A Multi-fidelity Prediction with Convolutional Neural Networks Using High-Dimensional Data[1]

H. Emre Tekaslan

April 24, 2023

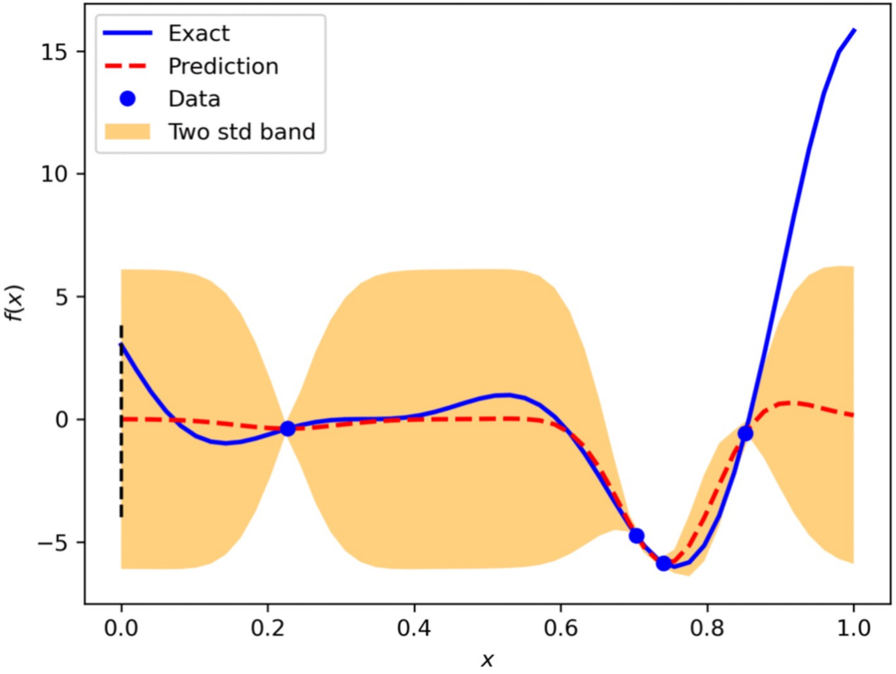
[1]Emre Tekaslan, H., and Nikbay, M., "Multifidelity Prediction Framework with Convolutional Neural Networks Using High-Dimensional Data," Journal of Aerospace Information Systems, Vol. 20, No. 5, 2023, pp. 264–275. https://doi.org/10.2514/1. I011159, URL https://doi.org/10.2514/1.I011159.

Outline

- Introduction
- Multi-fidelity Deep Neural Networks
- Modified Multi-fidelity Deep Neural Networks
- Multi-fidelity Convolutional Neural Networks
- Multi-fidelity Aerodynamic Prediction Using 2D Flow Fields
- Conclusion

Introduction

- Multi-fidelity (MF) = High-fidelity (HF) + Low-fidelity (LF)
- Low number of high-fidelity data, high number of low-fidelity data
- Predictions as accurate as high-fidelity data
- Relatively cheaper than high-fidelity data
- Highly appealed in design optimization and uncertainty quantification due to efficiency.
- Multi-fidelity deep neural networks[1] (MFDNN) was proposed recently.
- In this study, we propose two novel neural network architectures tailored for high-dimensional inputs.
- The proposed architectures outclasses the MFDNN in the high-dimensional aerodynamic problem.



[1]Meng, X., and Karniadakis, G. E., "A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems," Journal of Computational Physics, Vol. 401, 2020, p. 109020.3

Multi-fidelity Deep Neural Networks

- It makes both LF and HF predictions with a model.
- An MFDNN model consists of 3 sub-networks
  - **NN_L** : Low-fidelity estimator
  - **NN_{H1}** : Linear correlation network
  - **NN_{H2}** : Nonlinear correlation network
- **NN_L** takes low-fidelity inputs and makes low-fidelity predictions.
- Correlation networks take the stacked high-fidelity input and low-fidelity predictions.
- **NN_{H1}** approximates the linear correlation between HF and LF data.
- **NN_{H2}** approximates the nonlinear correlation between HF and LF data.

