JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 20, No. 5, May 2023

Multifidelity Prediction Framework with Convolutional Neural Networks Using High-Dimensional Data

Huseyin Emre Tekaslan* and Melike Nikbay†
Istanbul Technical University, 34469 Istanbul, Turkiye

https://doi.org/10.2514/1.I011159

This paper proposes two novel multifidelity neural network architectures developed for high-dimensional inputs such as computational flowfields. We employed a two-dimensional flow-varying transonic supercritical airfoil problem while exploring and comparing our methods with the former "multifidelity deep neural networks" from the literature. We call these novel methods "modified multifidelity deep neural networks" and "multifidelity convolutional neural networks." The main objective of this study is to establish an advanced multifidelity prediction framework that can be applied to any computational data; however, here, we applied our methods to the prediction of aerodynamic coefficients using pressure coefficient fields around the airfoil. To generate the dataset, first, a coarse grid is employed using the SU2 Euler solver for low-fidelity data; then, a relatively finer grid is used to obtain the viscous solutions by using the Spalart–Allmaras turbulence model for the high-fidelity data. The performance metrics to compare the methods are determined as the test accuracy, the physical training time, and the size of the high-fidelity samples. The results demonstrate that the proposed multifidelity neural network architectures outperform the multifidelity deep neural networks in predictive modeling using high-dimensional inputs by improving the multifidelity prediction accuracy significantly.

Nomenclature

$\boldsymbol{b}$	=	layer bias
C_D	=	drag coefficient
C_L	=	lift coefficient
C_N	=	normal force coefficient
C_p	=	pressure coefficient
D_i	=	an arbitrary array dimension
$\mathcal{F}_l$	=	linear correlation network
$\mathcal{F}_{nl}$	=	nonlinear correlation network
$\mathcal{L}$	=	loss function
M	=	Mach number
m_p	=	dimension of low- and high-fidelity predictions
NS_H	=	number of high-fidelity samples
NS_L	=	number of low-fidelity samples
n_H	=	high-fidelity input dimension
$n_{H,E}$	=	encoded input dimension
n_θ	=	number of learnable parameters
Re	=	Reynolds number
T_{hf}	=	total computation time of high-fidelity predictions, s
T_{mf}	=	total computation time of multifidelity predictions, s
t_h	=	computation time of a single high-fidelity simulation, s
t_l	=	computation time of a single low-fidelity simulation, s
$\boldsymbol{W}$	=	layer weight
W_i	=	dimension of a layer weight
x	=	scalar input or multidimensional input array
$\boldsymbol{x}$	=	vectorized multidimensional input array
$\boldsymbol{x}_C$	=	input of the correlation networks
$\boldsymbol{x}_H$	=	high-fidelity input array
$\boldsymbol{x}_H$	=	vectorized high-fidelity input array
$\boldsymbol{x}_L$	=	low-fidelity input array
$\boldsymbol{x}_L$	=	vectorized low-fidelity input array
y	=	scalar target or multidimensional target array
y_H	=	high-fidelity data
y_L	=	low-fidelity data
$\hat{\boldsymbol{y}}_H$	=	high-fidelity prediction
$\hat{\boldsymbol{y}}_L$	=	low-fidelity prediction
α	=	angle of attack, deg
δ	=	additive surrogate model
θ	=	learnable parameter set of a neural network
λ	=	regularization strength
ρ	=	multiplicative surrogate model
Ω	=	regularization function
ω	=	correlation weight

I. Introduction

ARTIFICIAL neural networks (ANNs) started a new era in computer and computational sciences. Practical conundrums a few decades ago (such as image, video, and audio recognition, as well as machine translation) can now be resolved with an adequate size of data and neural networks in the tasks of classification, clustering, and estimation. However, the idea of the neural network is not newly fashionable, and the concept dates back to the 1960s. Even though this method has a history nowadays, it has recently become renowned because central processing units (CPUs) and graphics processing units (GPUs) offer adequate computing power. Besides, data accumulation is getting easier and cheaper. Thus, this outstanding data-driven metamodeling technique has been disseminated over time and has become ubiquitous across different disciplines such as bioinformatics [1–3] and geosciences [4–6].

In fact, data-driven metamodeling has been used in computational sciences for a few decades. In this field, it is mostly known as surrogate or reduced-order modeling. In the literature, numerous data-driven surrogate modeling methods exploiting different mathematical approaches such as mapping and regression are available. Principal component analysis [7,8], proper orthogonal decomposition [9–11], dynamic mode decomposition [12,13], polynomial chaos expansion [14–16], and kriging [17,18] can be given as examples of surrogate modeling techniques that are studied in computational sciences for complex and expensive-to-solve problems. Recently, ANNs gained a considerable amount of content in aerospace. Guidance [19,20], air traffic management [21,22], optimization [23–26], uncertainty quantification [27,28], and inverse design [29–32] are some of the fields that incorporate deep learning methods in research. For instance, in computational aerodynamics and multidisciplinary studies involving aerodynamics, data are scarce

Presented as Paper 2022-3719 at the AIAA AVIATION 2022 Forum, Chicago, IL and Virtual, June 27–July 1, 2022; received 5 June 2022; revision received 21 November 2022; accepted for publication 8 February 2023; published online 14 March 2023. Copyright © 2023 by Melike Nikbay. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*M.S. Student, Research Assistant, AeroMDO Laboratory, Faculty of Aeronautics and Astronautics, Maslak. Student Member AIAA.

†Professor, Faculty of Aeronautics and Astronautics, Maslak. Associate Fellow AIAA.

due to the computational expense of high-fidelity computational fluid dynamics (CFD) simulations and the high economic cost of experimental test setups. Data acquisition is not an option the entire time, which makes available data precious. Therefore, the main purpose of the usage of neural networks is to establish a surrogate model using an available physical solver output or experimental data in design optimization and uncertainty quantification studies. Some of the research applying ANNs in aerodynamic problems that are highly nonlinear, costly, complex, and generally black box can be found in the literature [23,29,33–38].

Moreover, even with increased computational resources, solving highly nonlinear and involved physical problems with high accuracy still poses a problem in engineering applications. In most cases, using a cheaper numerical method yields results with lower accuracy when compared to more expensive tools. Detailed design studies necessitate more accurate and precise frameworks despite the practicality of low-fidelity tools in preliminary conceptual design. At this point, multifidelity analysis that leverages the computational efficiency of low-fidelity models and the accuracy of high-fidelity models simultaneously within the same framework is an outstanding option; accordingly, multifidelity methods have been highly appealing in computational sciences and engineering. An elaborate multifidelity concept can be found in Ref. [39]. Moreover, a multifidelity analysis leverages at least two different fidelities of data for which one is superior to the other in terms of physical accuracy. The main motivation for the application of multifidelity methods is to decrease the computational burden of high-fidelity solvers or the expense of experimental tests. Because some engineering applications, such as design optimization and uncertainty analysis, require numerous function evaluations (even up to several million), it is not viable to use only high-fidelity (HF) partial differential equation solvers within a design process. On the other hand, designers must sacrifice a considerable amount of accuracy using only low-fidelity (LF) solvers, especially in complex problems for which low-fidelity tools cannot sufficiently capture the underlying physics. Multifidelity methods combine a small number of HF analysis data and a large number of LF analysis data to remarkably reduce the computational expenses while keeping the accuracy as high as in HF solutions. In the literature, multifidelity polynomial chaos expansion with orthogonal polynomials and cokriging (multifidelity Gaussian regression) that uses a maximum likelihood approach are extensively appealing in design optimization [40–45]. Regardless of the method, some of the areas that implement multifidelity analysis are structural design [46–48], aeroelasticity [49–51], aerodynamics [52,53], and dynamic systems [54,55].

In addition, a novel multifidelity approach with the assistance of deep neural networks was first proposed in the literature [56] in 2020. This method leveraged three fully connected neural networks to estimate low-fidelity output and linear and nonlinear correlations between a high-fidelity and a low-fidelity dataset with an autoregressive scheme. Thus, both low- and high-fidelity predictions could be made with a single multifidelity deep neural network (MFDNN). This approach has been embraced and applied in the aerospace community. In Ref. [24], the computational efficiency of the gradient-enhanced MFDNN was compared with the efficiencies of gradient-enhanced neural networks and multifidelity neural networks using analytical functions and an airfoil optimization problem. The authors of Ref. [57] investigated the efficacy of the MFDNN in the prediction of aerodynamic coefficients using LF inviscid and HF viscous airfoil flows, and they shared a comparison with the results of cokriging. A comprehensive shape optimization study of an airfoil and a wing–body configuration in the transonic regime was provided in Ref. [25]. A multifidelity Bayesian neural network was presented in Ref. [58] and was applied to noisy data with different fidelity levels; it was also able to solve inverse problems using partial differential equations.

One of the drawbacks of the MFDNN is that it only consists of affine layers and activation functions. Thus, the input of an MFDNN architecture must be provided in a vector form. However, the vectorization of input data may cause loss of information for a certain type of data due to the dislocation of highly correlated data points. This problem was experienced in image processing studies in the past. We can elaborate on this issue: each image is a tensor with a combination of pixels. Each pixel has a unique location on an image such that the pixels form a visual representation of an object by surrounding each other. Nevertheless, taking each row of pixels and concatenating them side by side to obtain a vector of pixels deteriorates the quality and meaning of the visual representation; in other words, it brings about a loss of information on the image. For computational sciences, two- and three-dimensional differentiable physical solution domains can be thought of as images. A vertex of a grid on a two- or three-dimensional spatial domain (let us say a flowfield) is mostly expected to be highly correlated with the surrounding vertices. Therefore, the representation of a flowfield with a vector notation disarrays high correlation regions and leads to the loss of gradient information in a quantity of interest. This issue may engender a reduction in neural network performance due to the lack of features. In addition, the representation of an involved flowfield may require a large number of matrix/tensor elements; thus, fully connected networks that are embedded in the MFDNN suffer the curse of dimensionality with the necessity of a large learnable parameter set to process these high-dimensional vectorized flowfields. As a result, both memory requirement and training time increase, regardless of the loss of data information.

