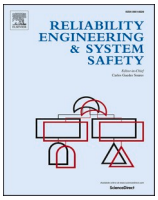
Reliability Engineering and System Safety 245 (2024) 109975

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress



Aleatory uncertainty quantification based on multi-fidelity deep neural networks

Zhihui Li *, Francesco Montomoli

UQ Laboratory, Department of Aeronautics, Faculty of Engineering, Imperial College London, London, SW7 2AZ, UK

A R T I C L E I N F O

Keywords:
Uncertainty quantification
Aleatory uncertainty
High dimension
Multi-fidelity
Deep neural networks
Turbine flow

A B S T R A C T

Traditional methods for uncertainty quantification (UQ) struggle with the *curse of dimensionality* when dealing with high-dimensional problems. One approach to address this challenge is to leverage the potent approximation capabilities of deep neural networks (DNNs). However, conventional DNNs often demand a substantial amount of high-fidelity (HF) training data to ensure precise predictions. Unfortunately, the availability of such data is limited due to computational or experimental constraints, primarily driven by associated costs. To mitigate these training expenses, this research introduces multi-fidelity deep neural networks (MF-DNNs), wherein a sub-network is constructed to simultaneously capture both linear and non-linear correlations between HF- and low-fidelity (LF) data. The efficacy of MF-DNNs is initially demonstrated by accurately approximating diverse benchmark functions. Subsequently, the developed MF-DNNs are employed for the first time to simulate the aleatory uncertainty propagation in 1-, 32-, and 100-dimensional contexts, considering either uniform or Gaussian distributions of input uncertainties. The UQ results affirm that MF-DNNs adeptly predict probability density distributions of quantities of interest (QoI) and their statistical moments without significant compromise of accuracy. Furthermore, MF-DNNs are applied to model the physical flow inside an aircraft propulsion system while accounting for aleatory uncertainties originating from experimental measurement errors. The distributions of isentropic Mach number are accurately predicted by MF-DNNs based on the 2D Euler flow field and few experimental data points. In conclusion, the proposed MF-DNN framework exhibits significant promise in addressing UQ and robust optimization challenges in practical engineering applications, particularly when dealing with multi-fidelity data sources.

Nomenclature

CO-KRG	co-Kriging model
DNNs	deep neural networks
FCNNs	fully connected neural networks
GPU	graphics processing unit
HF	high-fidelity
KRG	Kriging model
LF	low-fidelity
MF	multi-fidelity
MF-DNNs	multi-fidelity deep neural networks
MF_DNN_Comp	composite multi-fidelity deep neural networks
MF_DNN_ReLU	multi-fidelity deep neural networks with ReLU activation function
MISES	multiple blade interacting streamtube Euler solver
MSE	mean square error
NNs	neural networks
QoI	quantities of interest
RBF	radial basis function
UQ	uncertainty quantification
"ReLU"	rectified linear unit activation function
"Sigmoid"	logistic sigmoid function
"Tanh"	hyperbolic tangent function
H	number of hidden layers
M	number of LF training points
N	normal (Gaussian) distribution
P	number of HF training points
U	uniform distribution
L_c	overall mean square error of the Correction DNNs
L_h	training loss of Correction DNNs
L_l	training loss of LF-DNNs
L_r	L_2 regularization loss of Correction DNNs

* Corresponding author.
E-mail addresses: zhihui.li@imperial.ac.uk (Z. Li), f.montomoli@imperial.ac.uk (F. Montomoli).

https://doi.org/10.1016/j.ress.2024.109975
Received 10 November 2023; Received in revised form 15 January 2024; Accepted 26 January 2024
Available online 28 January 2024

Q_{LF}	low-fidelity data set
Q_{HF}	high-fidelity data set
X_{LF}	low-fidelity variables
X_{HF}	high-fidelity variables
b_h	biases of hth layer
$\widehat{y}_{HF}$	prediction values on HF data points
y_{LF}	label values of LF data points
θ_{NN}	neural network parameters
w_h	weights of hth layer
$\mathscr{F}$	correlation between LF- and HF data
T	temperature
λ	control parameter of L_2 regularization loss
c	multiplicative correction surrogate
δ	additive correction surrogate
ρ	air density
t	Time
p	Pressure
μ	dynamic viscosity
e	internal energy
κ	thermal conductivity
$\boldsymbol{u}$	flow velocity vector
∇	differential operator

1. Introduction

In practical scenarios, aleatory uncertainties emerge from inherent variability and randomness within systems, often presenting a high level of dimensionality. These uncertainties encompass a wide array of factors, for instance, the fluctuations in boundary/initial conditions [1], uncertain geometries due to manufacturing vibrations [2], harsh operating environments [3] and inherently uncertain model parameters [4]. Moreover, the experimental uncertainty [5] stemming from measurement errors and the variability of material properties [6] contribute to the spectrum of aleatory uncertainties. However, traditional uncertainty quantification (UQ) techniques like Monte Carlo (MC), polynomial chaos expansion (PCE) and Taylor series expansion-based methods face challenges posed by the *curse of dimensionality* when addressing these high-dimensional aleatory uncertainties [7–8]. Substantial volumes of high-fidelity (HF) training data are required to attain acceptable levels of predictive accuracy for these methods. However, in engineering applications, generating enough HF results often means computationally or experimentally prohibitive costs.

To mitigate this reliance on HF outcomes, multi-fidelity (MF) approaches have garnered substantial interest, harnessing a collection of HF- and low-fidelity (LF) data sets [9–12]. Leveraging the availability and cost-effectiveness of LF data, the core premise behind MF methodology is to enhance model prediction accuracy through an augmented influx of HF data [13]. Literature has proposed diverse strategies, including MF response surface models [14–15], MF artificial neural networks [16–17] and MF- or adaptive Gaussian processes [18–22], to establish surrogate models. Similarly, for MF UQ methods, the realm has seen the introduction of MF PCE [23–24], MF MC approaches [25–26] and MF PCE-Gaussian process [27] in prior research endeavours. However, the aforementioned methodologies often demonstrate limitations in effectively capturing the cross-correlation between HF- and LF data. For instance, in our experience, the selection of HF point locations plays a pivotal role in influencing the precision of MF response surface models. Furthermore, MF Gaussian process regression encounters challenges in hyperparameter optimization with sparse data [28] and in tackling high-dimensional problems [29]. The effectiveness of the PCE technique heavily relies on the regularity between Quantities of Interest (QoI) and input uncertainties, and its efficacy might wane in the absence of such regularity, such as in stochastic hyperbolic problems [30]. Reference [31] is recommended for a comprehensive overview of MF methods in UQ, statistical inference, and optimization.

In recent times, the applications of deep learning techniques have found broad utility in modelling physical systems [32–33], as well as in fields like communications [34] and biological systems [35], primarily due to their versatile capacity for universal function approximation. Thus, the advent of MF neural networks (NNs) presents a promising avenue, particularly due to the fact that compared to traditional function approximators, deep neural networks (DNNs) are acknowledged as universal approximators across both low- and high-dimensional contexts [36]. Noteworthy instances of the applications of MF neural networks include the work by Lu and Zhu [37], who employed single fully connected neural networks (FCNNs) as bi-fidelity surrogate models to estimate HF proper orthogonal decomposition coefficients within a reduced-order model framework. Yan and Zhou [38] crafted an adaptive surrogate model by amalgamating the MF approach with NNs to tackle Bayesian inverse problems within a Markov chain MC framework. Conti et al. [39] introduced MF long short-term memory networks for addressing time-dependent problems. Dhulipala [40] utilized DNNs as an LF model and a separated Kriging model for the correction. Motamed [41] developed two distinct NNs based on bi-fidelity training data. Specifically, the first NN aimed to approximate the correction function between LF- and HF data, with its output serving as supplementary HF training data for the second NNs. Here, the precision of the artificially introduced HF training data significantly influenced the overall NNs' prediction accuracy.

In a more straightforward approach, Meng and Karniadakis [42] established composite NNs where the LF model was initially trained, and the LF model's output was then refined based on HF data. The proposed multi-fidelity deep neural networks (MF-DNNs) comprise three sub-networks: one for LF prediction, one for HF prediction assuming a linear correlation between LF- and HF data, and another for HF prediction relying on non-linear correlation. Results demonstrated the composite NNs' proficiency in approximating benchmark functions. The utilization of these MF-DNNs in aerodynamic optimization was exemplified in the work by Zhang et al. [43]. Ahn et al. [44] utilized the reduced-order model to update the low-fidelity data within the framework of composite neural networks. Meanwhile, Guo et al. [16] conducted MF regression using NNs with diverse architectures and compared predictive accuracy against the Co-Kriging (Co-KRG) model. Guo's study included the creation of a shallow NN layer to approximate the correlation between LF- and HF data. Nonetheless, results indicated that its capability in modelling linear correction functions slightly diminished when contrasted with the parallel sub-networks structure. Furthermore, the performance in predicting high-dimensional problems remained undisclosed.

To overcome the requirement of parallel sub-networks structure proposed in previous work, this paper introduces a network architecture featuring a single correction sub-network to achieve a balanced capacity of modelling both linear and nonlinear corrections. This architectural choice brings forth several appealing attributes: 1) eliminating assumption bias: the need to allocate weights to separate linear and nonlinear sub-networks is circumvented, ensuring a more neutral and adaptive modelling approach. 2) alignment with mathematical principles: the proposed architecture closely aligns with the foundational mathematical principles underpinning the MF methodology that serves as its basis. 3) enhanced simplicity: the streamlined architecture simplifies both the training and testing processes of the MF-DNNs, particularly when tackling high-dimensional problems. 4) ease of implementation: the implementation of the new MF-DNN architecture in programming is notably more straightforward. To the best of the authors' knowledge, this marks the pioneering attempt to address high-dimensional aleatory UQ problems using MF-DNNs. The structure of the paper unfolds as follows: Section 2 delves into the intricate details of the proposed MF-DNNs, followed by Section 3 which showcases the outcomes of surrogate modelling and UQ for various benchmark cases. Section 4 concludes this research endeavour by presenting the key

findings. Additionally, Appendix A adds a comparison of different activation functions in terms of their approximation capabilities.