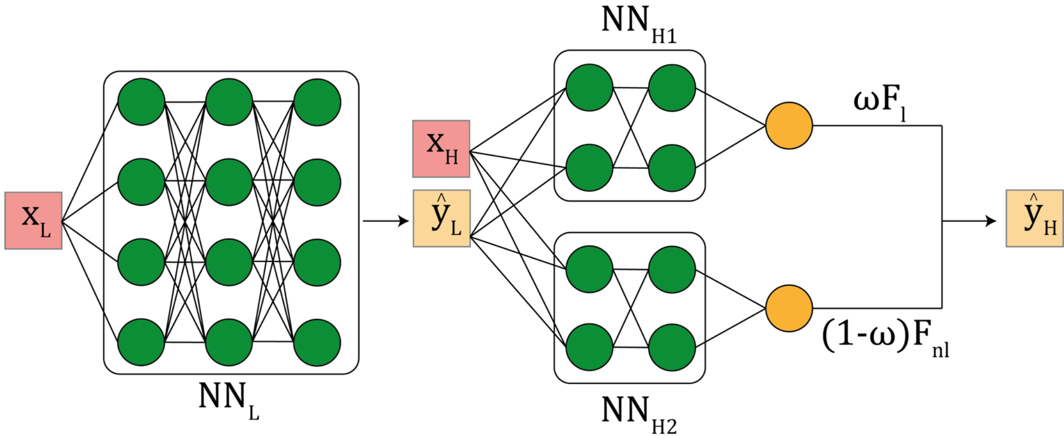


Figure: A generic MFDNN architecture.

Multi-fidelity Deep Neural Networks

- Autoregressive scheme[1]:

$$y_H(x, y_L) = \omega \mathcal{F}_l(x, y_L) + (1-\omega)\mathcal{F}_{nl}(x, y_L)$$

- $NN_{H1} = \mathcal{F}_l(x, y_L)$, $NN_{H2} = \mathcal{F}_{nl}(x, y_L)$
- NN_{H1} only consists of linear layers because it is supposed to learn the linear correlation.
- NN_{H2} consists of the combination of linear and nonlinear layers.
- MFDNNs work very well with low-dimensional inputs and predictions.

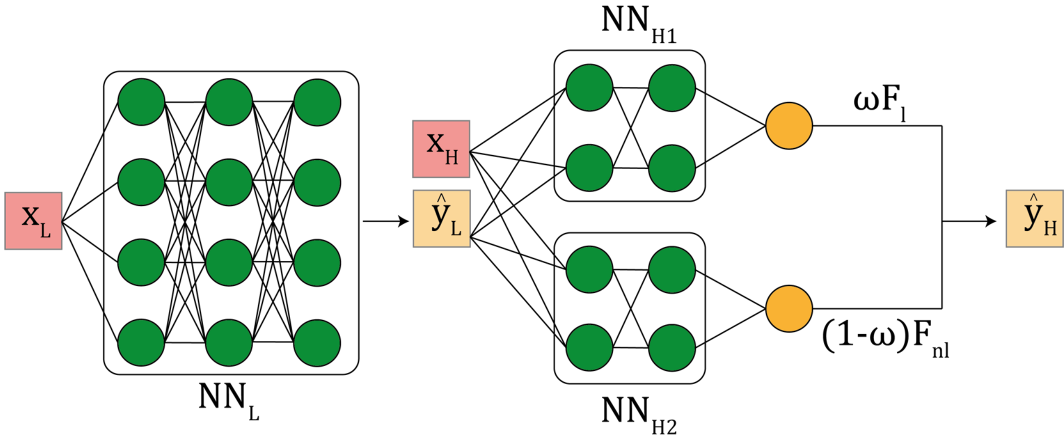


Figure: A generic MFDNN architecture.

[1]Meng, X., and Karniadakis, G. E., “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems,” Journal of Computational Physics, Vol. 401, 2020, p. 109020.5

Multi-fidelity Deep Neural Networks – Drawback 1

- The input of correlation networks use the concatenated high-fidelity input $\mathbf{x_H} \in \mathbb{R}^{N_H}$ and low-fidelity predictions $\hat{\mathbf{y}}_\mathbf{L} \in \mathbb{R}^{M_L}$.

$$\mathbf{x_C} \in \mathbb{R}^{N_H+M_L}$$

- If the input size N_H is larger enough than the prediction size M_L ($M_L << N_H$), the prediction is ignored during training.

- This is an issue in case of high-dimensional inputs such as flow fields.

$$y_H(x, y_L) = \omega \mathcal{F}_l(x, y_L) + (1-\omega)\mathcal{F}_{nl}(x, y_L)$$

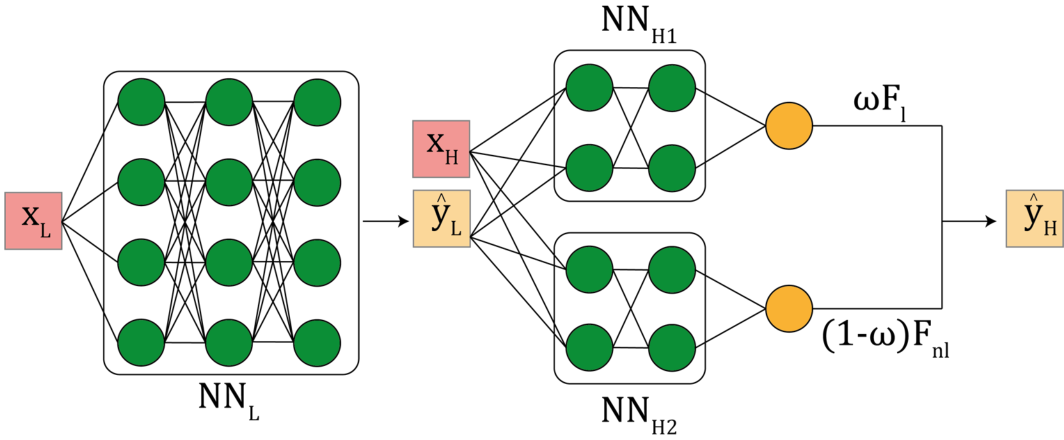


Figure: A generic MFDNN architecture.

The Modified-MFDNN

As a remedy of the first drawback,

- We adapted the MFDNN by adding a fully-connected encoder.
  - **NN_{XH} :** A fully-connected encoder
- NN_{XH} contains only linear layers and maps high-dimensional inputs onto lower-dimensional subspaces.
- Thus, input sizes get closer to the low-fidelity predictions $\hat{\mathbf{y}}_{\mathbf{L}} \in \mathbb{R}^{M_L}$.

$$NN_{XH} : \mathbb{R}^{N_H} \to \mathbb{R}^{\tilde{N}_H}$$

$$\mathbf{x_H} \in \mathbb{R}^{N_H} \longrightarrow \tilde{\mathbf{x}}_{\mathbf{H}} \in \mathbb{R}^{\tilde{N}_H}$$

$$\tilde{N}_H \approx M_L << N_H$$

- This architecture improves the multi-fidelity prediction.