This paper addresses the drawbacks of the MFDNN method stated earlier in this paper in the application of computational flowfields by proposing a multifidelity convolutional neural network (MFCNN). Convolutional neural networks (CNNs) [59] have been vastly put into practice in image and video processing due to their ability to extract local features. CNNs are expected to perform well with two-dimensional flowfield data and be incorporated into multifidelity analysis with an alike network architecture in the MFDNN. Instead of vectorizing two-dimensional flowfields, each flowfield is processed in a high-dimensional tensor form by an MFCNN. Therefore, a loss of information is prevented. Moreover, sparse interactions and parameter sharing of CNNs allow us to establish deep convolutional neural networks with a small number of learnable parameters, which alleviates the curse of dimensionality; so, they are preferable for high-dimensional data such as computational flowfields. Furthermore, an intermediate form between the MFDNN and MFCNN architectures, named the modified MFDNN, is also proposed and investigated in predictions using flowfields. The modified MFDNN leverages the MFDNN architecture with a combination of the dimensionality reduction feature of MFCNNs.

In the context of the paper, a multifidelity supercritical airfoil problem is addressed to evaluate the performances of three different multifidelity neural network architectures using computational flowfields: multifidelity deep neural networks presented in the literature, the modified multifidelity deep neural networks, and multifidelity convolutional neural networks. The generated dataset is used to make multifidelity predictions of aerodynamic coefficients. The performance of each method is evaluated according to its test accuracy. The obtained results prove that the proposed MFCNN architecture outclasses both the MFDNN and the modified MFDNN in the prediction of aerodynamic coefficients using flowfields.

In Sec. II, multifidelity analysis and different multifidelity neural network architectures are introduced. The dataset generation of the considered two-dimensional airfoil problem is presented in Sec. III. In Sec. IV, the investigation of all three methods on the multifidelity prediction of aerodynamic coefficients are provided. Eventually, Sec. V compares the obtained results of each method and presents a discussion. Section VI provides the concluding remarks.

II. Multifidelity Analysis

Multifidelity analysis methods are mostly adopted to diminish the computational burden of a set of high-fidelity simulations. These methods fuse a low number of expensive and physically more accurate data with a high number of cheap and less accurate data. In Ref. [60], multifidelity methods are categorized as adaptation, fusion, and filtering. In the context of the paper, the fusion-based multifidelity approach is discussed. The fusion principle requires first obtaining low-fidelity and high-fidelity data. Afterward, data with different fidelity levels are fused with a scheme such as cokriging, multifidelity polynomial chaos expansion, and multifidelity neural networks. A general expression for the multifidelity comprehensive correction model is given with Eq. (1):

$$y_H(x) = \rho(x) y_L + \delta(x) \tag{1}$$

where y_H and y_L, respectively, correspond to HF and LF data. Here, $\rho(x)$ and $\delta(x)$, respectively, denote the surrogate models for multiplicative and additive corrections. It should be noted that this approach can merely handle linearly correlated HF and LF data. Further information on multifidelity methods can be found in Ref. [39].

A. Multifidelity Deep Neural Networks

Multifidelity deep neural networks were recently developed as data-driven predictive modeling methods and were first introduced in Ref. [56]. In this paper, a generalized autoregressive method for multifidelity analysis was proposed as in Eq. (2):

$$y_H(x, y_L) = \mathcal{F}_l(x, y_L) + \mathcal{F}_{nl}(x, y_L) \tag{2}$$

where $\mathcal{F}_l(x, y_L)$ and $\mathcal{F}_{nl}(x, y_L)$ are, respectively, the linear and nonlinear correlation networks. To the best of the authors' knowledge, weighting these correlation networks was first proposed in Ref. [61]. The weighted autoregressive scheme is constructed as in Eq. (3):

$$y_H(x, y_L) = \omega \mathcal{F}_l(x, y_L) + (1 - \omega) \mathcal{F}_{nl}(x, y_L) \tag{3}$$

where ω is the linear correlation weight. In general, the type and strength of a correlation between two arbitrary engineering datasets are not ubiquitous. Thus, ω is a hyperparameter to be tuned. The proposed MFDNN architecture in Ref. [61] is illustrated in Fig. 1.

The proposed MFDNN architecture has three distinct neural networks, which are the low-fidelity deep neural network NN_L, the linear correlation network NN_{H_1}, and the nonlinear correlation network NN_{H_2}. To clarify, NN_L is used to approximate low-fidelity outputs with a given set of low-fidelity inputs. In addition, NN_{H_1} and NN_{H_2} are implemented to estimate the correlation between low-fidelity and high-fidelity datasets. In addition, linear layers in the MFDNN only process vectors and scalars, regardless of the dimension of data and target. For example, a flowfield represented with an n-dimensional tensor, $\boldsymbol{x} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_n}$, must be flattened to a one-dimensional vector of $\boldsymbol{x} \in \mathbb{R}^{D_1 D_2 \cdots D_n}$ to make it suitable for linear layers.

The correlation networks accept a combination of high-fidelity inputs and low-fidelity outputs as input in vector form. Let a high-fidelity input be $\boldsymbol{x_H} \in \mathbb{R}^{n_H}$ and a low-fidelity prediction be $\hat{\boldsymbol{y}}_{\boldsymbol{L}} \in \mathbb{R}^{m_p}$. Then, the input of correlation networks becomes $\boldsymbol{x_C} \in \mathbb{R}^{n_H + m_p}$, which is the concatenation of $\boldsymbol{x_H}$ and $\hat{\boldsymbol{y}}_{\boldsymbol{L}}$. Hence, high-fidelity inputs and low-fidelity predictions should be in proper shape to be concatenated and fed to correlation networks. In the literature, MFDNNs are constructed with only linear layers and hyperbolic tangent functions [24,56,57,61,62]. NN_L and NN_{H_2} have compositions including linear layers and activation functions, while NN_{H_1} only consists of linear layers due to the fact that it is supposed to catch the linear correlation between high- and low-fidelity datasets.

The training of the MFDNN is based on both low-fidelity and high-fidelity predictions. The loss function to be minimized is computed as in Eq. (4), where θ is a set of learnable model parameters, $\Omega(\theta)$ is a regularization term, and λ is the regularization strength. Therefore, the loss values resulting from low- and high-fidelity predictions are optimized simultaneously. For this study, L_2 regularization (also known as weight decay) is used as $\Omega(\theta)$. The main role of the regularization term is to prevent overfitting during training by bounding the norm of learnable parameters:

$$\mathcal{L}(y_L, y_H, \hat{y}_L, \hat{y}_H, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{y}_L, \hat{\boldsymbol{y}}_L) + \mathcal{L}(\boldsymbol{y}_H, \hat{\boldsymbol{y}}_H) + \lambda \Omega(\boldsymbol{\theta}) \tag{4}$$

For the numerical predictions discussed in this paper, the mean squared error loss function is useful to evaluate the prediction error. Then, Eq. (4) becomes the expression in Eq. (5):

$$\mathcal{L}(y_L, y_H, \hat{y}_L, \hat{y}_H, \boldsymbol{\theta}) = \frac{1}{m_p} (\|\boldsymbol{y}_L - \hat{\boldsymbol{y}}_L\|_2^2 + \|\boldsymbol{y}_H - \hat{\boldsymbol{y}}_H\|_2^2) + \lambda \Omega(\boldsymbol{\theta}) \tag{5}$$

B. Modified Multifidelity Deep Neural Networks

As explained in Sec. II.A, high-fidelity input and low-fidelity prediction vectors should have suitable dimensions so they can be given to the correlation networks. This condition restricts the simultaneous usage of high-fidelity inputs and low-fidelity predictions with a huge dimensional difference. Intuitively, the high-dimensional input vector dominates the input vector of the correlation networks; thus, low-fidelity predictions are ignored. To the best of the authors' knowledge, research articulating the proper vector dimensions for concatenation does not exist in the literature. Mathematically, let a high-fidelity input be in the $\boldsymbol{x_H} \in \mathbb{R}^{n_H}$ form and a low-fidelity prediction be $\hat{\boldsymbol{y}}_{\boldsymbol{L}} \in \mathbb{R}^{m_p}$ in vectorized forms where $m_p \ll n_H$, or vice versa. In this paper, we only consider the case of $m_p \ll n_H$. In this case, the correlation networks cannot learn the relation between the high-fidelity input and the low-fidelity prediction because n_H is a much higher value than m_p. The main reason for this issue is out of the scope of this study.

As a workaround for the considered problem, we propose the modified MFDNN, which includes a fully connected encoder incorporated into the MFDNN architecture. As is discussed, the main purpose of the encoder is to map a high-dimensional high-fidelity input to a lower-dimensional subspace such that the original size of the input becomes closer to the size of the low-fidelity predictions. The architecture of the modified MFDNN is given in Fig. 2. The only difference between the modified MFDNN and the MFDNN is a subnetwork named NN_E. Here, NN_E is a fully connected encoder to transform $\boldsymbol{x_H} \in \mathbb{R}^{n_H}$ to an encoded vector $\boldsymbol{x_{H,E}} \in \mathbb{R}^{n_{H,E}}$, where $n_{H,E} \approx m_p \ll n_H$. Thus, the input of the correlation networks is $\boldsymbol{x_C} \in \mathbb{R}^{n_{H,E} + m_p}$ instead of $\boldsymbol{x_C} \in \mathbb{R}^{n_H + m_p}$. As a result, the training and test experiments presented under Sec. IV show that the modified MFDNN performs better than the MFDNN.

Moreover, the size of the transformed high-fidelity input $n_{H,E}$ is a hyperparameter. An appropriate size should be selected to extract the

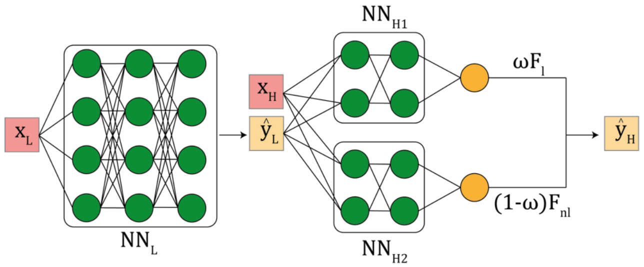

Fig. 1 A generic multifidelity deep neural network architecture.