2. Multi-fidelity deep neural networks

One widely used comprehensive correction [13] in bridging LF- and HF data is:

$$\widehat{y}_{HF} = c(\mathrm{X})\cdot y_{LF}(\mathrm{X}) + \delta(\mathrm{X}) \tag{1}$$

where $\widehat{y}_{HF}$ represents the model prediction values on HF data points, c is the multiplicative correction surrogate, y_{LF} represents the label values of LF data points, δ means the additive correction surrogate. Here the multiplicative correction c could be either a constant [45] or the non-constant value [46], which represents the linear or non-linear correction between $\widehat{y}_{HF}$ and y_{LF}. In other words, the comprehensive correction in Eq. (1) can be expressed as:

$$\widehat{y}_{HF} = \mathscr{F}(y_{LF}(\mathrm{X}), \mathrm{X}) \tag{2}$$

where $\mathscr{F}$ can comprehensively represent both the non-linear and linear correlation between the LF- and HF data. Thus, the idea for the proposed MF-DNNs is that it should consist of two sub-networks, one for the LF-DNNs to approximate the values of y_{LF}, and one for the Correction DNNs to predict the $\widehat{y}_{HF}$ based on the Eq. (2).

The architecture of the proposed MF-DNNs is shown in Fig. 1. Here we assume that there is a large set of LF training data $y_{LF}(\mathrm{X}_{LF})$, $\mathrm{X}_{LF} \in \mathbb{R}$ and a relatively small set of HF training data $y_{HF}(\mathrm{X}_{HF})$, $\mathrm{X}_{HF} \in \mathbb{R}$, $\mathrm{X}_{HF} \subset \mathrm{X}_{LF}$. The cost function was set to be the mean square error (MSE) between the prediction results and the actual values. The gradient information of the cost function to the network parameters could be obtained based on automatic differentiation, and the prediction error of the MF-DNNs was minimized by using gradient-based optimization algorithms. In detail, the ADAM and L-BFGS optimizers [47–48] were deployed here because of their better generalizing performance when compared to the other optimization methods [49–50]. To avoid overfitting, the L_2 regularization loss (also called Ridge regression [51]) was added to minimize the loss function by summing the squared magnitude of network weight coefficients. Here the LF-DNNs and Correction DNNs were trained in sequence, which is beneficial for the programming and training. The definition of the loss function for the LF-DNNs is shown as follows:

$$L_l = \frac{1}{M}\sum_{i=1}^{M}\left(\widehat{y_{LF}}(\mathrm{X}_{LF};\theta_{NN}) - y_{LF}(\mathrm{X}_{LF})\right)^2 \tag{3}$$

$$\theta_{NN} = \{(w_h, b_h)\}_{h=1}^{H+1} \tag{4}$$

where L_l represents the training loss of LF-DNNs, M means the number of LF training points, θ_{NN} means the set of neural network parameters, w_h is the weights of hth layer, b_h is the biases of hth layer and H means the number of hidden layers. The trained LF-DNNs then operated as an offline surrogate model. This implies that it remained fixed and was not retrained during the training process of the Correction DNNs. Furthermore, the definition of loss function for the Correction DNNs is shown as follows:

$$L_c = L_h + L_r \tag{5}$$

$$L_h = \frac{1}{P}\sum_{i=1}^{P}\left(\widehat{y_{HF}}(\mathrm{X}_{HF};\theta_{NN}) - y_{HF}(\mathrm{X}_{HF})\right)^2 \tag{6}$$

$$L_r = \lambda\sum w_h^2 \tag{7}$$

where L_c represents the overall training loss of Correction DNNs, L_h is the prediction MSE of Correction DNNs, L_r represents the L_2 regularization loss of Correction DNNs, P is the number of HF training points and λ is the control parameter of L_2 regularization loss. During the training process, the LF-DNNs shared the same HF independent variables with the Correction DNNs, i.e., the set $\{\mathrm{X}_{HF}\}$ used in the Correction DNNs were also incorporated during the training phase of the LF-DNNs. The purpose is to enable the Correction DNNs to effectively learn the disparities between the predictions generated by the LF-DNNs $\{\widehat{y_{LF}}\}$ and the corresponding HF dependent variable set $\{y_{HF}\}$, facilitating its ability to correct and refine the output predictions accordingly.

The hyperparameters of the MF-DNNs, encompassing factors like the number of hidden layers, neurons per hidden layer, and learning rate, were fine-tuned using the Bayesian optimization algorithm [52]. Bayesian optimization was chosen for its reputation as a robust global optimization technique suitable for scenarios with noisy black-box problems. It typically demonstrates improved performance in efficiently balancing the trade-off between exploration and exploitation, surpassing conventional grid search [53] and random search methods [54]. To assess the effectiveness of the optimized MF-DNNs, the K-fold

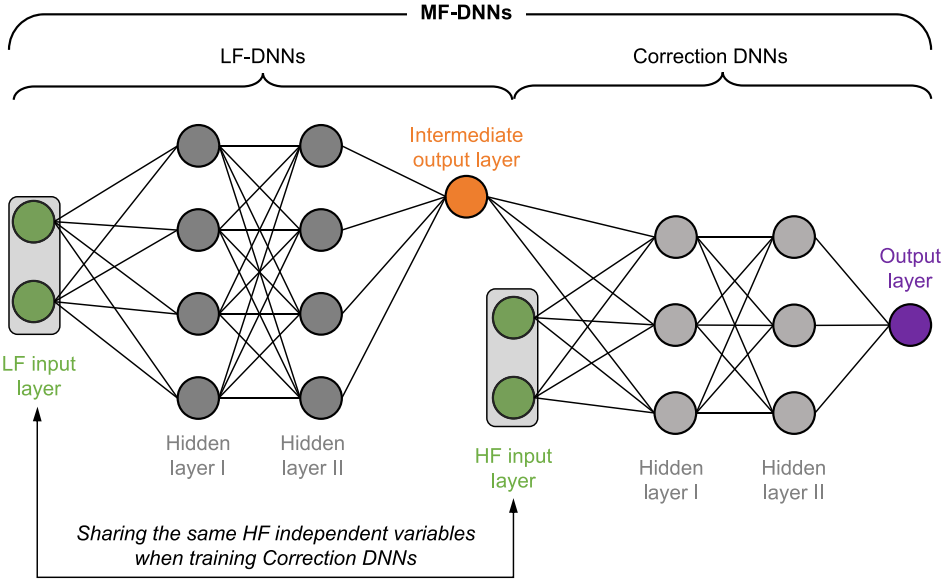


Fig. 1. Architecture of the proposed MF-DNNs (taking two-layer as an example).

cross-validation approach was employed [55]. Diverging from the prevalent usage of the hyperbolic tangent function ("Tanh") in conventional MF-DNNs [16,42], this study adopted the rectified linear unit activation function ("ReLU"). This choice aimed to achieve a harmonious equilibrium between approximating both linear and non-linear correlations within a single sub-network. The mathematical expression of "ReLU" is provided below:

$$f(x) = \max(0, x) \tag{8}$$

The "ReLU" function confers an advantage in expediting and enhancing the training of DNNs compared to traditional logistic sigmoid ("Sigmoid") and hyperbolic tangent ("Tanh") functions, particularly for datasets characterized by high dimensionality [56]. Further explorations of the "ReLU" function's function approximation capabilities are provided in Appendix A at the end of this paper. The construction and training of MF-DNNs were conducted using *Keras* within the *Tensorflow 2* environment [57]. The establishment flowchart and detailed training procedures for the MF-DNNs are as follows:

Algorithm Multi-fidelity Deep Neural Networks

1. Building the initial fully connected LF-DNNs;
2. Optimizing the hyperparameters in LF-DNNs using the Bayesian optimization method;
3. Training the LF-DNNs based on the M realizations of the LF data set $Q_{LF} = \{y_{LF}^{(1)}, \ldots, y_{LF}^{(M)}\}$ using the ADAM and L-BFGS optimizers;
4. Building the initial Correction DNNs and calling the trained LF-DNNs in 3) as a sub-module;
5. Optimizing the hyperparameters in the Correction DNNs using the Bayesian optimization method;
6. Training the Correction DNNs based on the HF data set $Q_{HF} = \{y_{HF}^{(1)}, \ldots, y_{HF}^{(N)}\}$ ($M \gg N$) using the ADAM and L-BFGS optimizers;
7. Validating and testing the accuracy of the trained MF-DNNs based on the K-fold cross-validation method.

3. Results and discussions

Within this section, a series of benchmark tests were executed to thoroughly assess the performance of the constructed MF-DNNs across the realms of surrogate modelling and UQ.

3.1. 1-dimensional function with linear correlation

In this section, the initial tests involved examining a 1-dimensional function characterized by a linear correlation between LF- and HF data. The theoretical expressions of these functions are as follows:

$$y_L(x) = 0.5y_H + 10(x - 0.5) - 5 \tag{9}$$

$$y_H(x) = (6x - 2)^2 \sin(12x - 4) \tag{10}$$

To effectively approximate both the LF- and HF functions, a dataset comprising 21 LF points (uniformly distributed within the interval $X_{LF} \in [0, 1]$) and 4 HF points ($X_{HF} \in [0,\ 0.35,\ 0.75,\ 1]$) was generated and subsequently employed as training data for MF-DNNs. Prior to the official training process of MF-DNNs, key hyperparameters such as the learning rate, the number of hidden layers, and the number of neurons within each hidden layer were meticulously tuned via Bayesian optimization. The ranges of discrete parameters considered were as follows: hidden layer count [1–4], neuron count per hidden layer ([8, 16, 24, 32, 40, 48, 56, 64]) and learning rate ([0.01, 0.001, 0.0001]). The objective of the optimization was to minimize the prediction loss of MF-DNNs. Each training was executed for a total of 2000 epochs. Following 20 iterations of Bayesian optimization, the optimal architecture for the LF-DNNs converged to feature 3 hidden layers with 64, 64, and 40 neurons within each respective hidden layer. The optimal architecture for the Correction DNNs comprised a single hidden layer housing 8 neurons. Furthermore, the optimal learning rate was determined to be 0.001 (Fig. 2).

The weights and biases of MF-DNNs underwent an initial update using the ADAM optimizer for the first 1000 steps. Subsequently, the optimization process continued by incorporating the L-BFGS optimization algorithm for the subsequent 2000 steps. The performance of the well-trained MF-DNNs is visually depicted in Fig. 3. In detail, Fig. 3(a) illustrates that the MF-DNNs precisely approximate the HF function, with the utilization of 4 HF data points. This outcome effectively affirms the competence of the crafted Correction DNNs equipped with the "ReLU" activation function, which adeptly captures the linear correlation between LF- and HF data. Moreover, a comprehensive performance

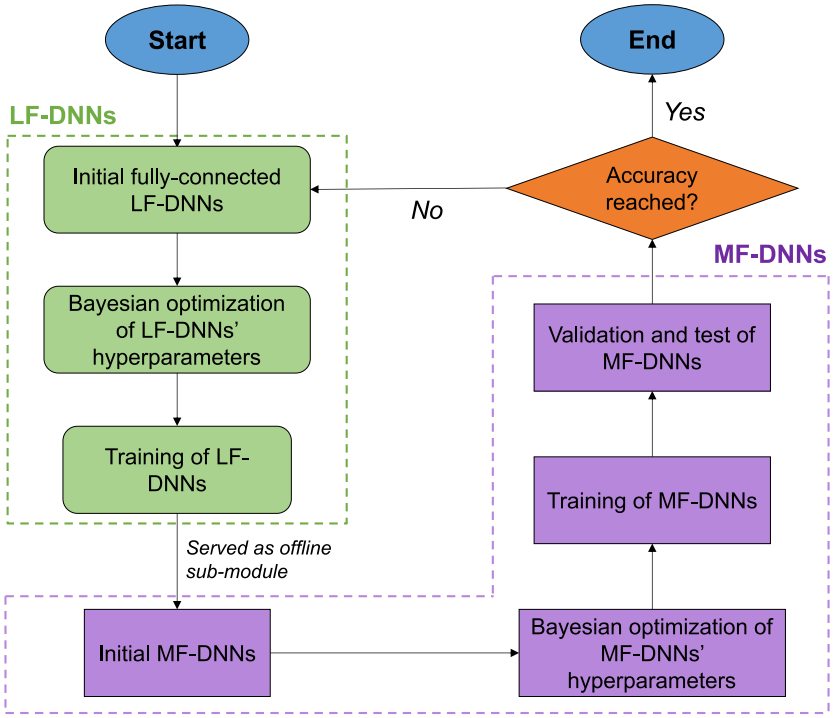


Fig. 2. Establishment flowchart of MF-DNNs.

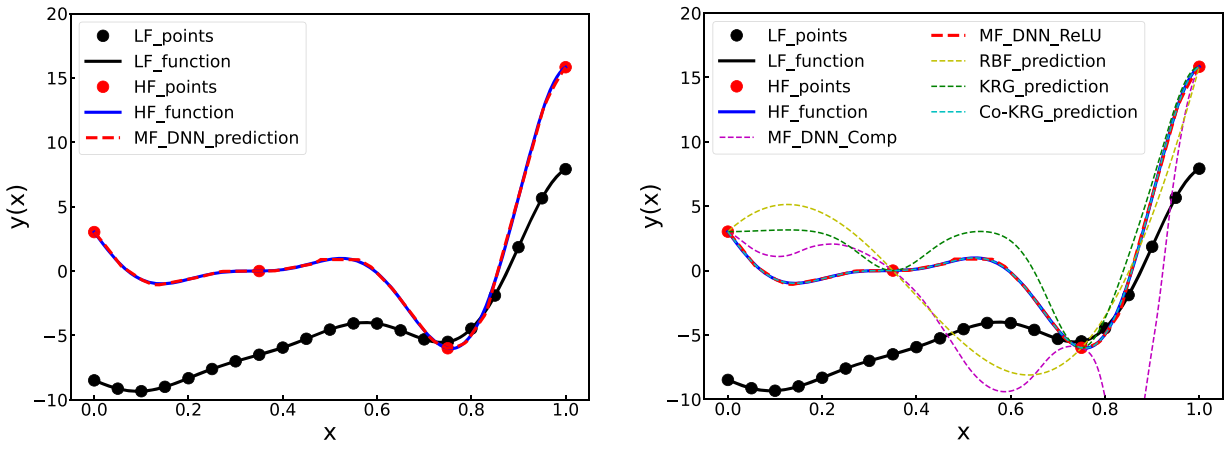


Fig. 3. Performance of MF-DNNs in approximating 1-dimension function with linear correlation (a: Comparison between the MF-DNN prediction results and analytical values; b: Comparison of MF-DNNs with the other surrogate models).

evaluation of the proposed MF-DNNs (denoted as MF_DNN_ReLU) is conducted by comparing it against alternative surrogate models, namely the Radial Basis Function (RBF), Kriging (KRG), Composite Multi-Fidelity Deep Neural Networks (MF_DNN_Comp) as proposed in Ref. [42], and Co-KRG as shown in Fig. 3(b). For a fair comparison, unless specified otherwise, MF_DNN_ReLU and MF_DNN_Comp share identical hyperparameter settings; the sole distinction lies in MF_DNN_Comp incorporating an additional parallel branch for the Correction DNNs.

Both RBF and KRG models were trained based on the HF data points only, and a further insight into the MSE of these models on 1000 validation data points is compiled in Table 1. Among these considered models, the proposed MF-DNNs demonstrate superior performance across the board, with an exception for the Co-KRG model. This result is in line with expectations, considering that Co-KRG was originally formulated based on linear assumptions and should be efficient in handling low-dimensional problems. This also underscores the effectiveness of proposed MF-DNNs in accurately approximating the linear correlation between LF- and HF data.

3.2. 1-dimensional function with non-linear correlation

The one-dimensional function featured by the non-linear correlation between the LF- and HF data was then tested here. The theoretical expressions of this function are as follows:

$$y_L(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5 \tag{11}$$

$$y_H(x) = 0.1y_L(x)^2 + 10 \tag{12}$$

A training dataset was prepared comprising 21 LF points drawn from a uniform distribution within the interval $X_{LF} \in [0,\ 1]$, as well as 6 HF points selected at $X_{HF} \in [0,\ 0.1,\ 0.3,\ 0.7,\ 0.9,\ 1]$. The "ReLU" activation function was implemented to assess its capacity in effectively approximating the non-linear correlation present between the LF- and HF functions. Following a training approach akin to that detailed in Section 3.1, the optimization of NN hyperparameters was performed using the Bayesian optimization method. Each training process spanned 2000 epochs. After 20 iterations of optimization, the LF-DNNs' optimal architecture emerged, consisting of 3 hidden layers with 64, 64, and 40 neurons within each respective hidden layer. Similarly, the Correction DNNs' optimal architecture comprised 2 hidden layers housing 64 and 56 neurons in each hidden layer.

Table 1
MSE for prediction of 1-dimensional function with linear correlation.

Models	MSE of test dataset
MF-DNN_ReLU	9.65e-03
MF-DNN_Comp	4.10e+00
RBF	18.39e+00
KRG	4.21e+00
Co-KRG	4.32e-09

Initially, the weights and biases of the MF-DNNs underwent an update using the ADAM optimizer for the initial 1000 steps. Subsequently, the optimization process continued through the application of the L-BFGS optimization algorithm for the ensuing 2000 steps. The performance of this approach is visually represented in Fig. 4(a), where the MF-DNNs effectively capture and approximate the HF function with accuracy, leveraging insights from 6 HF data points. A comprehensive comparison was then conducted against alternative models, including RBF, KRG, MF_DNN_Comp, and Co-KRG, as depicted in Fig. 4(b). Among these models, the proposed MF-DNNs stand out by exhibiting distributions that closely align with the HF function. The MSE of these models in predicting 1,000 validation data points was summarized in Table 2. The outcomes underscore that the proposed MF-DNNs attain a level of predictive accuracy comparable to the composite structure introduced in Ref. [42]. This shows the ability of Correction NNs equipped with the "ReLU" activation function to effectively approximate the non-linear correlation existing between the LF- and HF data.

3.3. 32-dimensional function

In various industrial design scenarios, dealing with 32 parameters constitutes a moderately high-dimensional space. Consequently, the effectiveness of the proposed MF-DNNs in approximating a 32-dimensional function was evaluated. The underlying mathematical expressions for this function are delineated as follows:

$$y_L(x_0, \ldots, x_{31}) = 0.8 * y_H - 0.4\sum_{i=0}^{30}(x_i x_{i+1}) - 50,\ x_i \in [-3,\ 3] \tag{13}$$

$$y_H(x_0, \ldots, x_{31}) = (x_0 - 1)^2 + \sum_{i=1}^{31}\left(2x_i^2 - x_{i-1}\right)^2,\ x_i \in [-3,\ 3] \tag{14}$$

A training dataset was generated, comprising 200,000 LF- and 2000 HF data points, employing the Latin Hypercube Sampling (LHS) technique [58]. The training process extended across 5000 epochs. Following 20 iterations, the results from the Bayesian optimization

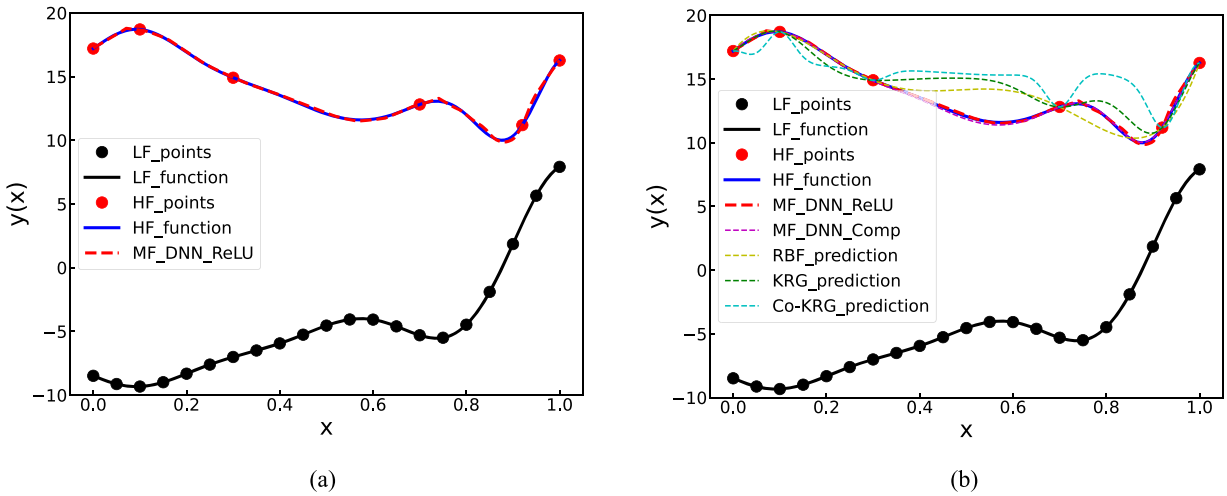


Fig. 4. Performance of MF-DNNs in approximating 1-dimension function with nonlinear correlation (a: Comparison between the MF-DNN prediction results and analytical values; b: Comparison of MF-DNNs with the other surrogate models).

Table 2
MSE for prediction of 1-dimensional function with non-linear correlation.