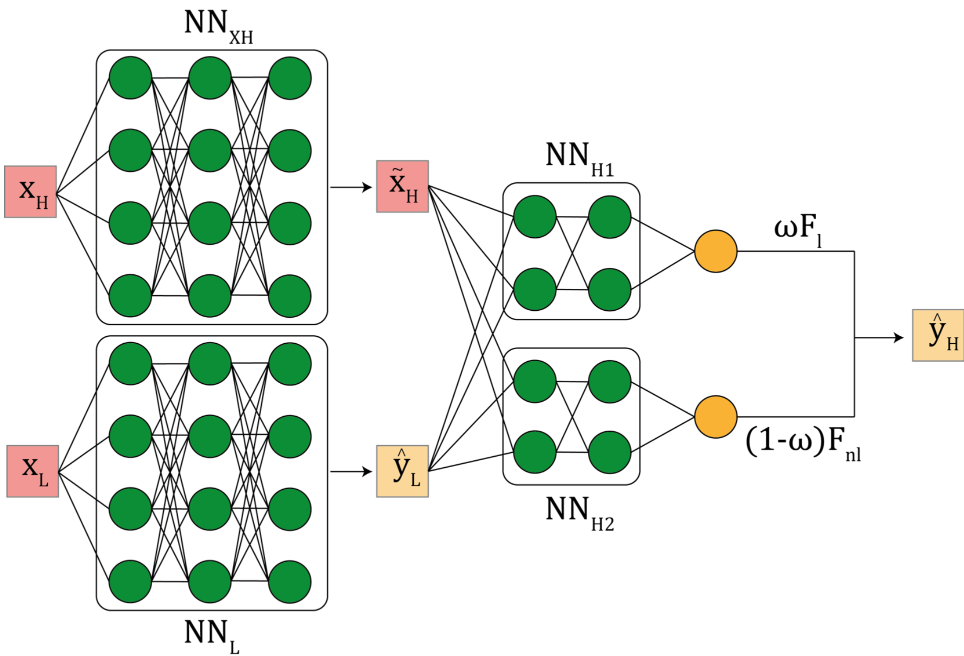


Figure: A generic modified-MFDNN architecture.

Multi-fidelity Deep Neural Networks – Drawback 2

- In addition, MFDNN can only process vector or scalar inputs.
- Flow fields compose highly correlated regions and are represented with matrix/tensor notation.
- Thus, each flow field must be vectorized to be processed by an MFDNN.
- Vectorization dislocates highly correlated vertices on a flow field.
- Thus, it causes a loss of correlation information on the data.

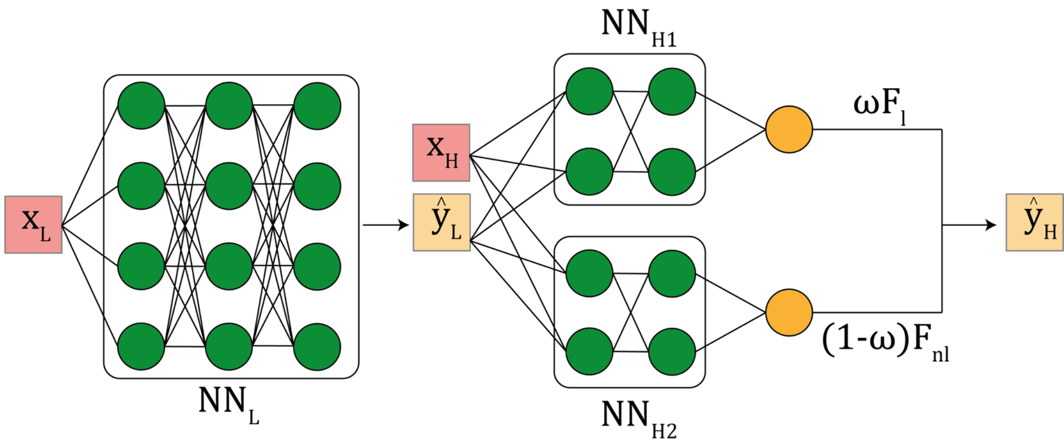
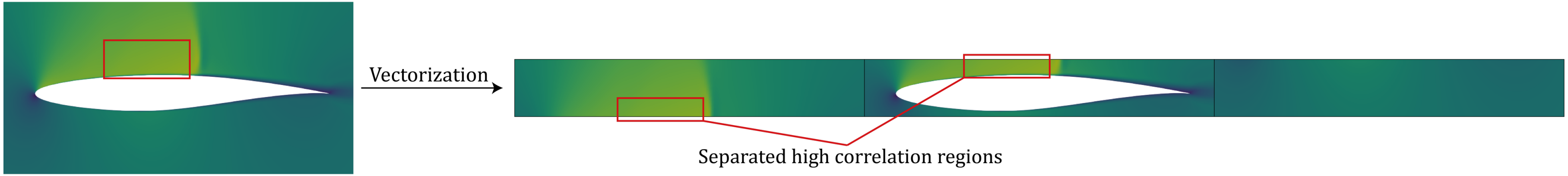


Figure: A generic MFDNN architecture.



Multi-fidelity Convolutional Neural Networks

As a remedy of the second drawback,

- We employed convolution layers to process inputs within a matrix/tensor form.
- Vectorization is prevented so does the loss of information.
- It also includes a convolutional encoder, $\mathbf{NN_{XH}}$.
  - $\mathbf{NN_{XH}}$: Convolutional encoder
- NN_{XH} contains only convolution layers and maps high-dimensional inputs onto lower-dimensional subspaces similar to the modified-MFDNN.
- Unlike previously presented methods, low-fidelity estimator, $\mathbf{NN_L}$, is a convolutional neural network.

$$y_H(x, y_L) = \omega \mathcal{F}_l(x, y_L) + (1-\omega)\mathcal{F}_{nl}(x, y_L)$$

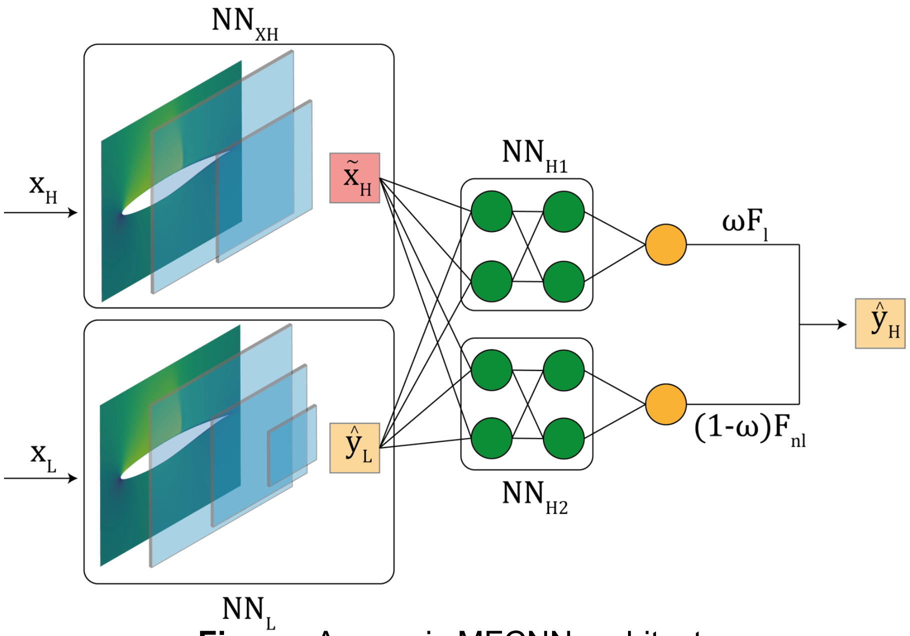


Figure: A generic MFCNN architecture.

Pre-processing Data

- Pressure coefficient fields are used as the input.
- Each flow domain is interpolated onto 64-by-64 Cartesian grid using linear interpolation.
- Interpolation makes geometry obscure but preserves the gradients on the flow field well!
- Min-max normalization is used on the dataset.
- Low-fidelity input data $\mathbf{x}_L \in \mathbb{R}^{300\times64\times64}$
- High-fidelity input data $\mathbf{x}_H \in \mathbb{R}^{100\times64\times64}$
- 85%-15% training and test split ratio is used.

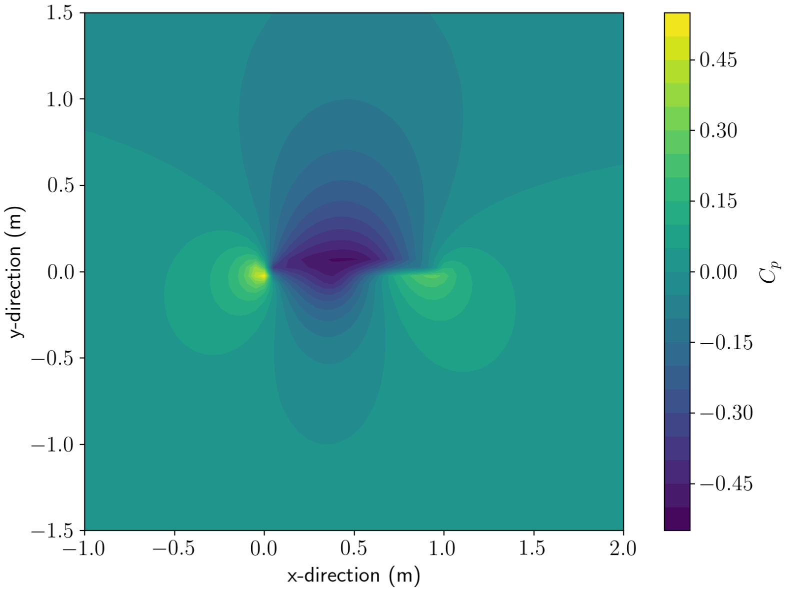


Figure: An interpolated flow domain.

Predictions with the MFDNN

- MFDNNs with 3 set of learnable parameters are tested.
  - Parameter sizes: 2.4e6, 4.6e6, 10.9e6
- Linear layers and rectified linear units are employed.
- Pressure coefficient fields are vectorized.

$$\tilde{\mathbf{x}}_L \in \mathbb{R}^{300\times 4096} \qquad \tilde{\mathbf{x}}_H \in \mathbb{R}^{100\times 4096}$$

- Increasing high-fidelity sample size, NS_H, is used.
- All MFDNNs yields comparable and unsatisfactory results in terms of MF-RMSE.
- Higher number of high-fidelity data does not affect the prediction accuracy!
- It can be concluded that the MFDNN is not able to learn the correlation between the HF and LF data.

N_θ	NS_H	**LF-RMSE**	**MF-RMSE**
	10	0.0177	0.0458
2.4×10^6	50	0.0177	0.0483
	100	0.0177	0.0477
	150	0.0177	0.0296
	200	0.0177	0.0473
	10	0.0142	0.0447
4.6×10^6	50	0.0142	0.0481
	100	0.0142	0.0475
	150	0.0142	0.0330
	200	0.0142	0.0480
	10	0.0177	0.0449
10.9×10^6	50	0.0177	0.0484
	100	0.0177	0.0484
	150	0.0177	0.0322
	200	0.0177	0.0486

Predictions with the modified-MFDNN

- The modified-MFDNNs with 3 different latent vector sizes are investigated.
  - $\tilde{N}_H$: 4,8,16
- Linear layers and rectified linear units are employed.
- Increasing high-fidelity sample size is used.
- The lower the $\tilde{N}_H$ is, the better multi-fidelity predictions we get!
- Higher number of high-fidelity data does not affect the prediction accuracy!
- Thus, compressing high-fidelity inputs causes the loss of information.
- The modified-MFDNN outperforms the MFDNN up to 66% .

$N_{\tilde{H}}$	NS_H	LF-RMSE	MF-RMSE	MF-RMSE Improvement (%)
4	[10,200]	0.0155	0.0162	66.0
8	[10,200]	0.0208	0.0215	50.8
16	[10,200]	0.017	0.0240	45.1

Predictions with the MFCNN

- A single MFCNN model is constructed.
- Convolution layers and rectified linear units are employed.
- Higher number of high-fidelity data improves the prediction accuracy!
- The MFCNN is superior to the previously presented methods for the considered case.
- The MFCNN improves the multi-fidelity predictions up to 78% in comparison of the MFDNN.
- It can also improves the multi-fidelity predictions up to 38% when compared to the modified-MFDNN.

NS_H	LF-RMSE	MF-RMSE	MF-RMSE Improvement (%)	
			MFDNN vs. MFCNN	Modified-MFDNN vs. MFCNN
10	0.0075	0.0225	50.9	-42.4
50	0.0059	0.0113	76.5	30.0
100	0.0060	0.0105	77.9	35.0
150	0.0065	0.0100	66.2	38.3
200	0.0075	0.0101	78.7	37.7

Comparison

- MFCNN is the fastest learning method.
- MFCNN is the best in learning.
- MFCNN is the best in making multi-fidelity predictions.

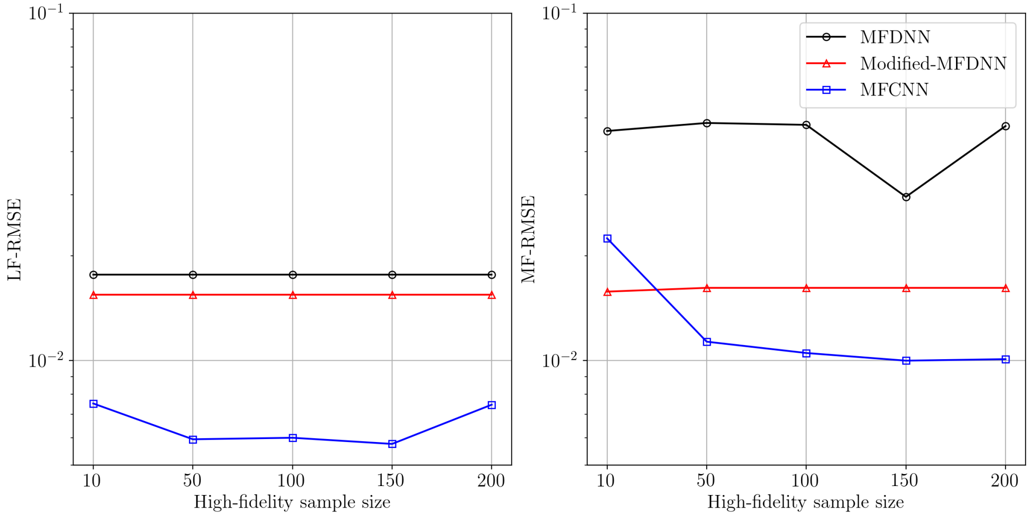


Figure: The comparison of RMSE on the test data.

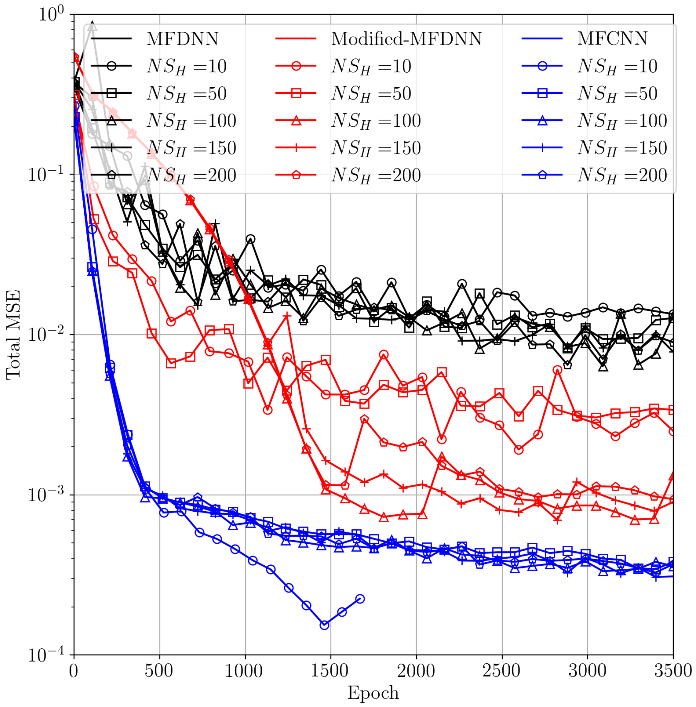
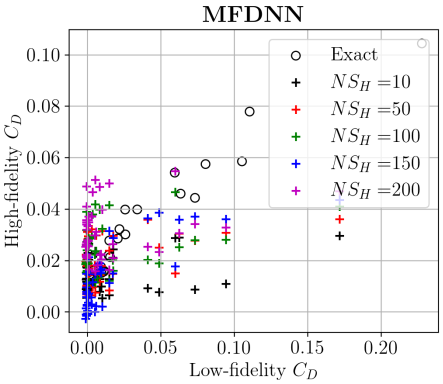
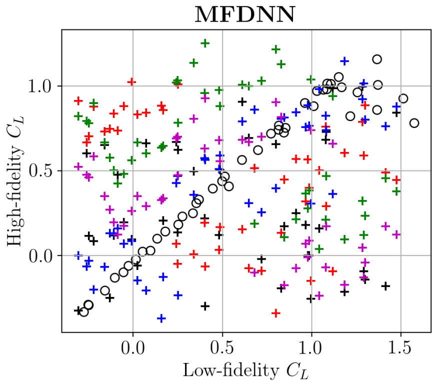


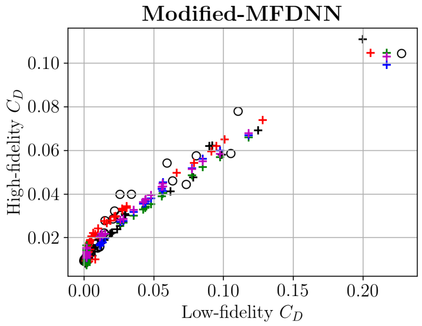
Figure: Learning curves (training loss).

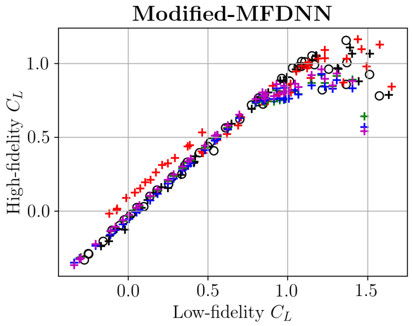
Conclusion

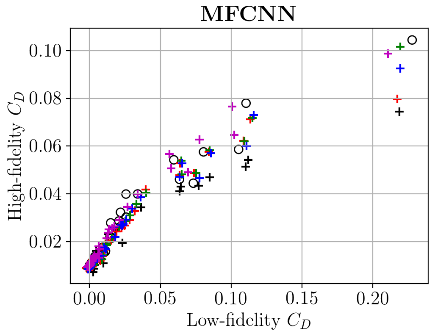
- MFDNN couldn't handle high-dimensional inputs with low-dimensional predictions.
- The modified-MFDNN enhances the MFDNN significantly in terms of multi-fidelity predictions.
- The MFCNN proves its efficacy on high-dimensional inputs.

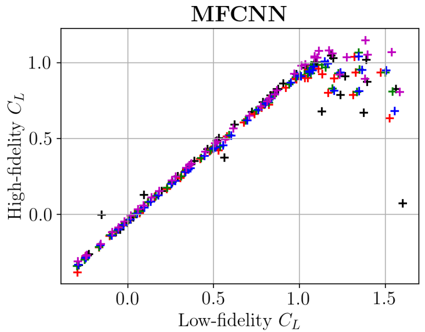












Thank you for listening!