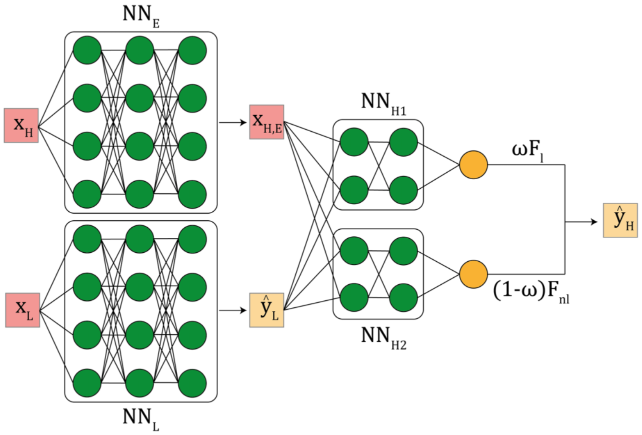

Fig. 2 A generic architecture of the modified MFDNN.

latent features of a given high-dimensional high-fidelity input without the loss of information.

C. Multifidelity Convolutional Neural Networks

As was mentioned before, the MFDNN uses input data only in a scalar or vector form, which may not be suitable for processing data with more than one dimension. Higher-dimensional inputs must be vectorized before giving them to a fully connected network, which causes a loss of information on data due to the dislocation of highly correlated regions. Vectorization of a flowfield around an airfoil is demonstrated in Fig. 3 to provide a better visual understanding.

To address this drawback, a coalescence of existing methods (the MFDNN and CNN) is proposed in this paper specifically for multifidelity predictive modeling of two-dimensional flowfields.

The autoregressive model defined in Eq. (3) is valid for this method as well. The only difference is the network where convolution layers are prevailing in the MFCNN. The multifidelity convolutional neural network architecture proposed in this paper is demonstrated in Fig. 4.

In the MFDNN, multifidelity predictions require high-fidelity inputs and low-fidelity predictions. Because both inputs and predictions are either scalar or vector, they can be concatenated. Thus, these two arrays are stacked and given to the correlation networks to estimate high-fidelity results. On the other hand, a new requirement arises with the usage of two-dimensional numerical data in the MFCNN. Unlike the MFDNN, as was discussed in Sec. II.B, high-fidelity inputs of the MFCNN are flowfields represented with four-dimensional tensors (including batch, channel, height, and width) and cannot be directly stacked with any arbitrary low-fidelity predictions unless inputs and predictions have the same dimension and sizes. Thus, the high-fidelity input size should be either compressed by an encoder or expanded by a decoder accordingly.

Like the MFDNN, the MFCNN comprises a low-fidelity estimator NN_L, a linear correlator NN_{H_1}, and a nonlinear correlator NN_{H_2}. Instead of the combination of affine and activation layers, NN_L of the MFCNN is the concatenation of convolution and activation layers. On top of that, the MFCNN brings an additional subnetwork, namely, NN_E. Here, NN_E is either an encoder with convolution layers or a decoder with transpose convolution layers. It aims to extract the features of a high-fidelity input and represent them in a different dimension. In this paper, aerodynamic quantities that are predicted have lower dimensions in comparison to high-fidelity inputs. Considering the data dimensions, an encoder version of NN_E is studied. Briefly, it performs dimensionality reduction for a given high-fidelity input into a proper size such that the resulting array can be concatenated with the low-fidelity predictions, similar to the discussion in the modified MFDNN.

III. Dataset Generation for Flow Around a Supercritical Airfoil

The open-source SU2 multiphysics suite [63] is integrated into this study to generate flowfield datasets. The RAE2822 airfoil problem with varying flow parameters is studied for multifidelity dataset generation.

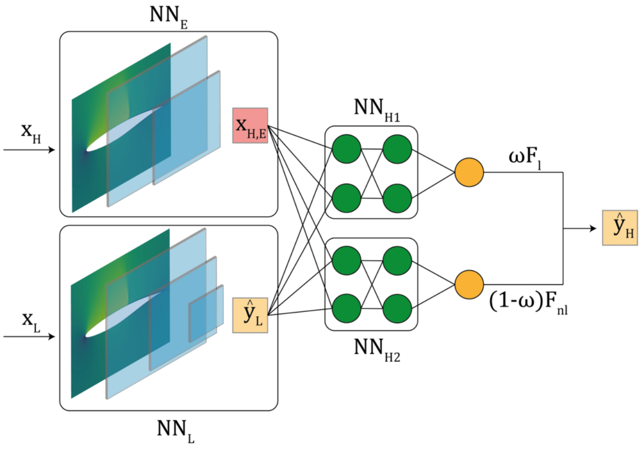

Fig. 4 A generic multifidelity convolutional neural network architecture.

For high-fidelity data, Reynolds-averaged Navier–Stokes (RANS) equations are solved by employing the Spalart–Allmaras (SA) turbulence equations; whereas inviscid Euler simulations are run for low-fidelity data generation, as in Refs. [64,65]. The generated flow domain around the airfoil has a roughly 50-chord-length radius. An O grid is used for both types of simulations. The y^+ value is set less than one over the surface of the airfoil for viscous simulations. The discretized flow domains are depicted in Fig. 5. The Jameson–Schmidt–Turkel (JST) flux differencing scheme, the flexible generalized minimum residual (FGMRES) linear solver, and the weighted least-squares gradient computation are employed with an adaptive Courant–Friedrichs–Lewy number. The convergence field is set as the density, and the convergence criterion of 10^{-8} is used. First, a validation study is performed at Mach 0.729 and a Reynolds number of 6.5×10^6 with respect to Ref. [66] while calibrating the angle of attack of 2.32 deg to fix the normal force coefficient of $C_N = 0.743$ in the current study. As reported in Ref. [67], we opt for using the fixed normal force coefficient for comparison rather than the lift coefficient to reduce the aberration in the pressure coefficient distributions obtained by numerical and experimental methods while calibrating the angle of attack to minimize this difference. The distinguishing parameters and computation costs experienced in the validation study using 36 CPU cores for the HF and LF simulations are provided in Table 1.

Pressure coefficients over the airfoil obtained with both HF and LF simulations are compared with experimental results [66] shared by NASA, and they are depicted in Fig. 6. As is seen, the HF pressure coefficients agree well with the experimental results; however, a remarkable difference occurs in the low-fidelity flowfield, especially at the end of the shock wave on the upper surface of the airfoil. Therefore, the impact of the fidelity level is obvious in favor of high-fidelity solutions, which come with a high computational burden. Mach contours of the validated solutions are represented in Fig. 7.

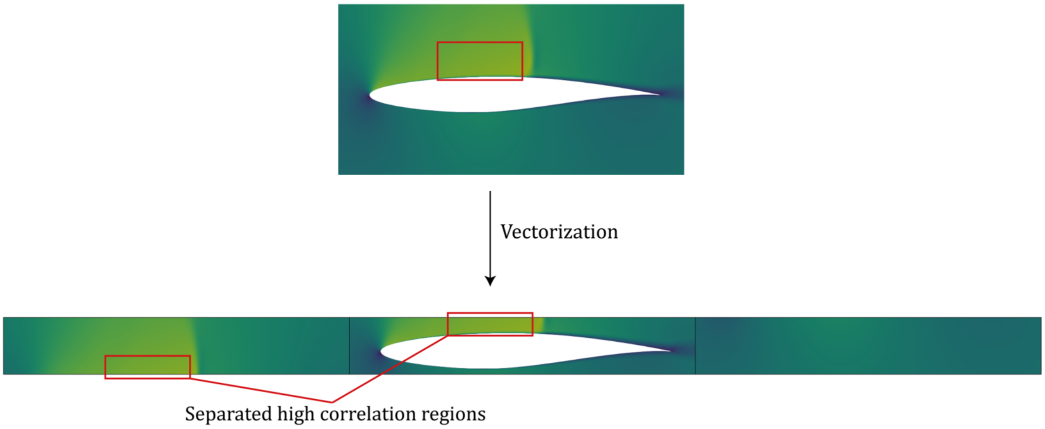

Fig. 3 A visual representation of the vectorization of a high-dimensional input.

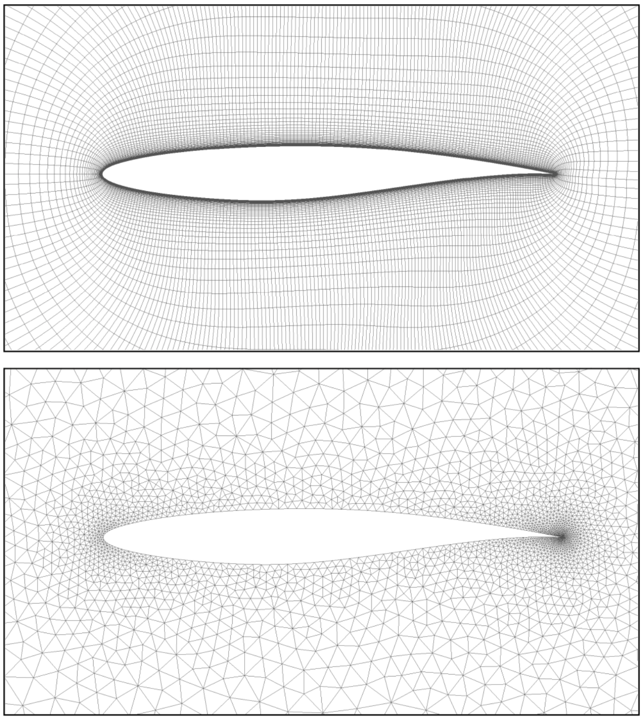

Fig. 5 The discretized flow domains: high fidelity (top), and low fidelity (bottom).

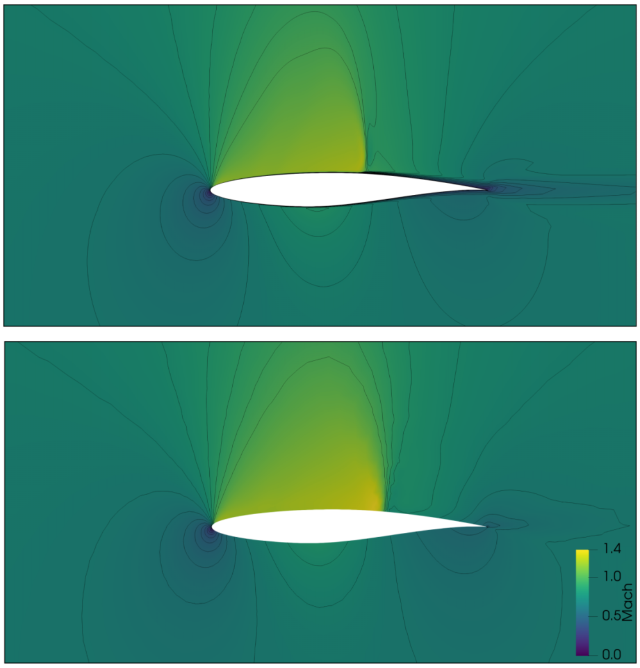

Fig. 7 The comparison of the Mach field for high-fidelity (top) and low-fidelity (bottom) RAE2822 flow solutions.

Table 1 Numerical differences of high-fidelity and low-fidelity simulations

Fidelity	Solver	Number of cells	Cost, s/analysis
High	RANS–SA	32,000	32.9
Low	Euler	10,000	1.7

Table 2 The lower bounds (LBs) and upper bounds (UBs) of flow parameters

Parameter	LB	UB
Mach	0.4	0.8
α, deg	−4	8
Re ($\times 10^6$)	5	7.5

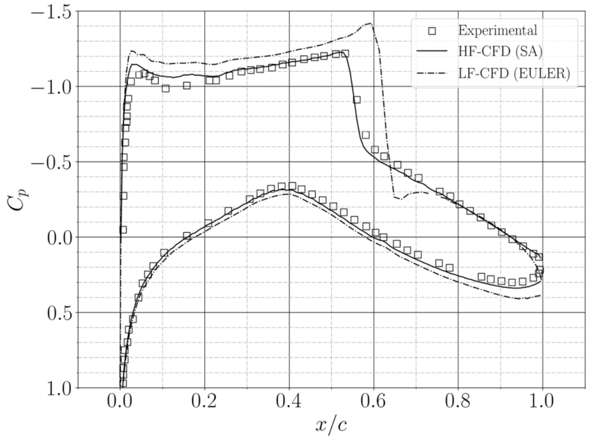

Fig. 6 The validation results for the RAE2822.

For dataset generation, the Mach number M, angle of attack α, and Reynolds number Re are considered as variables. A broad range is determined for each flow variable to allow complex flow phenomena to occur, such as flow separation and different shock waves. The lower and upper bounds of the considered variables are given in Table 2.

Furthermore, Halton quasi-random sampling [68] is employed to obtain high- and low-fidelity samples. Halton sampling uses a deterministic approach to generate sample points in a unit hypercube that seem randomly distributed using coprime numbers and is applied widely in multifidelity analysis. We run 200 high-fidelity and 300 low-fidelity simulations to obtain a dataset for this application. The number of high-fidelity samples is determined heuristically and more than is needed for the application considered in this paper. The main purpose of choosing a high number of high-fidelity data is to investigate the influence of the high-fidelity sample size on the performance of a multifidelity neural network.

To improve and expedite the heuristic neural network training approach, the inspection of the data has a crucial aspect. The correlation between the low-fidelity and high-fidelity datasets especially unveils the main concepts of employable neural networks. The correlation charts of the drag and lift coefficients are depicted in Fig. 8.

The viscous effects included in the RANS simulations are blatantly seen in Fig. 8. It can be seen that high-lift conditions create a distinct discrepancy between the inviscid low-fidelity and viscous high-fidelity solutions due to flow separation over the airfoil where the low-fidelity solver overestimates the lift coefficients C_L. A similar effect can also be observed on the drag coefficient C_D. For a transonic regime with a high angle-of-attack value, Euler solutions cannot capture the shock waves accurately under the given analysis conditions, causing high drag coefficient estimations. Furthermore, the lift coefficient has a mostly linear relation, as is expected in the lower C_L values. High-lift conditions occur at high angle-of-attack values where flow separation over the airfoil can only be seen in viscous high-fidelity solutions. On the other hand, the drag coefficient has a slightly nonlinear behavior. At first sight, shallow linear and nonlinear correlation networks seem to be enough to accurately predict the given aerodynamic coefficient datasets with a low number of high-fidelity data because the correlation between the two fidelity levels of the datasets is not complex.

Eventually, the dataset is preprocessed to make it suitable for neural network training according to the used methods. Pressure coefficient fields of the entire simulations are interpolated on a 64-by-64 Cartesian grid with linear interpolation, which was found

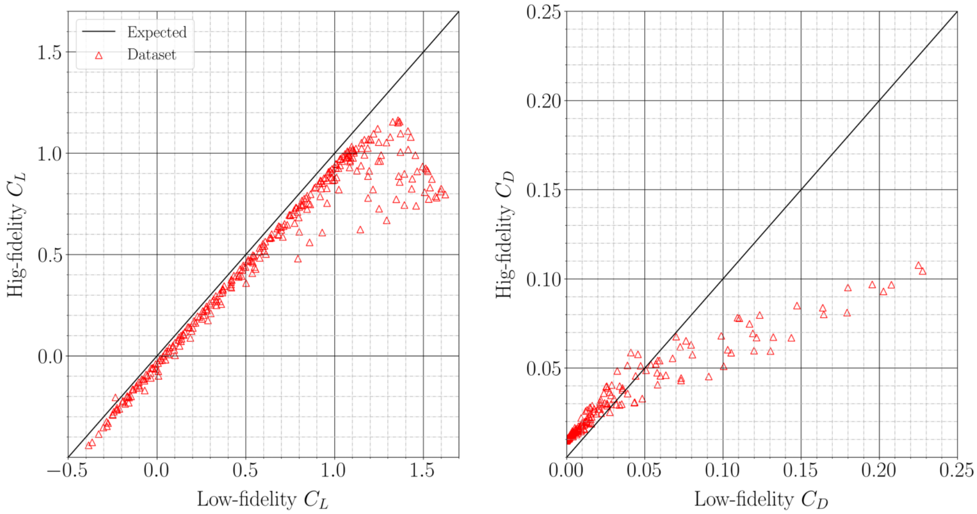

Fig. 8 The correlation between drag and lift coefficients of the parameter-varying RAE2822 airfoil dataset.

adequate for a two-dimensional airfoil application. In addition, the low- and high-fidelity targets are normalized with the minimum–maximum normalization scheme to the range of [0,1] to improve the numerical stability of training. The entire dataset is shuffled without disordering the input–target pairs and then split into training, validation, and test sets with the ratio of 80–10–10%. Finally, the constant batch size of 32 is used for partitioning the dataset to alleviate the memory requirement during training and testing.

IV. Aerodynamic Prediction from Flowfields Using Multifidelity Neural Networks

This section evaluates the multifidelity neural network architectures using the generated multifidelity supercritical airfoil dataset. For the MFDNN and the modified MFDNN, three similar architectures with varying learnable parameter set sizes are investigated. Additionally, the high-fidelity sample size is gradually increased for each model to observe the effect of the number of high-fidelity data points. The performance metrics for the evaluation of methods are determined as test accuracy in terms of the root-mean-squared error (RMSE), the relative percentage error (Rel. %) as given in Eq. (6), and the used high-fidelity dataset size with a given fixed low-fidelity sample size. Notice that the RMSE and the normalized RMSE yield the same exact error values because the prediction quantities lie in the range of [0,1]:

$$\text{Rel.\%} = \frac{\|y_H - \hat{y}_H\|_2}{\|y_H\|_2} \times 100 \tag{6}$$

A. Aerodynamic Prediction Using Multifidelity Deep Neural Networks

The generated supercritical airfoil dataset is first used to predict the aerodynamic lift and drag coefficients of the RAE2822 airfoil under varied flow conditions using the MFDNN. The low-fidelity pressure coefficient input dataset $\boldsymbol{x}_L \in \mathbb{R}^{300\times64\times64}$ and the high-fidelity pressure coefficient input dataset $\boldsymbol{x}_H \in \mathbb{R}^{200\times64\times64}$ are, respectively, vectorized into $\boldsymbol{x}_L \in \mathbb{R}^{300\times4096}$ and $\boldsymbol{x}_H \in \mathbb{R}^{200\times4096}$ so that they can be processed using affine layers. The constructed MFDNN architectures are elaborated on in Tables 3 and 4, where n_θ is the total number of learnable parameters that each multifidelity neural network possesses. All of the MFDNNs use the same linear NN_{H_1} and nonlinear correlation NN_{H_2} networks but varying low-fidelity estimators NN_L.

As Table 3 reveals, MFDNNs with three different learnable parameter sizes are used. All the models have the same correlation networks: NN_{H_1} and NN_{H_2}. The parameter size is only changed with the alteration of the low-fidelity estimator NN_L. The initial learnable parameters are uniformly sampled, where the bounds are determined as given in Eq. (7):

Table 3 Detailed MFDNN low-fidelity estimator architectures used to predict aerodynamic coefficients

n_θ	Subnetwork	Layers	Out features
2.4×10^6	NN_L	Linear-ReLU	512
		Linear-ReLU	128
		Linear-ReLU	16
		Linear-ReLU	2
4.6×10^6	NN_L	Linear-ReLU	1024
		Linear-ReLU	128
		Linear-ReLU	16
		Linear-ReLU	2
10.9×10^6	NN_L	Linear-ReLU	2048
		Linear-ReLU	1024
		Linear-ReLU	128
		Linear-ReLU	2

Table 4 Detailed MFDNN correlator network architectures used to predict aerodynamic coefficients

Subnetwork	Layers	Out features
NN_{H_1}	Linear	32
	Linear	2
NN_{H_2}	Linear-ReLU	32
	Linear-ReLU	32
	Linear-ReLU	2

$$\text{bound} = \frac{1}{\sqrt{W_2}} \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{W_1\times W_2}$ is the weight of a layer, and W_2 is the second dimension of a weight. Thus, any model parameter in the MFDNN is initialized with the samples from $\boldsymbol{W} \sim \mathcal{U}(-\text{bound}, \text{bound})$ and $\boldsymbol{b} \sim \mathcal{U}(-\text{bound}, \text{bound})$, where $\boldsymbol{b}$ is the bias of a layer.

Two convergence criteria are set for training the networks. Training continues if the total number of epochs is less than 3500 or the training mean-squared error (MSE) loss is more than 10^{-4}. These criteria are determined after a couple of training trials. The learning curves get flattened, and the learning almost stops between 3500–4000 epochs. Therefore, we predetermined the total number of epochs as 3500. In addition, a single Nvidia GTX 1060 GPU is used to speed up the training. A constant learning rate of 10^{-4} and a weight decay of 10^{-5} are used to update the model parameters using the Adam stochastic optimization method [69]. The same training setup is used for five cases of an increasing high-fidelity sample size and a

Table 5 The achieved MF RMSE and relative percentage error values of predictions using the MFDNN

		NS_H				
n_θ		10	50	100	150	200
2.4×10^6	MF RMSE	0.3912	0.4803	0.4756	0.1886	0.4727
	Rel. %	84.21	103.38	102.37	40.60	101.75
4.6×10^6	MF RMSE	0.4059	0.4849	0.4673	0.1724	0.4877
	Rel. %	87.36	104.38	100.59	37.12	104.97
10.9×10^6	MF RMSE	0.3953	0.4766	0.4585	0.2390	0.4769
	Rel. %	85.08	102.60	98.69	51.46	102.64

fixed low-fidelity sample size of 300 to investigate the effect of the number of high-fidelity data. The RMSE and relative percentage error of the multifidelity predictions of the aerodynamic coefficients are tabulated in Table 5, where NS_H is the high-fidelity sample size in the training dataset. The results show that the MFDNN does not provide acceptable multifidelity predictions of aerodynamic coefficients due to the incapability of learning the correlation. This is because a single high-fidelity input has a much higher number of data points (i.e., 4096) when compared to a low-fidelity prediction: $\hat{\boldsymbol{y}}_L \in \mathbb{R}^2$. Intuitively, the high-fidelity input is the dominant information in the concatenated vector. This situation brings about the neural network ignoring low-fidelity predictions. A remedy for this issue is addressed with the modified MFDNN.

B. Aerodynamic Prediction Using the Modified Multifidelity Deep Neural Networks

The proposed modified MFDNN is an enhanced version of the MFDNN for processing high-dimensional inputs. It brings an additional subnetwork to transform a high-fidelity input to a different dimension; hence, it does not dominate low-fidelity predictions in correlation computations. Similar to the MFDNN application, the low-fidelity pressure coefficient input dataset $\boldsymbol{x}_L \in \mathbb{R}^{300\times64\times64}$ and the high-fidelity pressure coefficient input dataset $\boldsymbol{x}_H \in \mathbb{R}^{200\times64\times64}$ are, respectively, vectorized into $\boldsymbol{x}_L \in \mathbb{R}^{300\times4096}$ and $\boldsymbol{x}_H \in \mathbb{R}^{200\times4096}$. We investigated the transformed high-fidelity input sizes of 4, 8, and 16 to observe the effect of extracted feature sizes on the multifidelity predictions. Because a low-fidelity prediction is in the shape of $\hat{\boldsymbol{y}}_L \in \mathbb{R}^2$, the concatenated high-fidelity input and low-fidelity prediction vector that are given to the correlator networks have shapes of $\boldsymbol{x}_C \in \mathbb{R}^6$, $\boldsymbol{x}_C \in \mathbb{R}^{10}$, and $\boldsymbol{x}_C \in \mathbb{R}^{18}$. When compared to the MFDNN application where the concatenated vector is $\boldsymbol{x}_C \in \mathbb{R}^{4098}$, we could represent the information of the high-fidelity input and low-fidelity predictions more uniformly. As is seen in Table 5, models with different parameter sizes give similar outcomes in terms of the multifidelity (MF) RMSE. Thus, we used the same NN_L, NN_{H_1}, and NN_{H_2} of the model with $n_\theta = 2.4 \times 10^6$ in the modified MFDNN to save computation time. The employed NN_E peculiar to the modified-MFDNN architecture is elaborated on in Table 6. At the end of the day, the modified-MFDNN models have approximately 6.6×10^6 learnable parameters. Thus, NN_E brings an additional computation cost where the modified MFDNN has a roughly 2.75 times higher number of learnable parameters in comparison to the MFDNN with the identical architecture, except the NN_E.

Table 6 Employed NN_E architecture used in the modified-MFDNN application

	NN_E subnetwork
Layers	Out features
Linear	1024
Linear	256
Linear	64
Linear	4/8/16

Table 7 The achieved MF-RMSE and relative percentage error values of predictions using the modified MFDNN

Encoded		NS_H				
dimension		10	50	100	150	200
$n_{H,E} = 4$	MF RMSE	0.0739	0.0871	0.0968	0.0755	0.0977
	Rel. %	15.92	18.77	20.83	16.25	21.03
$n_{H,E} = 8$	MF RMSE	0.1378	0.1271	0.1316	0.0911	0.1251
	Rel. %	29.65	27.36	28.33	19.62	26.92
$n_{H,E} = 16$	MF RMSE	0.1360	0.0914	0.1166	0.0498	0.0730
	Rel. %	29.28	19.68	25.10	10.71	15.72

The same hyperparameter set is used as in the MFDNN application. The average physical training time is 240 s. The 60% increment in the computational cost, when compared to the MFDNN, is caused by the additional learnable parameters brought by NN_E. The MF-RMSE results of the prediction of the normalized aerodynamic coefficient and the relative percentage error obtained with the MFDNN are listed in Table 7. The table reveals that the modified MFDNN reduces the relative percentage error as compared to the MFDNN for predictions using flowfields. However, the pattern in the error values seems irregular with the increasing high-fidelity sample size. We consider two possibilities that cause this issue: The first possibility is that data encoding causes loss of information during downsampling. The second possibility is that the additional high-fidelity samples do not considerably contribute to the feature space that the modified MFDNN can process.

C. Aerodynamic Prediction Using Multifidelity Convolutional Neural Networks

Lastly, the proposed multifidelity framework, the MFCNN, is employed in the prediction of multifidelity aerodynamic coefficients. The low-fidelity pressure coefficient input dataset of $\boldsymbol{x}_L \in \mathbb{R}^{300\times64\times64}$ and the high-fidelity pressure coefficient input dataset of $\boldsymbol{x}_H \in \mathbb{R}^{200\times64\times64}$ are used. The linear and nonlinear correlator networks (NN_{H_1} and NN_{H_2}) are used in the MFCNN model as given in Table 4. The low-fidelity estimator NN_L is constructed with the combination of two-dimensional convolution layers and rectified linear unit (ReLU) activation functions, where the layer parameters are shared in Table 8 with the architecture NN_E. Moreover, NN_E is a convolutional encoder and does not consist of activation functions. Therefore, it is a linear function from the input domain to a latent range of $f: \mathbb{R}^{1\times64\times64} \rightarrow \mathbb{R}^{16}$, where the latent vector size of 16 is used. Eventually, the MFCNN architecture is composed of 5143 learnable parameters. The number of learnable parameters diminishes by about 99.8% when compared to the MFDNN by employing the CNN instead of the fully-connected neural networks (FCNN).

To investigate the efficacy of the proposed multifidelity neural network architecture, the same learning rate and weight decay as in the previous applications are employed with the L_1 loss function, which is minimized with the Adam optimizer. The obtained MF RMSE and relative percentage error values are provided in Table 9.

Comparing the errors, the MFCNN architecture outperforms both the MFDNN and the modified MFDNN in data-driven predictive modeling using flowfields as inputs. Additionally, unlike the MFDNN

Table 8 Employed NN_L and NN_E architecture used in the MFCNN application

Subnetwork	Layers	Filter/kernel/stride
NN_L	Conv2d-ReLU	2/4/2
	Conv2d-ReLU	4/4/2
	Conv2d-ReLU	8/3/2
	Conv2d-ReLU	16/3/2
Linear-ReLU-linear	16[a]/–/2[a]	
NN_E	Conv2d	1/4/4
	Conv2d	1/4/4

[a]The number of output features.

Table 9 The MF-RMSE values of predictions of the aerodynamic coefficients using the modified MFDNN

	NS_H				
Metric	10	50	100	150	200
MF RMSE	0.0958	0.0425	0.0416	0.0372	0.0275
Rel. %	20.62	9.5	8.96	8.00	5.91

and the modified MFDNN, increasing the high-fidelity sample size enhances the multifidelity predictions. Hence, the MFCNN learns the correlation better than the previously presented methods.

V. Results and Discussion

First, an overall comparison of the presented methods for predictive modeling using the flowfields is provided in this section. The best neural network model of each method is considered for comparison and visualization of the results. To begin with, the learning curves of each method with respect to the high-fidelity sample sizes are illustrated in Fig. 9.

It is seen in Fig. 9 that the fastest learning method is the MFCNN based on the first 500 epochs. In 3500 epochs, the minimum training and validation losses are achieved with the MFCNN models.

The test performances and training times of each method vs the high-fidelity sample sizes are compared in Fig. 10. The impact of the training performances is seen on the MF-RMSE values where the MFCNN yields more accurate predictions. On the multifidelity prediction side with the given results, a higher number of high-fidelity sample sizes only improves the multifidelity predictions of the MFCNN, which we interpret as a higher correlation learning capability. Thus, other methods are not able to distinguish the high-fidelity add-on samples for the presented application. On top of that, we demonstrate the predictions of the drag and lift coefficients for all three methods in Fig. 11 to provide a better visual understanding. It is conspicuous that the MFDNN is not capable of dealing with the combination of high-dimensional inputs and low-dimensional

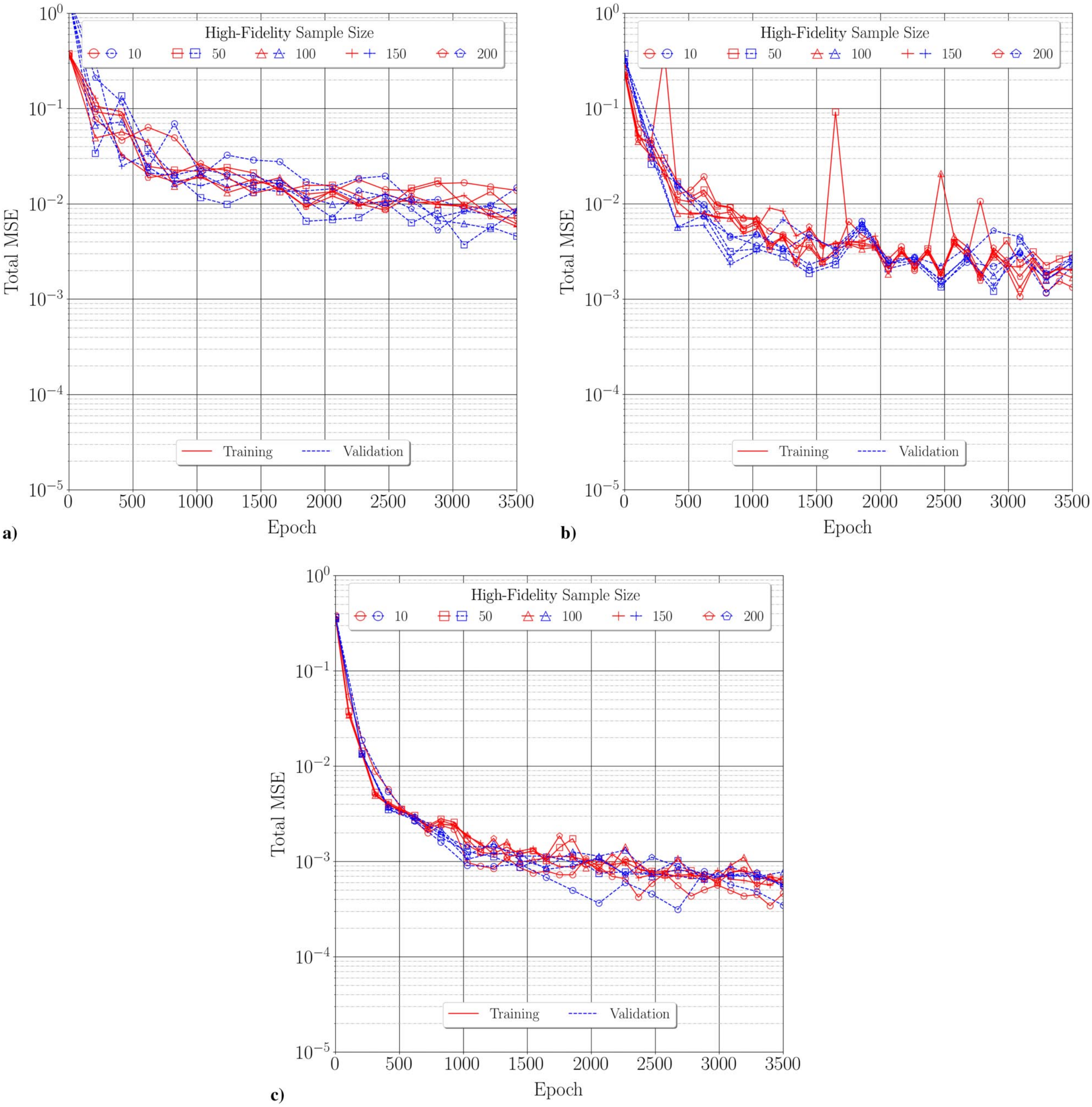

Fig. 9 Learning curves of multifidelity neural networks with respect to the high-fidelity sample size: a) MFDNN, b) Modified MFDNN, and c) MFCNN.

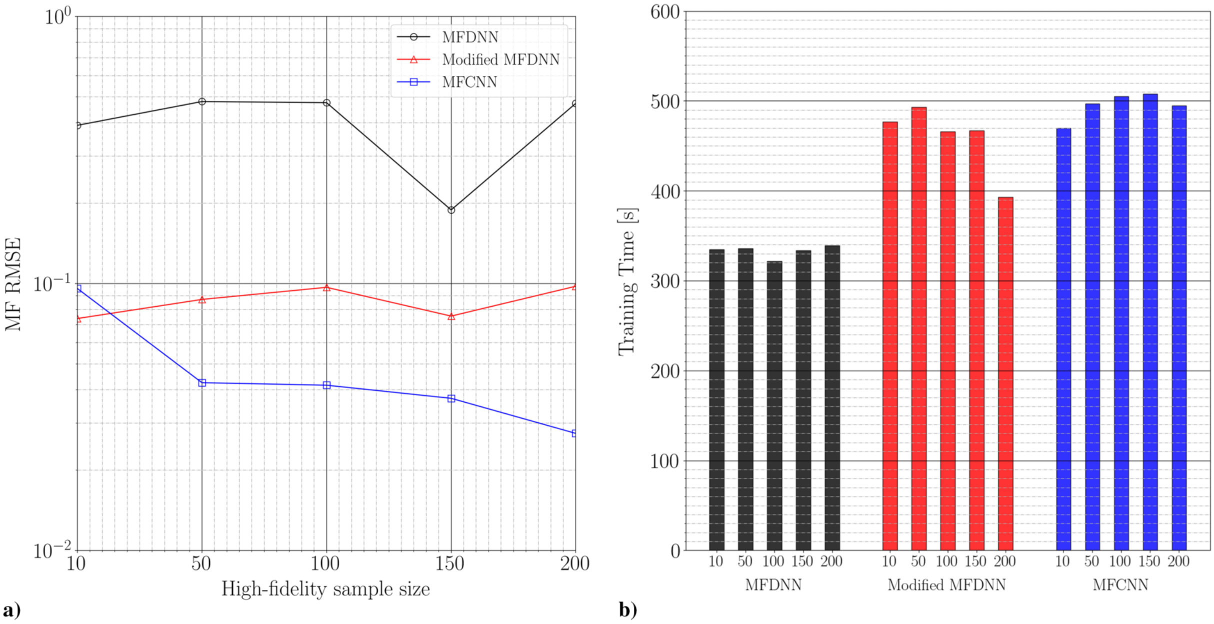

Fig. 10 Comparisons of a) MF-RMSE achieved with the test dataset and b) training time vs high-fidelity sample size.

low-fidelity predictions. On the other hand, processing high-dimensional inputs to represent them with lower-dimensional latent vectors significantly enhances the multifidelity prediction performances. Besides, the MFCNN outperforms the modified MFDNN by preventing vectorized flow fields and saving the features as flow fields are composed of highly correlated spatial variables.

Finally, the computational savings using the modified MFDNN and the MFCNN for different high-fidelity sample sizes are given in Table 10. The percentage savings are computed with Eqs. (8) and (9):

$$T_{\mathrm{mf}} = t_l \times NS_L + t_h \times NS_H \quad T_{\mathrm{hf}} = t_h \times (NS_L + NS_H) \tag{8}$$

Here, T is the total computation time needed for multifidelity analysis, and subscripts mf and hf, respectively, correspond to multifidelity and high-fidelity. The computational times of each low-fidelity and high-fidelity simulation are given with t_l and t_h, respectively. For the presented supercritical airfoil application, the computational simulation times are $t_l = 1.7$ s and $t_h = 32.9$ s, as given in Table 1. Remember that NS_L is the number of low-fidelity analyses and NS_H is the number of high-fidelity analyses:

$$\mathrm{Saving}\% = \frac{T_{\mathrm{hf}} - T_{\mathrm{mf}}}{T_{\mathrm{hf}}} \times 100 \tag{9}$$

VI. Conclusions

As concluding remarks, two novel multifidelity neural network architectures in predictive modeling are proposed using high-dimensional inputs. In total, three different neural network models are compared: the multifidelity deep neural network, the modified multifidelity deep neural network, and the multifidelity convolutional neural network. The modified MFDNN and MFCNN methods are the original contributions of this paper. A multifidelity two-dimensional aerodynamic problem is taken into account. The RAE2822 supercritical airfoil is analyzed using a low-fidelity Euler solver with a coarse grid and a high-fidelity RANS solver using the Spalart–Allmaras turbulence model with a relatively finer grid. Different flow conditions with varying Mach numbers, angles of attack, and Reynolds numbers are used to obtain flowfields around the airfoil. The main objective is the prediction of drag and lift coefficients under varying flow conditions in a transonic regime using pressure coefficient fields. Halton sampling is used to generate varying flow parameters, and 100 high-fidelity and 300 low-fidelity simulations are run using the SU2 flow solver. The obtained flowfields are preprocessed to transform them into a suitable form for neural network training and testing. Furthermore, three different MFDNN models with different learnable parameter sizes are investigated. Likewise, for the modified MFDNN, three different latent vector sizes are analyzed. In addition, training of all cases is run in the Python environment using PyTorch deep learning and optimization libraries. For the prediction of lift and drag coefficients, L_1 (known as the mean absolute error loss function) is employed with the renowned Adam optimizer. The same hyperparameter set is used for all cases. For each method, the test accuracy, the physical training time, and the number of the high-fidelity sample size are evaluated to quantify the performances. The results obtained with the two-dimensional airfoil problem reveal that the multifidelity deep neural network is not capable of predicting aerodynamic coefficients using the pressure coefficient fields; however, the multifidelity convolutional neural network outclasses the modified multifidelity neural network in terms of prediction accuracy on the test dataset with a given high-fidelity sample size, except for one case.

The results presented in this paper are preliminary outcomes of ongoing research. Only a set of hyperparameters, a single loss function, and an optimizer are considered for all investigated cases. Because heuristics approaches are essential for deep learning methods, there may be room for improvement in all of the results. A remedy for hyperparameter tuning can be Bayesian optimization to find the optimal multifidelity neural network architecture. In future works, the two proposed multifidelity neural network architectures, the modified MFDNN and the MFCNN, should be scrutinized using not only flowfields but any high-dimensional input with low-dimensional predictions as well. Although it was proved that the current architectures perform well with the considered application, the same framework should be employed within surrogate-based optimization problems and uncertainty analyses for a more comprehensive methodology comparison. On top of that, the correlation between the low- and high-fidelity cases in the presented application is mostly linear and slightly nonlinear. The prediction performances of the considered architectures should be analyzed with more complex cases where the correlation is highly nonlinear. Additionally, this research employs a fixed-sized dataset of which the sample size is determined with a constant computational budget. The cost of dataset generation can be further reduced with the implementation of active learning approaches to achieve the same multifidelity prediction performance.

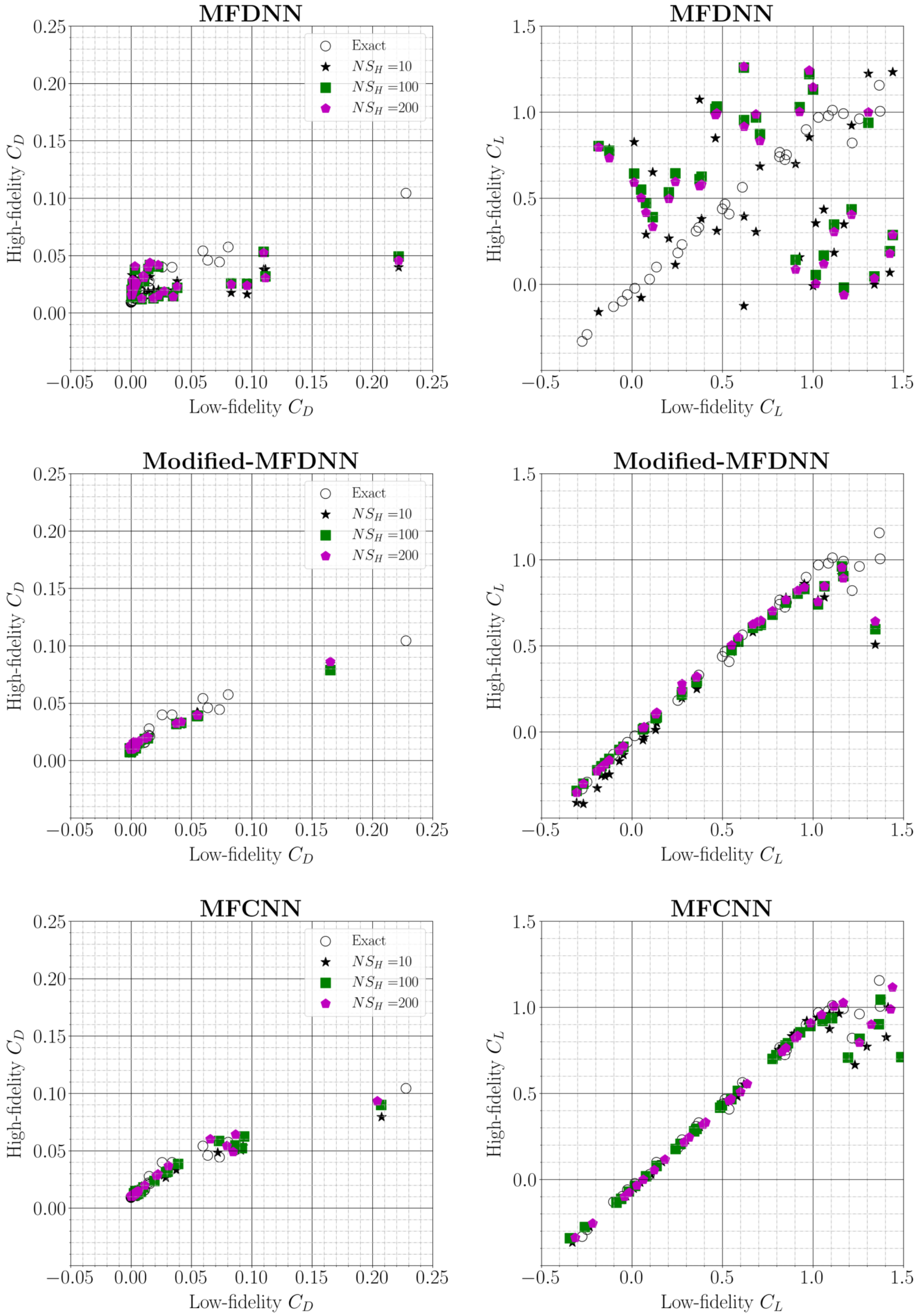

Fig. 11 **The multifidelity aerodynamic predictions of the MFDNN, the modified MFDNN, and the MFCNN using pressure coefficient fields.**

Table 10 The computational savings using the modified MFDNN and the MFCNN

NS_H	Savings, %
10	91.8
50	81.3
100	71.1
150	63.2
200	56.7

Acknowledgments

The authors would like to acknowledge Scientific and Technological Research Council of Türkiye (TUBITAK) for the research grant provided under the 218M471 TUBITAK 1001 project titled “Development of Multifidelity and Multidisciplinary Methodologies Integrating Sonic Boom, Aeroelasticity and Propulsion System for Supersonic Aircraft Design.” The second author would like to acknowledge the Istabul Technical University Scientific Research Projects Coordination Center (ITU-BAP) Scientific Research Projects Coordination Center for the support provided under the FHD-2023-44365 project titled “Multi-Fidelity and Multi-Disciplinary Design Optimization of a Low Boom Supersonic Transport Aircraft.” The authors would like to express their gratitude to General Electric Gas Power for the research grant titled “Development of Convolutional Neural Network Based Predictive Modeling Code.”

References

[1] Min, S., Lee, B., and Yoon, S., “Deep Learning in Bioinformatics,” *Briefings in Bioinformatics*, Vol. 18, No. 5, 2016, pp. 851–869. https://doi.org/10.1093/bib/bbw068

[2] Spencer, M., Eickholt, J., and Cheng, J., "A Deep Learning Network Approach to Ab Initio Protein Secondary Structure Prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 12, No. 1, 2015, pp. 103–112.
https://doi.org/10.1109/tcbb.2014.2343960

[3] Cao, R., Bhattacharya, D., Hou, J., and Cheng, J., "DeepQA: Improving the Estimation of Single Protein Model Quality with Deep Belief Networks," *BMC Bioinformatics*, Vol. 17, No. 1, 2016, pp. 1–9.
https://doi.org/10.1186/s12859-016-1405-y

[4] Toms, B. A., Barnes, E. A., and Ebert-Uphoff, I., "Physically Interpretable Neural Networks for the Geosciences: Applications to Earth System Variability," *Journal of Advances in Modeling Earth Systems*, Vol. 12, No. 9, 2020, Paper e2019MS002002.
https://doi.org/10.1029/2019ms002002

[5] Mamalakis, A., Ebert-Uphoff, I., and Barnes, E. A., "Neural Network Attribution Methods for Problems in Geoscience: A Novel Synthetic Benchmark Dataset," Preprint, Submitted 18 March 2021, https://arxiv.org/abs/2103.10005.

[6] Lary, D. J., Alavi, A. H., Gandomi, A. H., and Walker, A. L., "Machine Learning in Geosciences and Remote Sensing," *Geoscience Frontiers*, Vol. 7, No. 1, 2016, pp. 3–10.
https://doi.org/10.1016/j.gsf.2015.07.003

[7] Liu, Y., Li, L., Zhao, S., and Song, S., "A Global Surrogate Model Technique Based on Principal Component Analysis and Kriging for Uncertainty Propagation of Dynamic Systems," *Reliability Engineering and System Safety*, Vol. 207, March 2021, Paper 107365, https://www.sciencedirect.com/science/article/pii/S0951832020308541.
https://doi.org/10.1016/j.ress.2020.107365

[8] Zhou, T., and Peng, Y., "Kernel Principal Component Analysis-Based Gaussian Process Regression Modelling for High-Dimensional Reliability Analysis," *Computers and Structures*, Vol. 241, Dec. 2020, Paper 106358, https://www.sciencedirect.com/science/article/pii/S0045794920301619.
https://doi.org/10.1016/j.compstruc.2020.106358

[9] Iuliano, E., and Quagliarella, D., "Proper Orthogonal Decomposition, Surrogate Modelling and Evolutionary Optimization in Aerodynamic Design," *Computers and Fluids*, Vol. 84, Sept. 2013, pp. 327–350, https://www.sciencedirect.com/science/article/pii/S0045793013002223.
https://doi.org/10.1016/j.compfluid.2013.06.007

[10] Zimmermann, R., "Gradient-Enhanced Surrogate Modeling Based on Proper Orthogonal Decomposition," *Journal of Computational and Applied Mathematics*, Vol. 237, No. 1, 2013, pp. 403–418, https://www.sciencedirect.com/science/article/pii/S0377042712002452.
https://doi.org/10.1016/j.cam.2012.06.010

[11] Dolci, V., and Arina, R., "Proper Orthogonal Decomposition as Surrogate Model for Aerodynamic Optimization," *International Journal of Aerospace Engineering*, Vol. 2016, Sept. 2016, pp. 1–15.
https://doi.org/10.1155/2016/8092824

[12] Hess, M. W., Quaini, A., and Rozza, G., "A Data-Driven Surrogate Modeling Approach for Time-Dependent Incompressible Navier–Stokes Equations with Dynamic Mode Decomposition and Manifold Interpolation," Preprint, Submitted 26 Jan. 2022, https://arxiv.org/abs/2201.10872.
https://doi.org/10.48550/ARXIV.2201.10872

[13] Alla, A., and Kutz, J. N., "Nonlinear Model Order Reduction via Dynamic Mode Decomposition," *SIAM Journal on Scientific Computing*, Vol. 39, No. 5, 2017, pp. B778–B796.
https://doi.org/10.1137/16M1059308

[14] Xiong, F., Xue, B., Yan, Z., and Yang, S., "Polynomial," *2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, IEEE Publ., Piscataway, NJ, 2011, pp. 868–873.
https://doi.org/10.1109/ICQR2MSE.2011.5976745

[15] Kim, N. H., Wang, H., and Queipo, N. V., "Efficient Shape Optimization Under Uncertainty Using Polynomial Chaos Expansions and Local Sensitivities," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 1112–1116.

[16] Keshavarzzadeh, V., Fernandez, F., and Tortorelli, D. A., "Topology Optimization Under Uncertainty via Non-Intrusive Polynomial Chaos Expansion," *Computer Methods in Applied Mechanics and Engineering*, Vol. 318, May 2017, pp. 120–147, https://www.sciencedirect.com/science/article/pii/S0045782516313019.
https://doi.org/10.1016/j.cma.2017.01.019

[17] Xing, J., Luo, Y., and Gao, Z., "A Global Optimization Strategy Based on the Kriging Surrogate Model and Parallel Computing," *Structural and Multidisciplinary Optimization*, Vol. 62, No. 1, 2020, pp. 405–417.
https://doi.org/10.1007/s00158-020-02495-6

[18] Gaspar, B., Teixeira, A., and Guedes Soares, C., "Adaptive Surrogate Model with Active Refinement Combining Kriging and a Trust Region Method," *Reliability Engineering and System Safety*, Vol. 165, Sept. 2017, pp. 277–291, https://www.sciencedirect.com/science/article/pii/S0951832016301892.
https://doi.org/10.1016/j.ress.2017.03.035

[19] Julian, K. D., and Kochenderfer, M. J., "Reachability Analysis for Neural Network Aircraft Collision Avoidance Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 6, 2021, pp. 1132–1142.
https://doi.org/10.2514/1.G005233

[20] Hovell, K., and Ulrich, S., "Deep Reinforcement Learning for Spacecraft Proximity Operations Guidance," *Journal of Spacecraft and Rockets*, Vol. 58, No. 2, 2021, pp. 254–264.
https://doi.org/10.2514/1.A34838

[21] Brittain, M. W., Yang, X., and Wei, P., "Autonomous Separation Assurance with Deep Multi-Agent Reinforcement Learning," *Journal of Aerospace Information Systems*, Vol. 18, No. 12, 2021, pp. 890–905.
https://doi.org/10.2514/1.I010973

[22] van Rooijen, S. J., Ellerbroek, J., Borst, C., and van Kampen, E., "Toward Individual-Sensitive Automation for Air Traffic Control Using Convolutional Neural Networks," *Journal of Air Transportation*, Vol. 28, No. 3, 2020, pp. 105–113.
https://doi.org/10.2514/1.D0180

[23] Tekaslan, H. E., Imrak, R., and Nikbay, M., "Reliability Based Design Optimization of a Supersonic Engine Inlet," *AIAA Propulsion and Energy 2021 Forum*, AIAA Paper 2021-3541, 2021.
https://doi.org/10.2514/6.2021-3541

[24] Nagawkar, J. R., Leifsson, L. T., and He, P., "Aerodynamic Shape Optimization Using Gradient-Enhanced Multifidelity Neural Networks," *AIAA SciTech 2022 Forum*, AIAA Paper 2022-2350, 2022.
https://doi.org/10.2514/6.2022-2350

[25] Zhang, X., Xie, F., Ji, T., Zhu, Z., and Zheng, Y., "Multi-Fidelity Deep Neural Network Surrogate Model for Aerodynamic Shape Optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 373, Jan. 2021, Paper 113485.
https://doi.org/10.1016/j.cma.2020.113485

[26] Wang, Y., Liu, T., Zhang, D., and Xie, Y., "Dual-Convolutional Neural Network Based Aerodynamic Prediction and Multi-Objective Optimization of a Compact Turbine Rotor," *Aerospace Science and Technology*, Vol. 116, Jan. 2021, Paper 106869, https://www.sciencedirect.com/science/article/pii/S1270963821003795.
https://doi.org/10.1016/j.ast.2021.106869

[27] Marepally, K., Jung, Y. S., Baeder, J., and Vijayakumar, G., "Uncertainty Quantification of Wind Turbine Airfoil Aerodynamics with Geometric Uncertainty," *Journal of Physics: Conference Series*, Vol. 2265, No. 4, 2022, Paper 042041.
https://doi.org/10.1088/1742-6596/2265/4/042041

[28] Yin, J., and Du, X., "Uncertainty Quantification by Convolutional Neural Network Gaussian Process Regression with Image and Numerical Data," AIAA Paper 2022-1100, 2022.
https://doi.org/10.2514/6.2022-1100

[29] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., "Inverse Design of Airfoil Using a Deep Convolutional Neural Network," *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 993–1003.
https://doi.org/10.2514/1.j057894

[30] Sun, G., Sun, Y., and Wang, S., "Artificial Neural Network Based Inverse Design: Airfoils and Wings," *Aerospace Science and Technology*, Vol. 42, April–May 2015, pp. 415–428, https://www.sciencedirect.com/science/article/pii/S1270963815000644.
https://doi.org/10.1016/j.ast.2015.01.030

[31] Lei, R., Bai, J., Wang, H., Zhou, B., and Zhang, M., "Deep Learning Based Multistage Method for Inverse Design of Supercritical Airfoil," *Aerospace Science and Technology*, Vol. 119, Dec. 2021, Paper 107101, https://www.sciencedirect.com/science/article/pii/S1270963821006118.
https://doi.org/10.1016/j.ast.2021.107101

[32] Wang, C., Wang, S., Wang, L., Cao, C., Sun, G., Li, C., and Yang, Y., "Framework of Nacelle Inverse Design Method Based on Improved Generative Adversarial Networks," *Aerospace Science and Technology*, Vol. 121, Feb. 2022, Paper 107365, https://www.sciencedirect.com/science/article/pii/S1270963822000396.
https://doi.org/10.1016/j.ast.2022.107365

[33] Yu, J., and Hesthaven, J. S., "Flowfield Reconstruction Method Using Artificial Neural Network," *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 482–498.
https://doi.org/10.2514/1.j057108

[34] Peng, W., Zhang, Y., and Desmarais, M., "Spatial Convolution Neural Network for Efficient Prediction of Aerodynamic Coefficients," *AIAA SciTech 2021 Forum*, AIAA Paper 2021-0277, 2021.
https://doi.org/10.2514/6.2021-0277

[35] Wang, Q., Medeiros, R. R., Cesnik, C. E., Fidkowski, K., Brezillon, J., and Bleecke, H. M., "Techniques for Improving Neural Network-Based

Aerodynamics Reduced-Order Models," *AIAA SciTech 2019 Forum*, AIAA Paper 2019-1849, 2019.
https://doi.org/10.2514/6.2019-1849

[36] Du, X., He, P., and Martins, J. R., "Rapid Airfoil Design Optimization via Neural Networks-Based Parameterization and Surrogate Modeling," *Aerospace Science and Technology*, Vol. 113, June 2021, Paper 106701.
https://doi.org/10.1016/j.ast.2021.106701

[37] Achour, G., Sung, W. J., Pinon-Fischer, O. J., and Mavris, D. N., "Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization," *AIAA SciTech 2020 Forum*, AIAA Paper 2020-2261, 2020.
https://doi.org/10.2514/6.2020-2261

[38] Tekaslan, H. E., Demiroglu, Y., and Nikbay, M., "Surrogate Unsteady Aerodynamic Modeling with Autoencoders and LSTM Networks," *AIAA SciTech 2022 Forum*, AIAA Paper 2022-0508, 2022.
https://doi.org/10.2514/6.2022-0508

[39] Fernández-Godino, M. G., Park, C., Kim, N. H., and Haftka, R. T., "Issues in Deciding Whether to Use Multifidelity Surrogates," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2039–2054.
https://doi.org/10.2514/1.j057750

[40] Liu, Y., Chen, S., Wang, F., and Xiong, F., "Sequential Optimization Using Multi-Level Cokriging and Extended Expected Improvement Criterion," *Structural and Multidisciplinary Optimization*, Vol. 58, No. 3, 2018, pp. 1155–1173.
https://doi.org/10.1007/s00158-018-1959-6

[41] Keane, A. J., "Cokriging for Robust Design Optimization," *AIAA Journal*, Vol. 50, No. 11, 2012, pp. 2351–2364.
https://doi.org/10.2514/1.j051391

[42] Nagawkar, J., Leifsson, L. T., and Du, X., "Applications of Polynomial Chaos-Based Cokriging to Aerodynamic Design Optimization Benchmark Problems," *AIAA SciTech 2020 Forum*, AIAA Paper 2022-0542, 2020.
https://doi.org/10.2514/6.2020-0542

[43] March, A., Willcox, K., and Wang, Q., "Gradient-Based Multifidelity Optimisation for Aircraft Design Using Bayesian Model Calibration," *Aeronautical Journal*, Vol. 115, No. 1174, 2011, pp. 729–738.
https://doi.org/10.1017/S0001924000006473

[44] Cheng, K., Lu, Z., and Zhen, Y., "Multi-Level Multi-Fidelity Sparse Polynomial Chaos Expansion Based on Gaussian Process Regression," *Computer Methods in Applied Mechanics and Engineering*, Vol. 349, June 2019, pp. 360–377.
https://doi.org/10.1016/j.cma.2019.02.021

[45] West, T. K., and Phillips, B. D., "Multifidelity Uncertainty Quantification of a Commercial Supersonic Transport," *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 491–500.
https://doi.org/10.2514/1.c035496

[46] Sun, G., Li, G., Zhou, S., Xu, W., Yang, X., and Li, Q., "Multi-Fidelity Optimization for Sheet Metal Forming Process," *Structural and Multidisciplinary Optimization*, Vol. 44, No. 1, 2010, pp. 111–124.
https://doi.org/10.1007/s00158-010-0596-5

[47] Yi, J., Cheng, Y., and Liu, J., "A Novel Fidelity Selection Strategy-Guided Multifidelity Kriging Algorithm for Structural Reliability Analysis," *Reliability Engineering and System Safety*, Vol. 219, March 2022, Paper 108247, https://www.sciencedirect.com/science/article/pii/S0951832021007250.
https://doi.org/10.1016/j.ress.2021.108247

[48] Yoo, K., