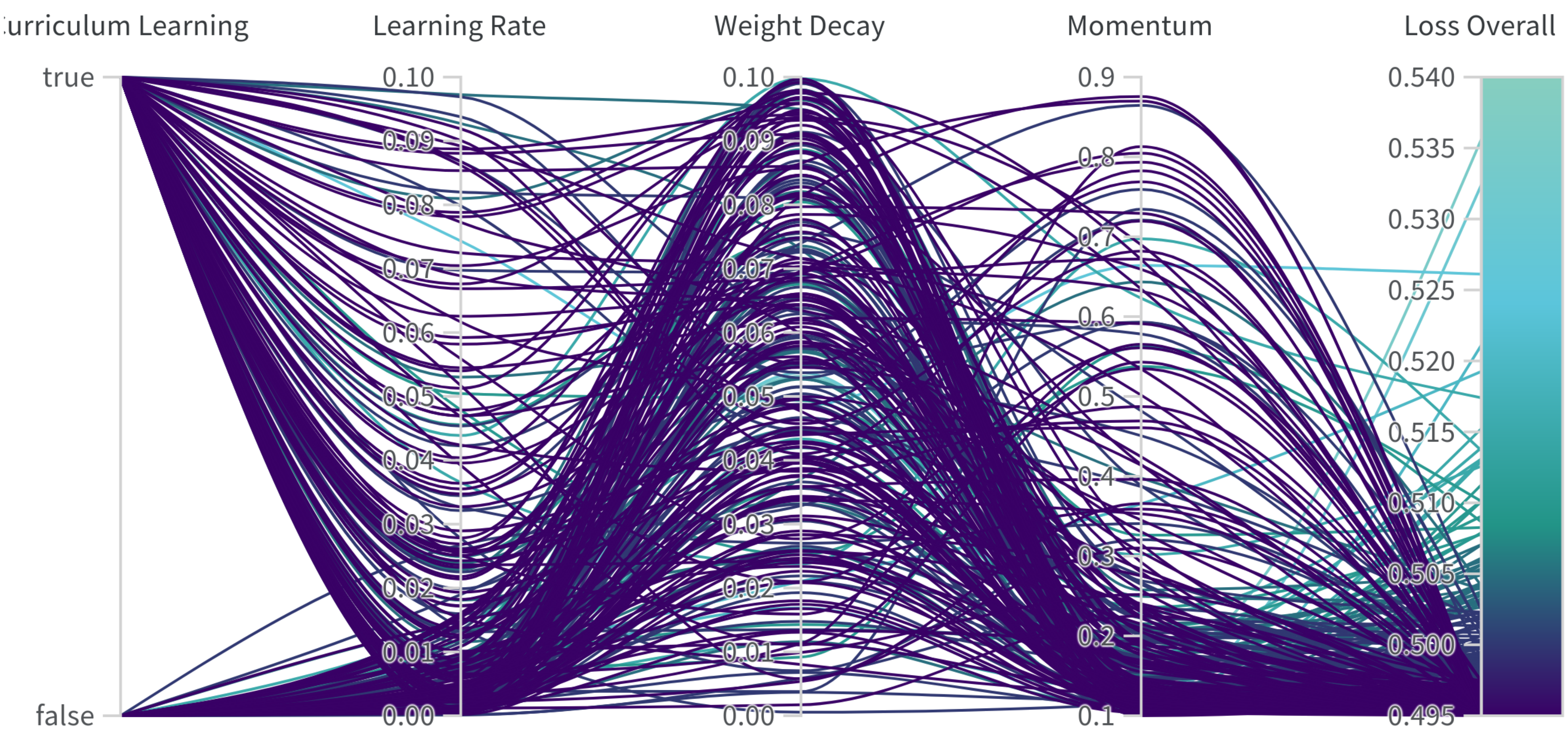
Setup

Curriculum Learning

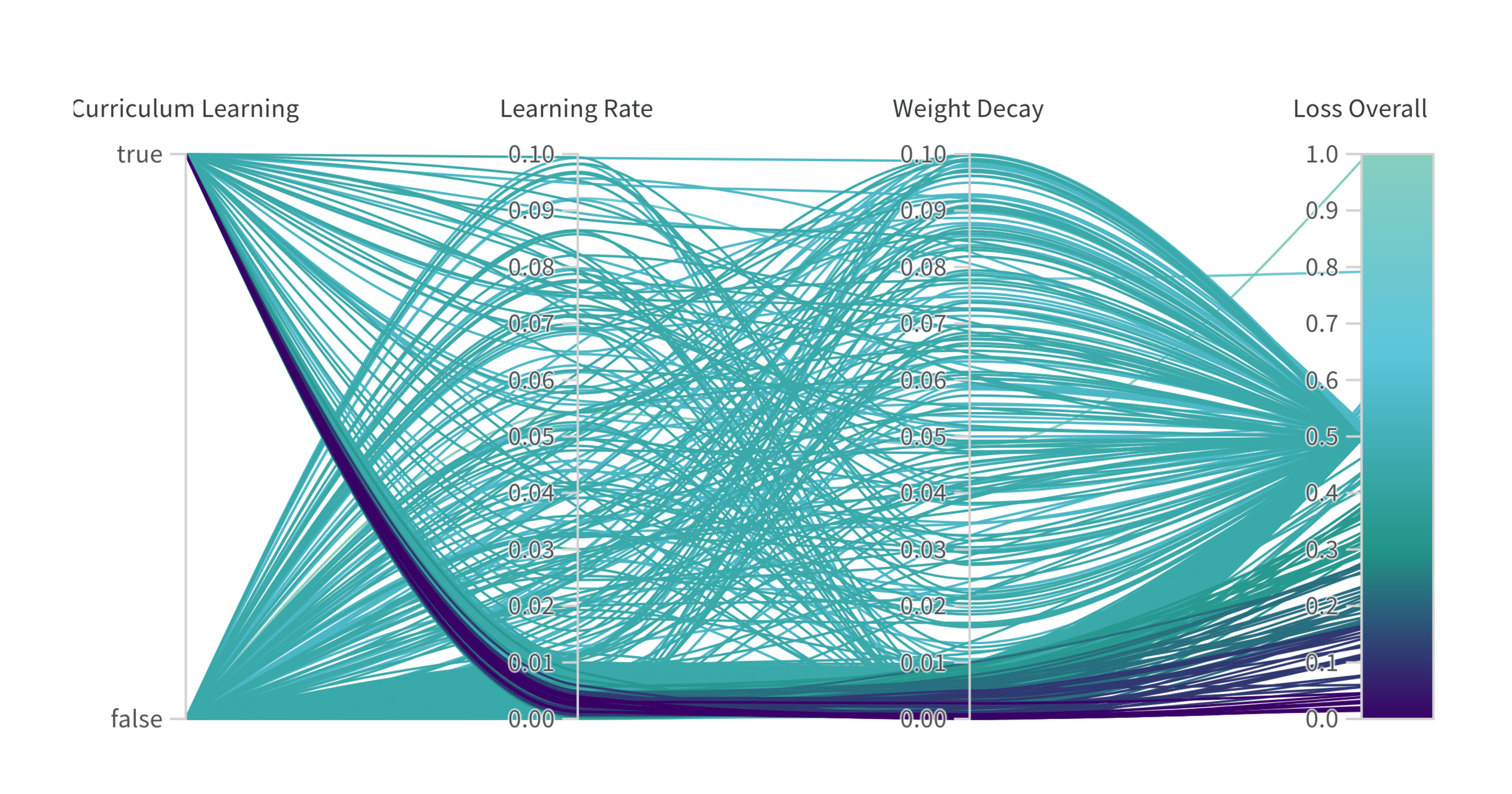
During the curriculum learning, the convection coefficient c is increased from 1 to 30 in 30 steps, meaning that during each step the convection coefficient is increased by 1. Each step consists of 250 epochs, 100 samples from the domain and the noise is fixed at a SNR of 50.