Models	MSE of test dataset
MF-DNN_ReLU	1.15e-02
MF-DNN_Comp	1.18e-02
RBF	1.22e+00
KRG	2.05e+00
Co-KRG	4.97e+00

method indicated that the LF-DNNs' optimal architecture consisted of 2 hidden layers with 512 neurons and 256 neurons within each respective layer. The Correction DNNs exhibited an optimal architecture featuring a single hidden layer hosting 32 neurons. The "ReLU" activation function was deployed within the Correction-DNNs, while the learning rate was fine-tuned to 0.001 to facilitate an optimal training process. The computational power of the *NVIDIA TESLA K80* Graphics Processing Unit (GPU) available in *Google Colaboratory* was harnessed, thereby enhancing the efficiency of the MF-DNN training process.

The initial phase of training involved updating the weights and biases of MF-DNNs using the ADAM optimizer for the initial 1000 steps. Following this, the training process proceeded with the application of the L-BFGS optimization algorithm, aimed at further diminishing the training loss over the subsequent 2000 steps. Illustrated in Fig. 5 is a comparison between the prediction results of MF-DNNs and the corresponding analytical solutions. The x and y coordinates of the red scatter points represent the MF-DNN predictions and the corresponding analytical solutions, respectively. Ideally, a zero prediction error would result in the red scatter points aligning precisely with the black line (the line slope is 1). In addition, the blue points represent predictions derived from the KRG model, which solely utilizes HF data. The alignment of the red scatter points with the analytical solutions for MF-DNNs underscores the effectiveness in capturing the underlying patterns, particularly in addressing high-dimensional challenges.

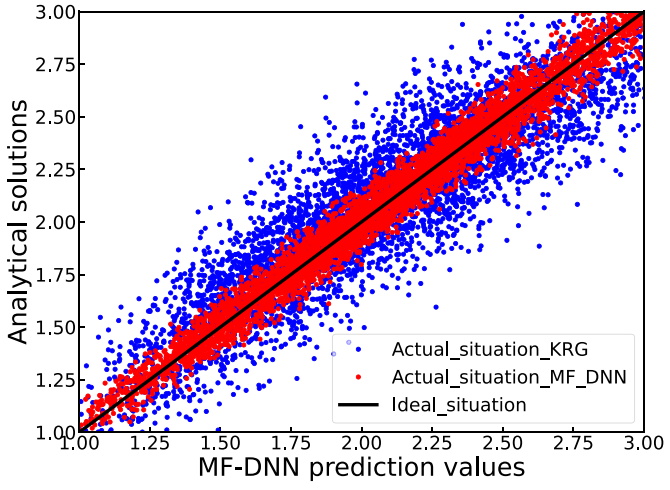


Fig. 5. Performance of MF-DNNs in approximating 32-dimensional function.

Table 3 provides an overview of the MSE exhibited by various models on a validation dataset encompassing 50,000 data points. The outcomes validate that the proposed MF-DNN model achieves a level of accuracy comparable to, even slightly better than, the composite architecture, surpassing the performance of the other assessed models. It's noteworthy that the Co-KRG model faces limitations in addressing this specific problem due to the substantial size of the covariance matrix ([200,000, 200,000]), necessitating a memory capacity of at least 320 GB to yield results.

3.4. 100-dimensional function

The performance of the MF-DNNs, introduced in this study, is further assessed by evaluating its predictive capabilities on a 100-dimensional benchmark function: a significant high-dimensional function in the engineering field. The mathematical expression for the benchmark function under examination can be formulated as follows:

$$y_L(x_0, \ldots, x_{99}) = 0.8 * y_H - 0.4 \sum_{i=0}^{98} (x_i x_{i+1}) - 50, \; x_i \in [-3, \; 3] \tag{15}$$

$$y_H(x_0, \ldots, x_{99}) = (x_0 - 1)^2 + \sum_{i=1}^{99} \left(2x_i^2 - x_{i-1}\right)^2, \; x_i \in [-3, \; 3] \tag{16}$$

Table 3
MSE for prediction of 32-dimensional function with non-linear correlation.

Models	MSE of test dataset
MF-DNN_ReLU	9.83e-04
MF-DNN_Comp	1.01e-03
RBF	1.08e+00
KRG	4.38e-02
Co-KRG	NON

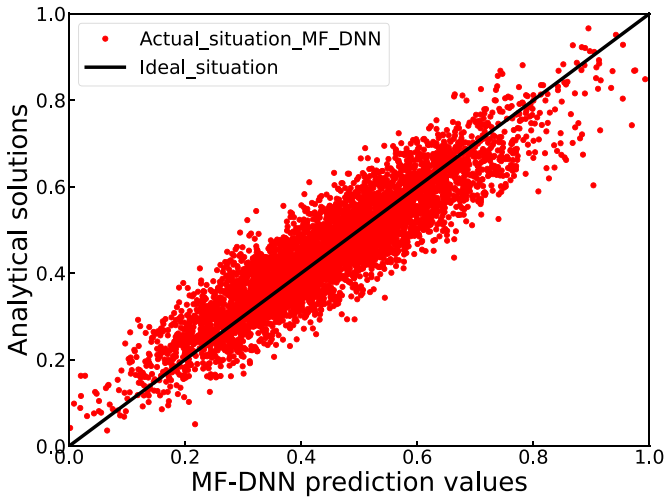


Fig. 6. Performance of MF-DNNs in approximating 100-dimensional function.

The training of the MF-DNN model was conducted utilizing an extensive dataset comprising 10,000,000 LF data points and 100,000 HF data points. Based on the Bayesian optimization results of hyper-parameters, the architectural design of the LF-DNNs encompassed 4 layers, with respective neuron counts of 512, 512, 256, and 128. The Correction DNNs were fashioned as a single-layer network comprising 64 neurons. In both the LF-DNNs and Correction DNNs, the "ReLU" activation function was employed. Additionally, the learning rate was set to 0.001 to facilitate effective learning during training. The comprehensive training procedure was also executed within the *TensorFlow 2* framework, hosted in the *Google Colaboratory* environment.

As illustrated in Fig. 6, MF-DNNs effectively demonstrate its capability to predict the analytical solutions for the 100-dimensional function. The evaluation of multiple models on a validation dataset, comprising 1,000,000 scaled data points, is summarized in Table 4, which provides insight into the MSE values. Notably, the results show that the proposed MF-DNNs outperform the other models in terms of accuracy, thereby affirming its efficacy in addressing high-dimensional surrogate modelling challenges. It's worth noting that the RBF, KRG, and Co-KRG models face limitations during the training process due to their substantial memory requirements, which hinder their successful completion of training.

Table 4
MSE for prediction of 100-dimensional function with non-linear correlation.

Models	MSE of test dataset
MF-DNN_ReLU	4.09e-03
MF-DNN_Comp	4.51e-03
RBF	NON
KRG	NON
Co-KRG	NON

3.5. UQ for 1-dimensional function

The well-trained MF-DNNs were employed here to assess the impact of input aleatory uncertainties on the QoI. Initially, Eqs. (9)–(10) were selected as the governing equations for uncertainty propagation. Subsequently, two representative types of input uncertainty distributions were introduced into the UQ process: uniform distributions and Gaussian distributions. More specifically, the variable $x \in \Gamma$ was treated as either a uniformly distributed random variable over the interval $\Gamma = \mathrm{U}\ [0.6,\ 0.8]$ or a Gaussian distributed random variable with the interval $\Gamma = \mathrm{N}\ [0.7,\ 0.03^2]$. For the generation of Gaussian distributions within a specific bounded interval, the multidimensional truncated Gaussian method [59] was employed. In the UQ process, the MC method was chosen due to its straightforward implementation and robustness. In detail, 10 million sample points generated by the MC method were utilized here to compute the statistical moments. Comparisons were made between the MF-DNN-derived statistical moments of the QoI under the two distinct input uncertainties and the analytical solutions. The outcomes are summarized in Table 5. Notably, the MF-DNNs demonstrated accurate predictions of the QoI's mean and variance. The greatest prediction error was observed in the kurtosis, a measure associated with the tail distributions of the QoI.

Fig. 7 presents a histogram-based comparison of the probability density distributions of the QoI under varying distributions of input uncertainties. The analysis demonstrates that the QoI with the highest probability is concentrated around -6 for both the cases of uniform and Gaussian input uncertainty distributions. However, a notable shift in the probability density occurs when transitioning from uniform to Gaussian input uncertainty distributions, resulting in a concentration of distributions within the interval $[-6, -4]$. Overall, this comparison reveals that MF-DNNs are capable of effectively simulating the low-dimensional uncertainty propagation process.

3.6. UQ for 32-dimensional function

To further assess the capacity of the MF-DNNs in tackling moderately high-dimensional UQ problems, we selected the 32-dimensional benchmark test function (expressed in Eqs. (13)–(14)) as the governing equation for uncertainty propagation. For this test, the input uncertainty variable $x \in \Gamma$ was considered as either a 32-dimensional uniformly distributed random variable within $\Gamma = \mathrm{U}\ [-3,\ 3]$ or a 32-dimensional Gaussian distributed random variable within $\Gamma = N\ (0,\ 1^2)$. To generate the Gaussian distributions across all 32 dimensions, we employed the multidimensional truncated Gaussian method.

Table 5
Comparison of statistical moments of QoI for 1-dimensional function.

	Statistic Description- Input uncertainties with uniform distribution			
	mean	**variance**	**skewness**	**kurtosis**
Analytical results	−3.926	3.706	0.549	−1.165
MF-DNNs prediction (1e+07 samples)	−3.929	3.723	0.551	−1.165
Prediction error (%)	0.076	0.462	0.542	0.012
	Statistic Description- Input uncertainties with Gaussian distribution			
	mean	**variance**	**skewness**	**kurtosis**
Analytical results	−4.426	1.344	0.691	−0.087
MF-DNNs prediction (1e+07 samples)	−4.423	1.356	0.680	−0.090
Prediction error (%)	0.068	0.884	1.592	3.448