Results

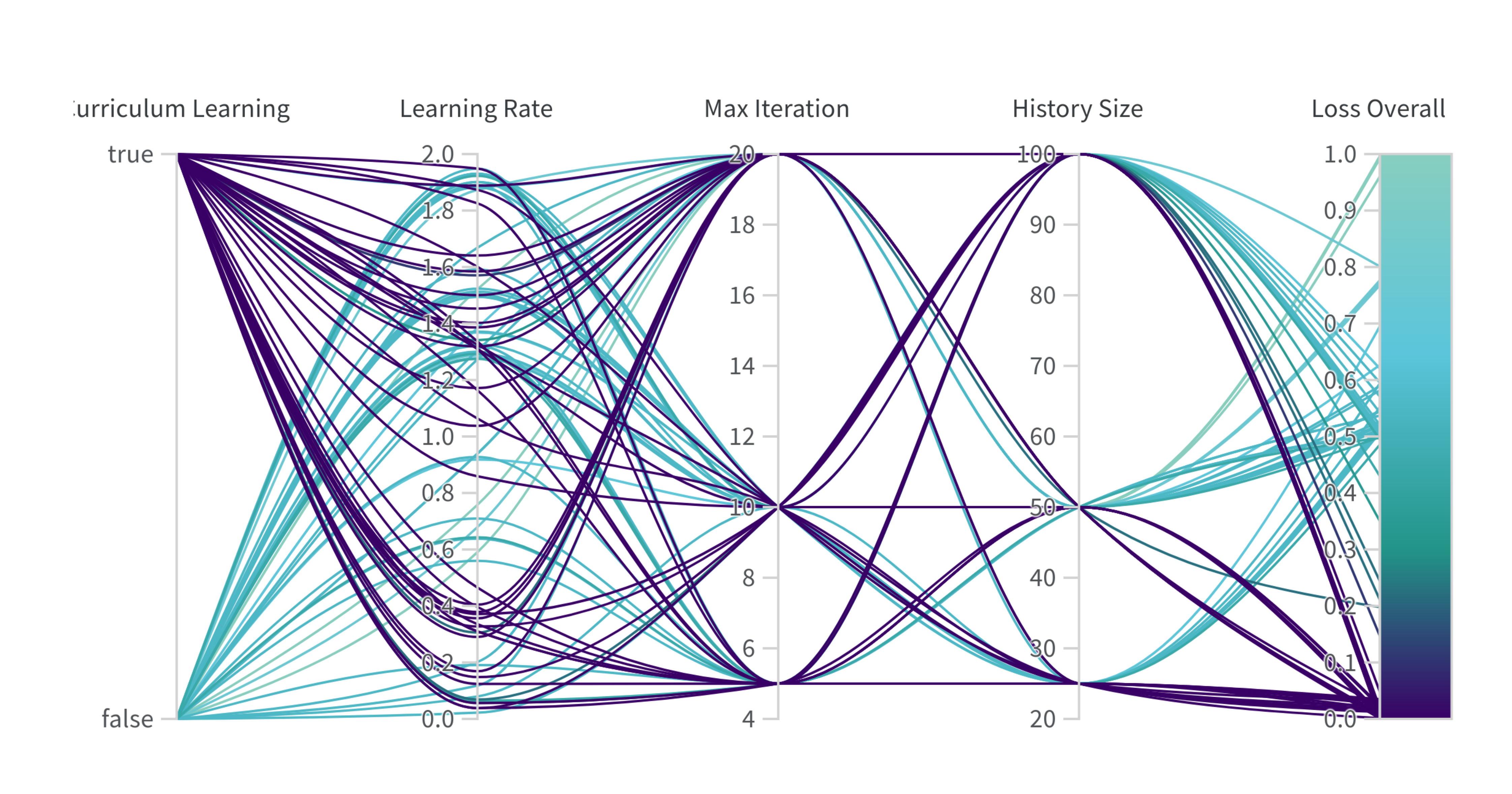
Result: Stochastic Gradient Descent Hyperparameter Sweep



Result: Adam Hyperparameter Sweep



Result: L-BFGS Hyperparameter Sweep



A Crowded and Noisy Classroom

The previous experiment is an idealized scenario, where the data is almost noise-free. However, our goal is to investigate the effects of sampling size and noise in the training data on the performance of PINNs. Therefore, using the optimal hyperparameters, we vary the sample size and noise to see how they affect the ability of PINNs to learn the CDE.

Setup Changes

Optimization

Optimizer	Learning Rate	Weight Decay	Max Iterations	History Size
Adam	.0025	.0005	-	-
L-BFGS$_{\mathrm{high}}$	1.5	-	10	25
L-BFGS$_{\mathrm{low}}$	.15	-	10	100

The hyperparameters for the optimizers are fixed to the optimal hyperparameters found in the previous experiment.

Setup Changes

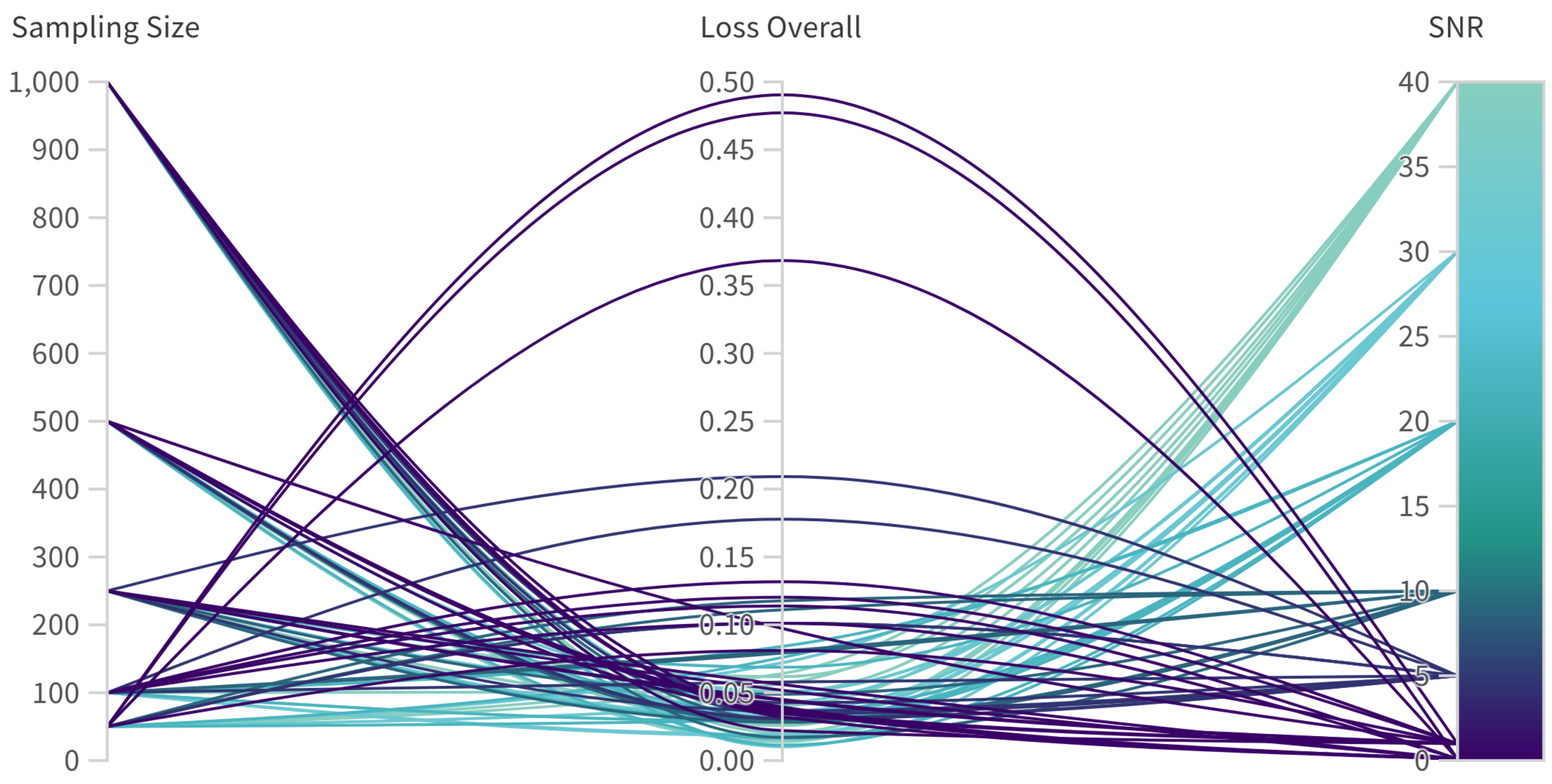
Sample Size

The sample size is varied between 10, 50, 100, 500 and 1000 samples and randomly sampled from the domain $\Omega = [0, 1] \times [0, 2\pi]$ consisting of 10000 equidistant points.

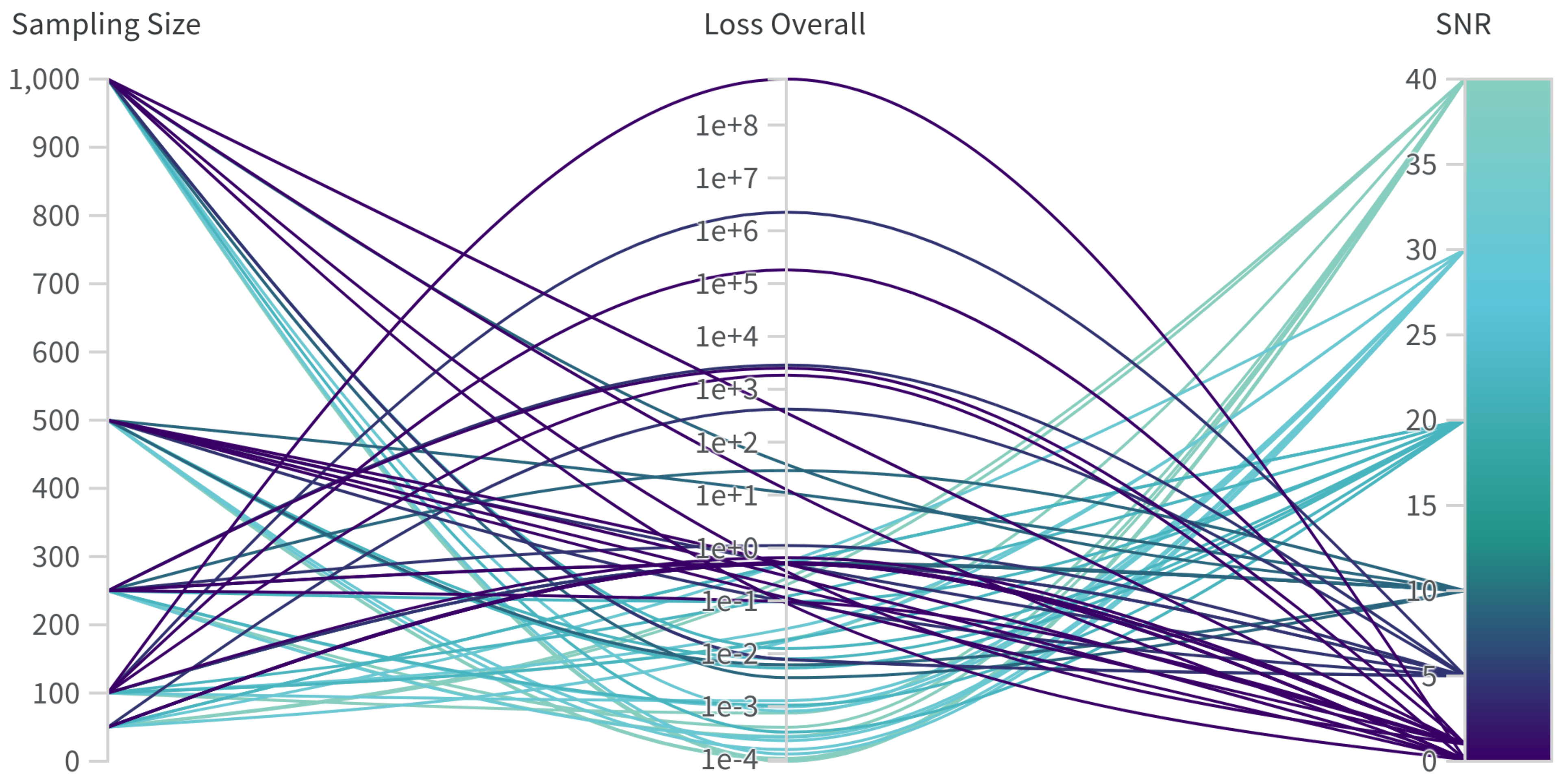
Noise

The signal-to-noise ratio (SNR) is varied from 0.1, 1, 5, 10, 20, 30 and 40 and added as Gaussian noise to the training data.

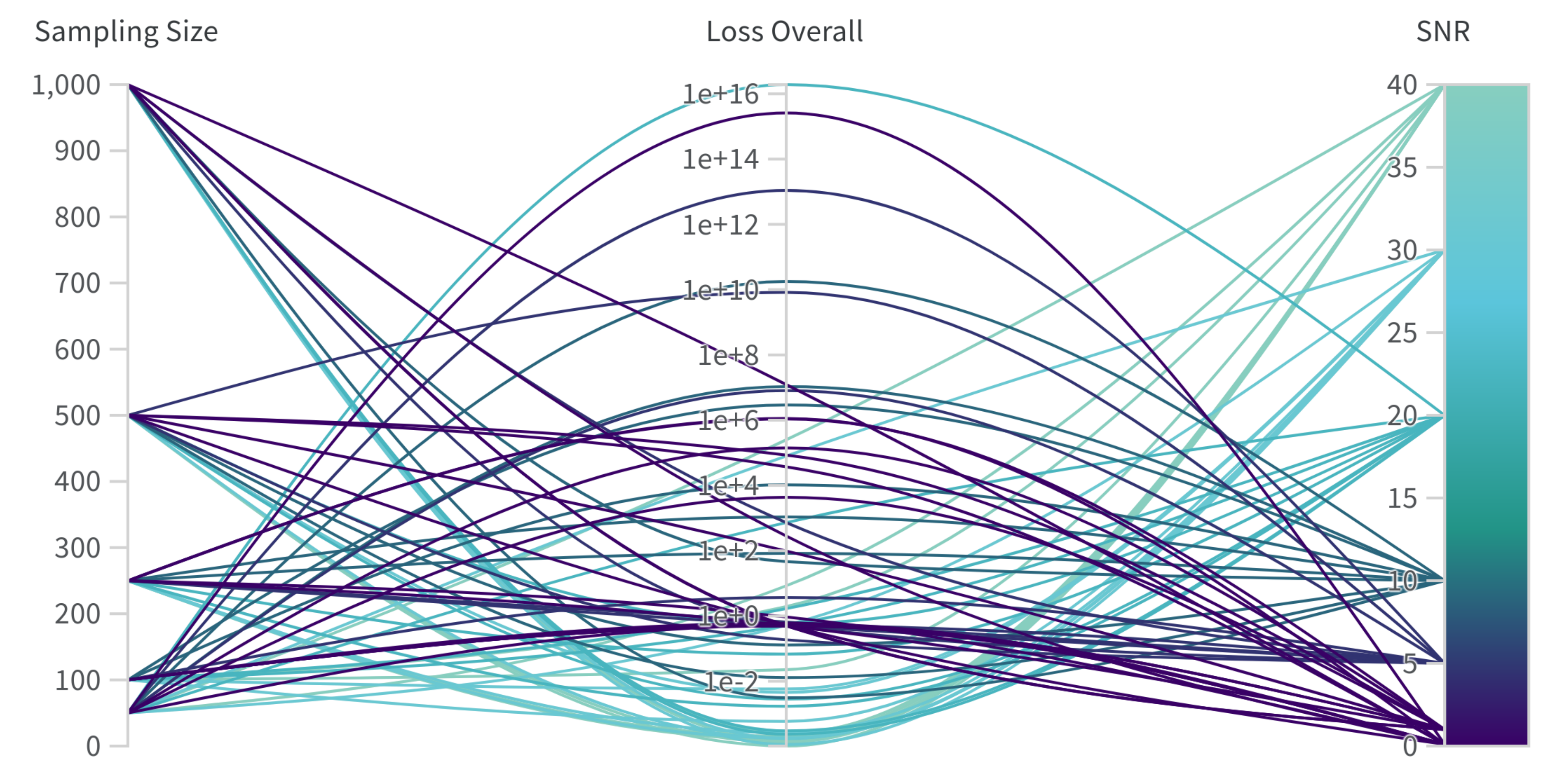
Results: Adam



Results: L-BFGS$_{\mathrm{high}}$



Results: L-BFGS$_{\text{low}}$



A Slower Approach on Learning the Subject

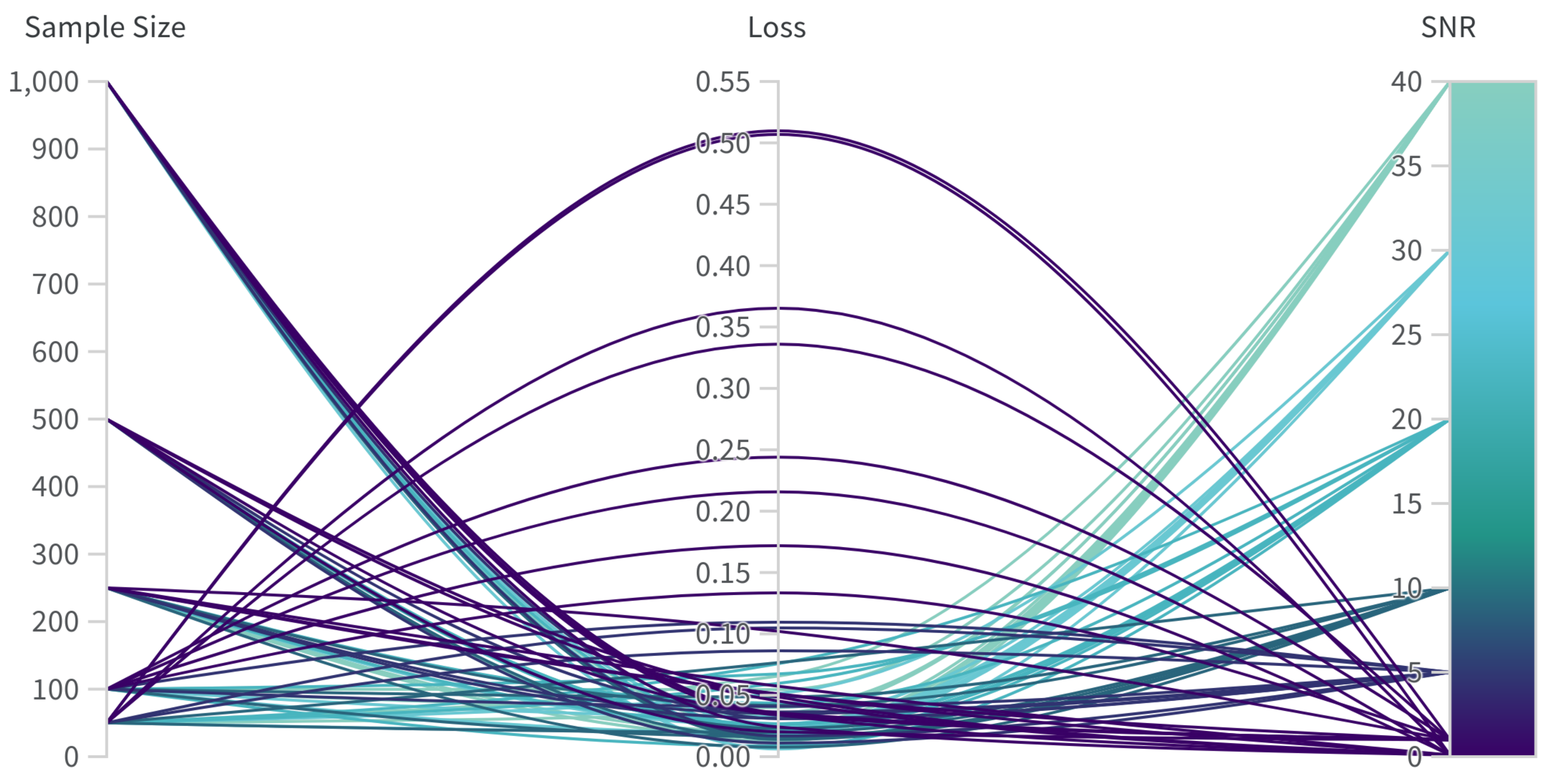
Introducing noise can have a negative effect on the performance, which can be reduced to some degree by increasing the sample size. Another approach to possibly reduce the negative effects of noise is to increase the number of curriculum steps.

Setup Changes

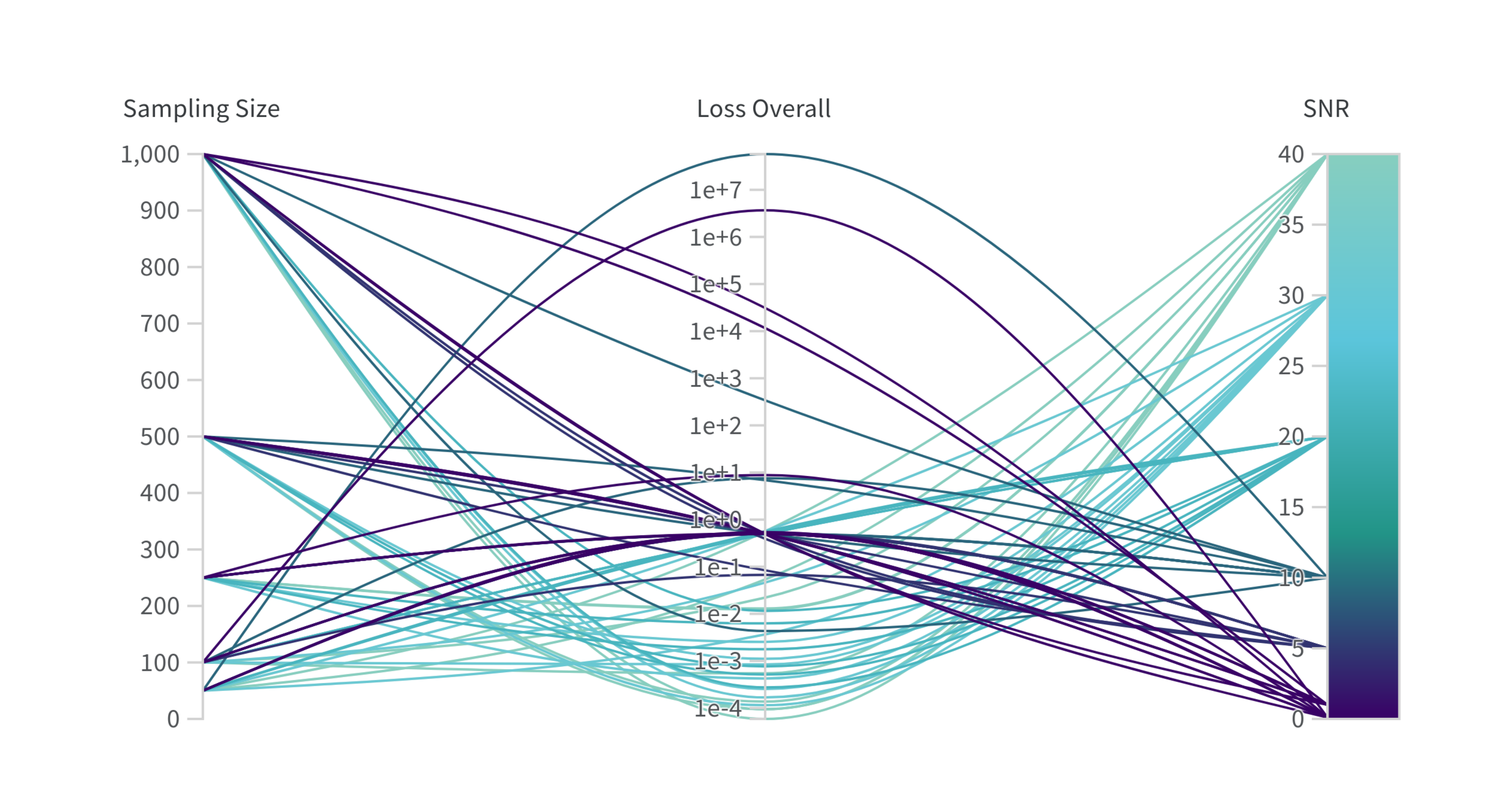
Curriculum Learning

The curriculum learning steps are increased from 30 to 60. The convection coefficient c is increased by 0.5 per curriculum step, starting at 0.5 and ending at 30.

Results: Adam



Results: L-BFGS$_{\mathrm{high}}$



Thank you for your attention!