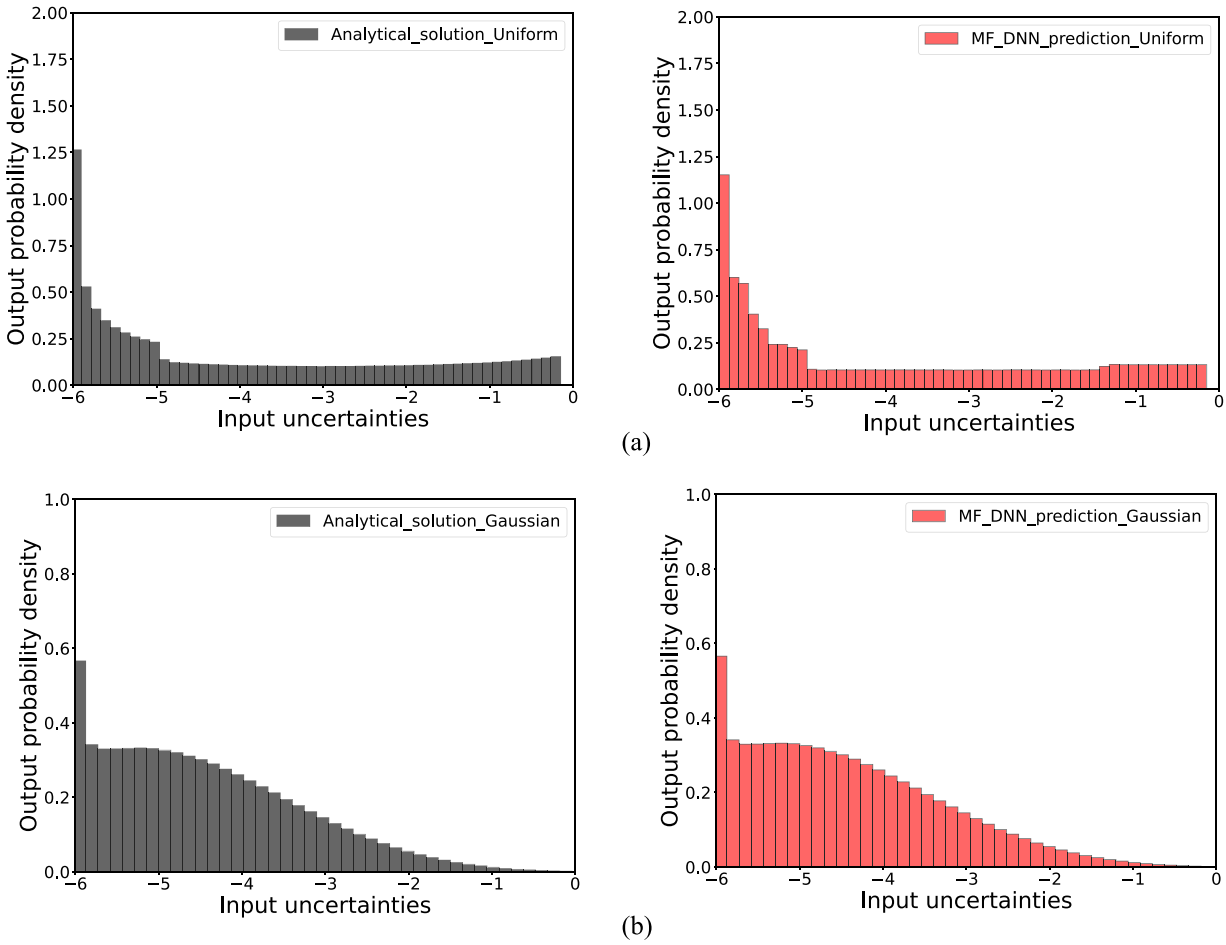


Fig. 7. Histogram comparison of QoI probability density distributions for 1-dimensional function (a: Probability density distributions with uniform uncertainty; b: Probability density distributions with Gaussian uncertainty).

Table 6
Comparison of statistical moments of QoI for 32-dimensional function.

Statistic description- Uniform distribution				
	mean	**variance**	**skewness**	**Kurtosis**
Analytical results	2.106	0.269	0.313	0.103
MF-DNNs predictions (1e+7 samples)	2.105	0.266	0.300	0.080
Prediction error (%)	0.022	0.255	1.344	14.483
Statistic description- Gaussian distribution				
	mean	**variance**	**skewness**	**Kurtosis**
Analytical results	0.364	0.029	0.931	1.144
MF-DNNs predictions (1e+7 samples)	0.364	0.029	0.908	0.939
Prediction error (%)	0.200	0.690	2.469	17.834

Subsequently, the MC method (10 million sample points) was utilized to simulate the uncertainty propagation process. By leveraging the MF-DNNs, the statistical moments of the QoI were computed for both input uncertainty scenarios. These results were then compared against the analytical outcomes, as outlined in Table 6.

While there might be some discrepancies in the kurtosis predictions, the MF-DNNs demonstrate remarkable precision in estimating the mean and variance of the QoI responses. Fig. 8 depicts the histogram comparison of the probability density distributions of the QoI for the 32-dimensional function. The probability density distributions of the QoI exhibit a leftward "bias" as the input uncertainties transit from the uniform to the Gaussian distributions. MF-DNNs can effectively capture and represent both the qualitative and quantitative aspects of the uncertainty propagation process for moderately high-dimensional UQ problems.

3.7. UQ for 100-dimensional function

The 100-dimensional UQ problem was then addressed using the MF-DNN model. The input uncertainty, represented as $x \in \Gamma$, could take two distinct forms: a 100-dimensional uniformly distributed random variable over the interval $\Gamma = \mathrm{U}[-1, 1]$, or a 100-dimensional Gaussian

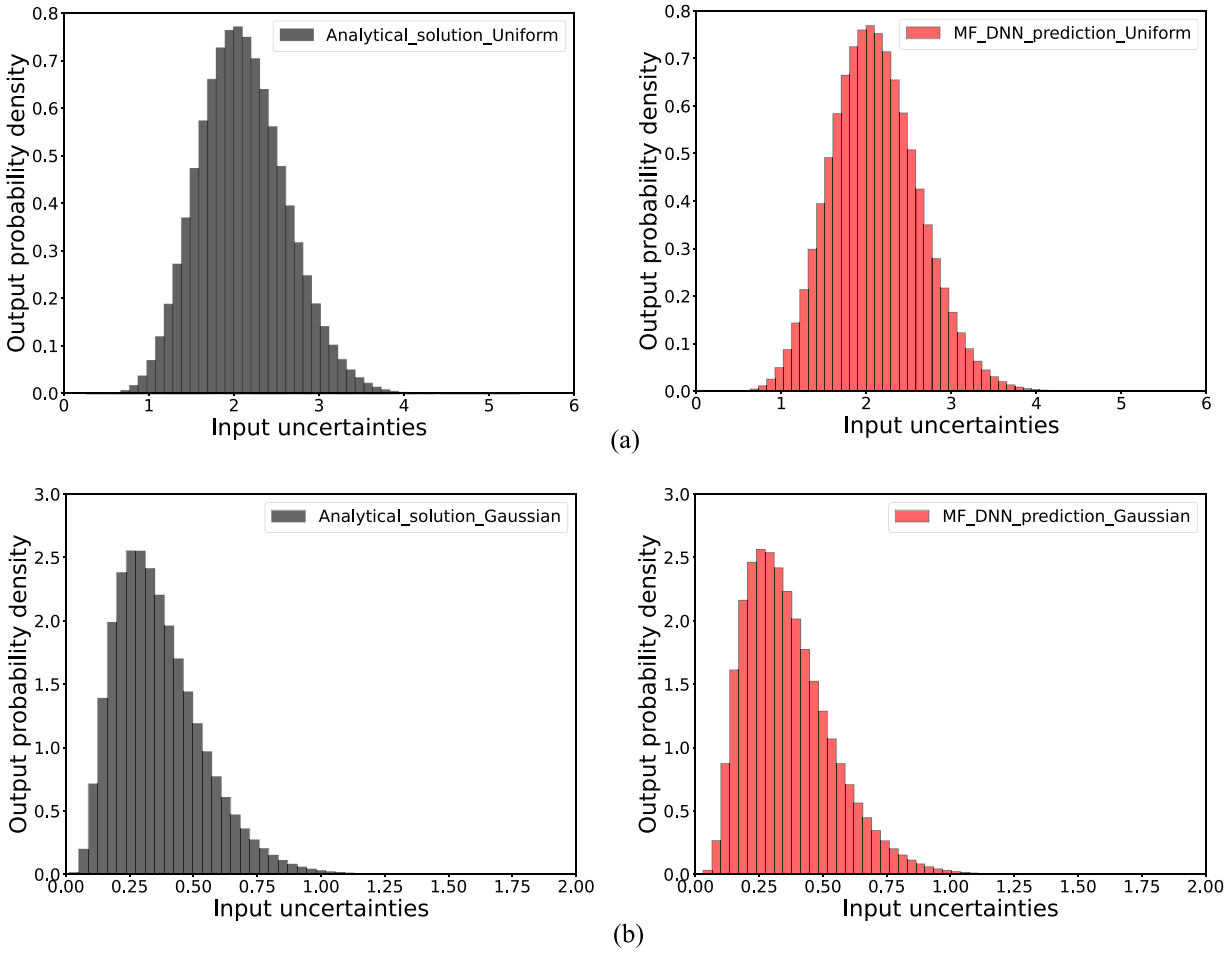


Fig. 8. Histogram comparison of QoI probability density distributions for 32-dimensional function (a: Probability density distributions with uniform uncertainty; b: Probability density distributions with Gaussian uncertainty).

Table 7
Comparison of statistical moments of QoI for 100-dimensional function.

Statistic description- Uniform distribution				
	mean	**variance**	**skewness**	**Kurtosis**
Analytical results	6.716	0.861	0.176	0.029
MF-DNNs predictions (1e+7 samples)	6.723	0.828	0.159	0.033
Prediction error (%)	0.104	3.833	9.659	12.121
Statistic description- Gaussian distribution				
	mean	**variance**	**skewness**	**Kurtosis**
Analytical results	1.159	0.094	0.521	0.358
MF-DNNs predictions (1e+7 samples)	1.150	0.090	0.436	0.279
Prediction error (%)	0.776	4.255	16.315	22.067

distributed random variable with the interval $\Gamma = N\ (0,\ 1^2)$. For both types of input uncertainties, the MC-based (10 million sample points) MF-DNNs were employed to compute the statistical moments of the QoI. These calculated moments were subsequently compared to their corresponding analytical counterparts, and the comparison results are summarized in Table 7. MF-DNNs consistently deliver high accuracy in predicting the mean value, while the most significant deviation is observed in the prediction of kurtosis. Additionally, the probability density distributions of the QoI were determined using MF-DNNs, and the histogram comparison of these distributions is illustrated in Fig. 9. Although some discrepancies are observed in the prediction of variance, skewness, and kurtosis when compared to the analytical results, the probability distributions derived from the MF-DNNs closely align with the analytical solutions.

3.8. Application in predicting turbine nozzle flow

As a crucial component, the aeroengine provides thrust, electric power, and compressed air for the aircraft, and any potential failures could result in catastrophic consequences. Thus, assessing the aeroengine performance and its reliability in the presence of external aleatory uncertainties is critical during both the design and operation phases [60]. This paper utilizes MF-DNNs to predict the flow field within a representative aeroengine component, specifically focusing on the LS89

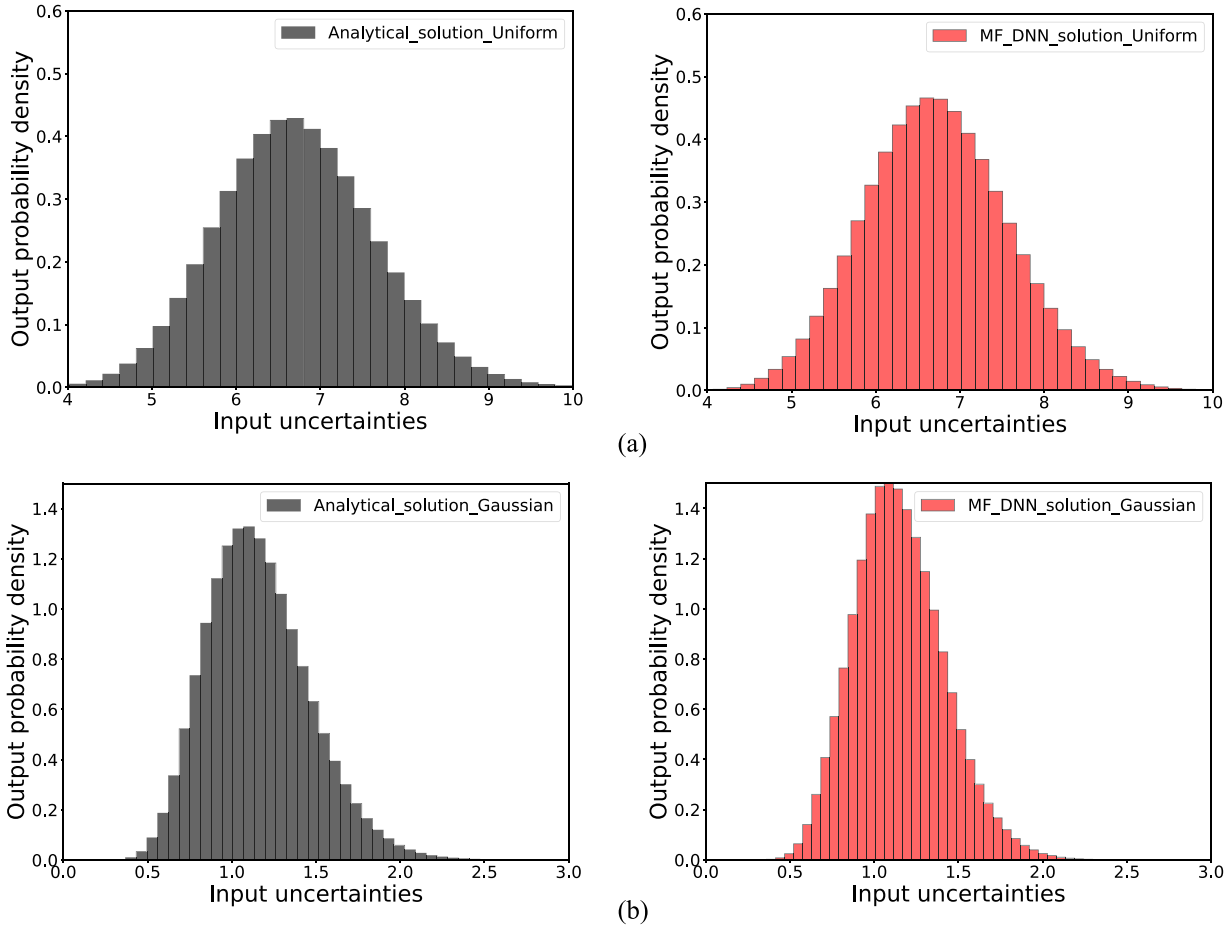


Fig. 9. Histogram comparison of QoI probability density distributions for 100-dimensional function (a: Probability density distributions with uniform uncertainty; b: Probability density distributions with Gaussian uncertainty).

Table 8
Key parameters of LS89 vane.

Parameter	Value
Chord	67.647 mm
Pitch	57.500 mm
Stagger angle	55°
LE radius	4.127 mm
TE radius	0.710 mm
Inlet total pressure	159, 600 Pa
Inlet total temperature	420 K
Inlet flow angle	0°
Inlet turbulence intensity	3 %
Outlet static pressure	82,350 Pa

turbine vane [61] as the reference case. The LS89 nozzle guide vane was originally designed and subjected to experimental testing at the von Karman Institute for Fluid Dynamics, with comprehensive measurements available. The essential geometric and operational parameters of the LS89 blade are outlined in Table 8.

To predict the near-wall isentropic Mach number distributions on the vane surface, which play a pivotal role in determining the flow patterns of turbine nozzle flow, two fidelities of training data were employed. Abundant 2D Euler flow data served as the LF source, while a limited set of experimental measurements acted as the HF source. The governing equations corresponding to these two levels of fidelities for the compressible flow field are presented below:

$$\textbf{\textit{Euler flow}}: \begin{cases} \dfrac{D\rho}{Dt} = -\rho\nabla\cdot\boldsymbol{u} \\ \dfrac{D\boldsymbol{u}}{Dt} = -\dfrac{\nabla p}{\rho} \\ \dfrac{De}{Dt} = -\dfrac{p\nabla\cdot\boldsymbol{u}}{\rho} \end{cases} \tag{17}$$

$$\textbf{\textit{Navier}} - \textbf{\textit{Stokes flow}}: \begin{cases} \dfrac{D\rho}{Dt} = -\rho\nabla\cdot u \\ \dfrac{D\boldsymbol{u}}{Dt} = -\dfrac{\nabla p}{\rho} + \dfrac{\mu}{\rho}\nabla^2\boldsymbol{u} + \dfrac{\mu}{\rho}\nabla\left(\dfrac{\nabla\cdot\boldsymbol{u}}{3}\right) \\ \dfrac{De}{Dt} = -\dfrac{p\nabla\cdot\boldsymbol{u}}{\rho} + \nabla\cdot(\kappa\nabla T) + \dfrac{\mu}{\rho}\nabla^2\boldsymbol{u} + \dfrac{\mu}{\rho}\nabla\left(\dfrac{\nabla\cdot\boldsymbol{u}}{3}\right) \end{cases} \tag{18}$$

where ρ is the air density, t the time, p the pressure, $\boldsymbol{u}$ the flow velocity vector, ∇ the differential operator, μ the dynamic viscosity, e the internal energy, T the temperature, κ the thermal conductivity and $\frac{D}{Dt}$ the material derivative. The 2D Euler flow was numerically solved using the computational code *MISES* (Multiple Blade Interacting Streamtube Euler Solver), which was developed by Drela and Youngren at MIT [57]. The mesh for the simulation was generated using *MISES/ISET* with around 3, 180 cells generated in the LF simulations. The detailed 2D mesh within the LS89 passage along with the contour of the Mach number distribution are illustrated in Fig. 10.

The Euler flow analysis reveals the presence of two shock waves within the vane passage. The front shock wave arises due to the choking

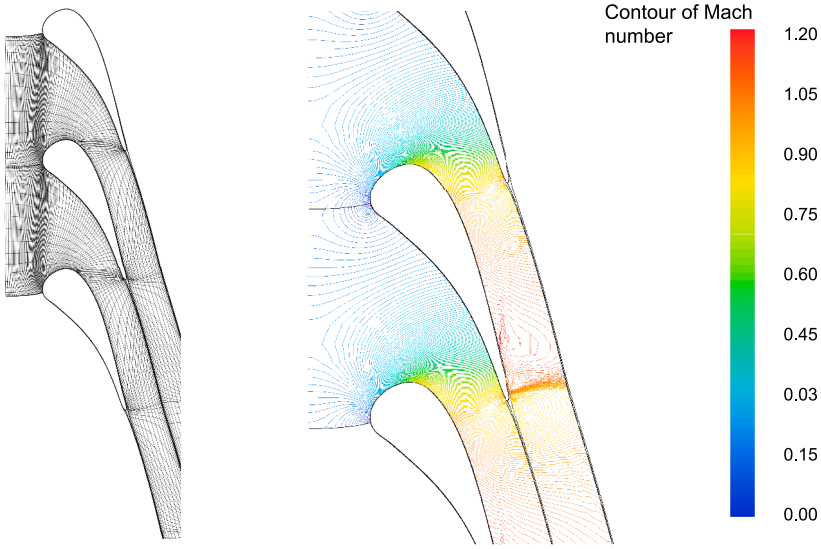


Fig. 10. Generated mesh and contour of Mach number inside turbine nozzle passages.

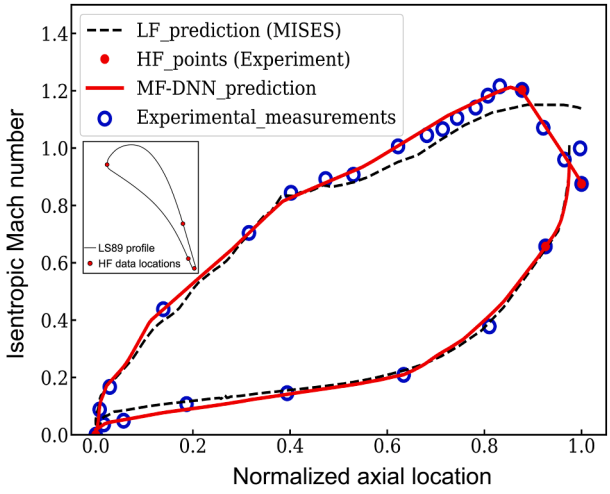


Fig. 11. MF-DNN prediction of near-wall isentropic Mach number on LS89 vane surface.

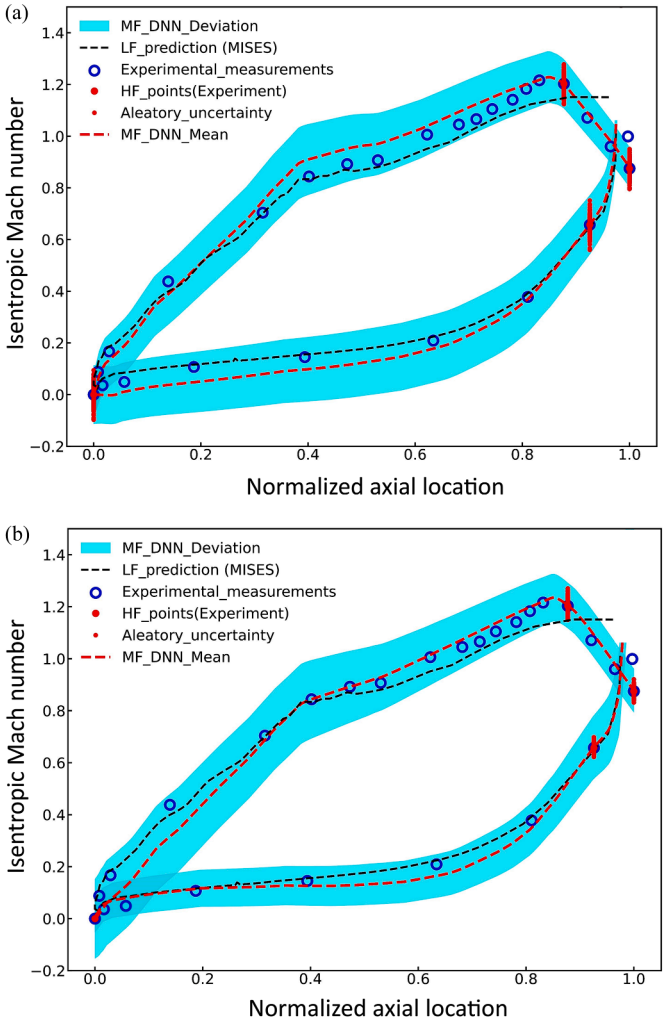


Fig. 12. MF-DNN predictions of near-wall isentropic Mach number on LS89 vane surface considering aleatory uncertainty from experimental measurements (a: Error with a fix variation interval of 0.2; b: Error with a normalized variation interval to 10 %).

effect in the throat area, while the rear shock wave forms near the trailing edge due to the flow acceleration on the suction surface. However, it's important to note that the *MISES* solver typically determines the behaviour of the trailing edge flow based on the Kutta condition, rather than explicitly resolving the underlying flow physics [62]. Consequently, the Euler flow results may not accurately predict the base pressure and losses in the trailing edge region [63–64]. To address this limitation, MF-DNNs are adopted here to capture the fundamental flow patterns using LF Euler results and then refine the trailing edge flow field using HF experimental measurements.

In this context, the primary focus of MF-DNNs is to predict the distributions of isentropic Mach numbers on the vane surface. The reason is that the distributions of Mach numbers directly determine the flow patterns inside the turbine nozzle. More comprehensive reconstruction of the flowfield utilizing MF-DNNs can be achieved after obtaining HF experimental data from within the cascade passage. For the training data input, a combination of 160 LF points and 4 HF points were selected. The modelling process separately addressed the near-wall flow fields on both the pressure and suction sides of the LS89 vane.

The LF-DNN architecture comprised 5 hidden layers with 256, 128, 128, 64, and 32 neurons in each layer, respectively. Meanwhile, the Correction DNNs were constructed with 4 hidden layers containing 128, 64, 32, and 32 neurons on each respective layer. The comparison outcomes of isentropic Mach number distributions are depicted in Fig. 11. Unlike the approach of placing pressure probes across the entire chord of the vane surface, MF-DNNs accurately predicted the distributions of the isentropic Mach number based on the 2D Euler flow field and only 4 experimental measurement data points. The MF-DNNs inherited the flow pattern from the upstream region and intelligently corrected the Mach number distributions near the shock wave occurring close to the trailing edge.

In the following section, the training process of the MF-DNNs includes the incorporation of aleatory uncertainties stemming from HF experimental measurement errors. Specifically, two distinct types of measurement errors were taken into consideration. The first type involved uncertainties with a fixed value-based variation interval, wherein uncertain Mach numbers were distributed within intervals of 0.2. The second type comprised uncertainties normalized to 10 %. In both cases, aleatory uncertainties were modelled using Gaussian distributions. In contrast to the previous MF-DNN configurations with determined outputs, the MF-DNNs were updated to accommodate these aleatory uncertainties. This entailed adjusting the output layer to include 2 neurons, enabling the prediction of the mean and standard deviation, respectively. The prediction outcomes of the updated MF-DNNs are depicted in Fig. 12. Evidently, the MF-DNNs adeptly predicted mean values, utilizing information from 4 HF data points encompassing two different forms of aleatory uncertainties. Furthermore, the MF-DNNs offered reasonable estimations of deviations, corresponding to 95 % confidence intervals. These results serve to confirm the capability of the proposed MF-DNNs in effectively addressing aleatory uncertainty challenges within real-world engineering applications.

4. Conclusions

This paper introduces multi-fidelity deep neural networks (MF-DNNs) as a solution for high-dimensional aleatory uncertainty quantification (UQ) problems. The key conclusions drawn from this research are as follows:

1) The effectiveness of MF-DNNs is demonstrated through the accurate approximation of various benchmark functions with increasing dimensions. While the Co-Kriging model can efficiently address low-dimensional problems characterized by linear correlation, the superior performance of MF-DNNs in managing high-dimensional challenges is highlighted when compared to other models.
2) MF-DNNs demonstrate their effectiveness in predicting probability density distributions and statistical moments of quantities of interest (QoI) across various UQ problems. The method has been proved with different PDF inputs (gaussian and uniform) and the method was consistent with Monte Carlo output.
3) The model has been applied to a real aero engine problem. MF-DNNs precisely capture the near-wall flow characteristics of LS89 turbine vane, inheriting the upstream low-fidelity (LF) Euler flow pattern and intelligently correcting Mach number distributions near shock waves using high-fidelity (HF) experimental data.

CRediT authorship contribution statement

Zhihui Li: Writing – original draft, Methodology, Investigation, Data curation, Conceptualization. **Francesco Montomoli:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The authors would like to sincerely thank the support of the grant from the European Union's Marie Skłodowska-Curie Actions Individual Fellowship (MSCA-IF-MENTOR-101029472).

Appendix A. Comparison of activation functions in neural networks

A comparison of the capabilities of different activation functions, namely "ReLU", "Sigmoid" and "Tanh" in regression was carried out in this section. Theoretical analysis reveals that all three of these activation functions are nonlinear, according to the definition of nonlinearity (where the derivative of the dependent variable changes with the independent variable). They have been extensively used for approximating nonlinear functions. This section investigates their effectiveness in approximating both linear and nonlinear functions. To study this, four types of simple linear and nonlinear unary functions are chosen as the governing equations, as follows:

$$y = 2x,\ x \in [-1, 1] \tag{A.1}$$

$$y = 2x^2,\ x \in [-1, 1] \tag{A.2}$$

$$y = 2x^3,\ x \in [-1, 1] \tag{A.3}$$

$$y = 2\sin(4x),\ x \in [-1, 1] \tag{A.4}$$

To evaluate the influence of different activation functions ("ReLU", "Sigmoid" and "Tanh") on neural network performance, three separate neural networks (NNs) were established. Each NN employed a distinct activation function while keeping other factors, such as hyperparameters and initialization, consistent. A total of 21 training points were generated uniformly within the interval $x \in [-1,\ 1]$. The NN architecture featured 2 hidden layers, each containing 20 neurons. Training was carried out for 1000 epochs, with a batch size of 7.

Fig. A.1 illustrates the comparison of predictions obtained using different activation functions. Notably, both "ReLU" and "Tanh" activations exhibit closer agreement with theoretical values compared to the "Sigmoid" activation. To provide a quantitative evaluation, Table A.1 presents the MSE values on the validation dataset. Across various linear and nonlinear functions, the "ReLU" activation consistently demonstrates higher prediction accuracy compared to the other activation functions. These results affirm that the proposed Correction DNNs, employing the "ReLU" activation function, can adeptly approximate both linear and nonlinear correlations across a diverse array of scenarios.

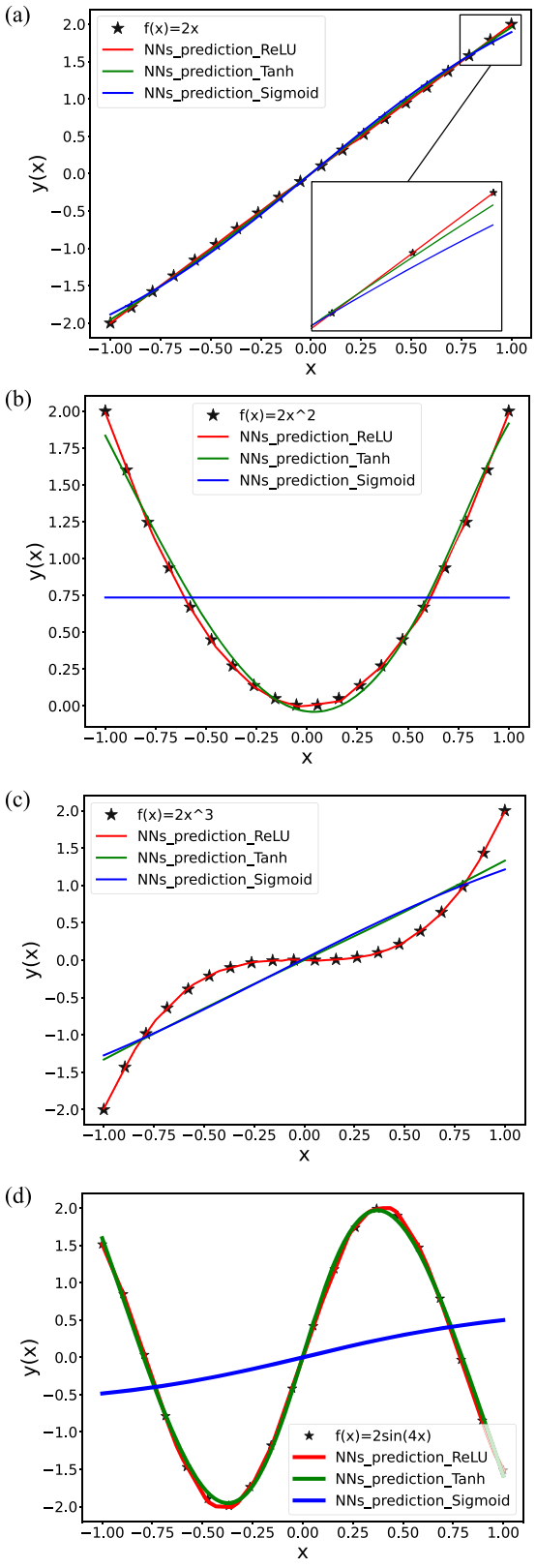


Fig. A.1. Comparison of NNs with various activation functions in approximating linear/nonlinear functions

Table A.1
MSE comparison of NNs with various activation functions.

	ReLU	Tanh	Sigmoid
Test MSE for A.1	9.48e-06	4.50e-04	2.35e-03
Test MSE for A.2	5.13e-05	1.25e-03	3.61e-01
Test MSE for A.3	9.77e-05	1.38e-03	1.00e-02
Test MSE for A.4	2.62e-04	1.12e-03	5.62e-04

Reference

[1] Wu SJ, Hsu CT. Modeling of uncertainty for flood wave propagation induced by variations in initial and boundary conditions using expectation operator on explicit numerical solutions. Int J Numer Methods Eng 2018;113(9):1447–65.
[2] Li Z, Liu Y, Agarwal RK. Uncertainty quantification of geometric and flow variables affecting the performance of a transonic axial compressor. In: 2018 AIAA aerospace sciences meeting (0068); 2018.
[3] Montomoli F, Carnevale M, D'Ammaro A, Massini M, Salvadori S. Uncertainty quantification in computational fluid dynamics and aircraft engines. New York: Springer International Publishing; 2015.
[4] Sciacchitano A. Uncertainty quantification in particle image velocimetry. Meas Sci Technol 2019;30(9):092001.
[5] Jung Y, Jo H, Choo J, Lee I. Statistical model calibration and design optimization under aleatory and epistemic uncertainty. Reliab Eng Syst Saf 2022;222:108428.
[6] Hu Z, Mahadevan S. Uncertainty quantification in prediction of material properties during additive manufacturing. Scr Mater 2017;135:135–40.
[7] Yao W, Zheng X, Zhang J, Wang N, Tang G. Deep adaptive arbitrary polynomial chaos expansion: a mini-data-driven semi-supervised method for uncertainty quantification. Reliab Eng Syst Saf 2023;229:108813.
[8] Kuo FY, Sloan IH. Lifting the curse of dimensionality. Notices AMS 2005;52(11):1320–8.
[9] Forrester AI, Sóbester A, Keane AJ. Multi-fidelity optimization via surrogate modelling. Proc R Soc A Math Phys Eng Sci 2007;463(2088):3251–69.
[10] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N. and González, J. (2019). Deep Gaussian processes for multi-fidelity modelling. *arXiv preprint arXiv:1903.07320*.
[11] Jofre, L., Geraci, G., Fairbanks, H., Doostan, A. and Iaccarino, G. (2018). Multi-fidelity uncertainty quantification of irradiated particle-laden turbulence. *arXiv preprint arXiv:1801.06062*.
[12] Reeve ST, Strachan A. Error correction in multi-fidelity molecular dynamics simulations using functional uncertainty quantification. J Comput Phys 2017;334:207–20.
[13] Fernández-Godino, M. G., Park, C., Kim, N. H. and Haftka, R. T. (2016). Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*.
[14] Vitali R, Haftka RT, Sankar BV. Multi-fidelity design of stiffened composite panel with a crack. Struct Multidiscip Optim 2002;23(5):347–56.
[15] Alexandrov NM, Lewis RM, Gumbert CR, Green LL, Newman PA. Approximation and model management in aerodynamic optimization with variable-fidelity models. J Aircr 2001;38(6):1093–101.
[16] Guo M, Manzoni A, Amendt M, Conti P, Hesthaven JS. Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities. Comput Methods Appl Mech Eng 2022;389:114378.
[17] Basir S, Senocak I. Physics and equality constrained artificial neural networks: application to forward and inverse problems with multi-fidelity data fusion. J Comput Phys 2022:111301.
[18] Brooks CJ, Forrester AIJ, Keane AJ, Shahpar S. Multi-fidelity design optimisation of a transonic compressor rotor. In: 9th European Conf. Turbomachinery fluid dynamics and thermodynamics; 2011. *21-25 Mar 2011*.
[19] Parussini L, Venturi D, Perdikaris P, Karniadakis GE. Multi-fidelity Gaussian process regression for prediction of random fields. J Comput Phys 2017;336:36–50.
[20] Raissi, M. and Karniadakis, G. (2016). Deep multi-fidelity Gaussian processes. *arXiv preprint arXiv:1604.07484*.
[21] Yang X, Zhu X, Li J. When bifidelity meets cokriging: an efficient physics-informed multifidelity method. SIAM J Sci Comput 2020;42(1):A220–49.
[22] Xu Y, Renteria A, Wang P. Adaptive surrogate models with partially observed information. Reliab Eng Syst Saf 2022;225:108566.
[23] Palar PS, Tsuchiya T, Parks GT. Multi-fidelity non-intrusive polynomial chaos based on regression. Comput Methods Appl Mech Eng 2016;305:579–606.
[24] Berchier M. Multi-fidelity surrogate modelling with polynomial chaos expansions. Doctoral dissertation, Msc thesis. ETH Zurich; 2016.
[25] Konrad J, Farcaş IG, Peherstorfer B, Di Siena A, Jenko F, Neckel T, Bungartz HJ. Data-driven low-fidelity models for multi-fidelity Monte Carlo sampling in plasma micro-turbulence analysis. J Comput Phys 2022;451:110898.
[26] Patsialis D, Taflanidis AA. Multi-fidelity Monte Carlo for seismic risk assessment applications. Struct Saf 2021;93:102129.
[27] Lee D, Kramer B. Multifidelity conditional value-at-risk estimation by dimensionally decomposed generalized polynomial chaos-Kriging. Reliab Eng Syst Saf 2023;235:109208.
[28] Manzhos, S. and Ihara, M. (2021). Rectangularization of Gaussian process regression for optimization of hyperparameters. *arXiv preprint arXiv:2112.02467*.
[29] Perdikaris P, Raissi M, Damianou A, Lawrence ND, Karniadakis GE. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proc R Soc A Math Phys Eng Sci 2017;473(2198):20160751.
[30] Motamed M, Nobile F, Tempone R. Analysis and computation of the elastic wave equation with random coefficients. Comput Math Appl 2015;70(10):2454–73.
[31] Peherstorfer B, Willcox K, Gunzburger M. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Rev 2018;60(3):550–91.
[32] Kutz JN. Deep learning in fluid dynamics. J Fluid Mech 2017;814:1–4.
[33] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[34] O'shea T, Hoydis J. An introduction to deep learning for the physical layer. IEEE Trans Cognit Commun Netw 2017;3(4):563–75.
[35] Angermueller C, Pärnamaa T, Parts L, Stegle O. Deep learning for computational biology. Mol Syst Biol 2016;12(7):878.
[36] Lu Y, Lu J. A universal approximation theorem of deep neural networks for expressing probability distributions. Adv Neural Inf Process Syst 2020;33:3094–105.
[37] Lu C, Zhu X. Bifidelity data-assisted neural networks in nonintrusive reduced-order modeling. J Sci Comput 2021;87:1–30.
[38] Yan, L. and Zhou, T. (2019). An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems. *arXiv preprint arXiv:1911.08926*.
[39] Conti P, Guo M, Manzoni A, Hesthaven JS. Multi-fidelity surrogate modeling using long short-term memory networks. Comput Methods Appl Mech Eng 2023;404:115811.
[40] Dhulipala SL, Shields MD, Chakroborty P, Jiang W, Spencer BW, Hales JD, Laboure VM, Prince ZM, Bolisetti C, Che Y. Reliability estimation of an advanced nuclear fuel using coupled active learning, multifidelity modeling, and subset simulation. Reliab Eng Syst Saf 2022;226:108693.
[41] Motamed M. A multi-fidelity neural network surrogate sampling method for uncertainty quantification. Int J Uncertainty Quantif 2020;10(4):315–32.
[42] Meng X, Karniadakis GE. A composite neural network that learns from multi-fidelity data: application to function approximation and inverse PDE problems. J Comput Phys 2020;401:109020.
[43] Zhang X, Xie F, Ji T, Zhu Z, Zheng Y. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Comput Methods Appl Mech Eng 2021;373:113485.
[44] Ahn JG, Yang HI, Kim JG. Multi-fidelity meta modeling using composite neural network with online adaptive basis technique. Comput Methods Appl Mech Eng 2022;388:114258.
[45] Perdikaris P, Venturi D, Royset JO, Karniadakis GE. Multi-fidelity modelling via recursive co-Kriging and Gaussian–Markov random fields. Proc R Soc A Math Phys Eng Sci 2015;471(2179):20150018.
[46] Qian PZ, Wu CJ. Bayesian hierarchical modelling for integrating low-accuracy and high-accuracy experiments. Technometrics 2008;50(2):192–204.
[47] Kingma, D. P. and Ba, J. (2014). ADAM: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
[48] Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 1995;16(5):1190–208.
[49] Amari SI. Backpropagation and stochastic gradient descent method. Neurocomputing 1993;5(4-5):185–96.
[50] Netrapalli P. Stochastic gradient descent and its variants in machine learning. J Indian Inst Sci 2019;99(2):201–13.
[51] Saccoccio M, Wan TH, Chen C, Ciucci F. Optimal regularization in distribution of relaxation times applied to electrochemical impedance spectroscopy: ridge and lasso regression methods-a theoretical and experimental study. Electrochim Acta 2014;147:470–82.
[52] Klein A, Falkner S, Bartels S, Hennig P, Hutter F. Fast Bayesian optimization of machine learning hyperparameters on large datasets. Artificial intelligence and statistics. PMLR; 2017. p. 528–36.
[53] Ndiaye E, Le T, Fercoq O, Salmon J, Takeuchi I. Safe grid search with optimal complexity. In: International conference on machine learning. PMLR; 2019. p. 4771–80.
[54] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res 2012;13(2).

[55] Refaeilzadeh P, Tang L, Liu H. Cross-validation. Encycl. Database Syst. 2009;5:532–8.
[56] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2011. p. 315–23.
[57] Gulli A, Kapoor A, Pal S. Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API. Packt Publishing Ltd; 2019.
[58] Florian A. An efficient sampling scheme: updated latin hypercube sampling. Probab Eng Mech 1992;7(2):123–30.
[59] Botev ZI. The normal law under linear restrictions: simulation and estimation via minimax tilting. J R Stat Soc Ser B (Statistical Methodology) 2017;79(1):125–48.
[60] Pan WH, Feng YW, Lu C, Liu JQ. Analyzing the operation reliability of aeroengine using quick access recorder flight data. Reliab Eng Syst Saf 2023;235:109193.
[61] Arts T, Lambertderouvroit M, Rutherford AW. Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations. NASA STI/Recon; 1990. Technical Report N, 91, 23437.
[62] Drela M, Youngren H. A user's guide to MISES 2.53. Cambridge, MA: Massachusetts Institute of Technology; 1998.
[63] Andrew PL, Kahveci HS. Validation of MISES 2-D boundary layer code for high pressure turbine aerodynamic design. In: Turbo expo: power for land, sea, and air. 47950; 2007. p. 879–93.
[64] Poli R, Pini M, Rao AG. A simplified method for wall temperature prediction in externally cooled turbines. 1st Global power and propulsion forum (GPPF-2017). 2017.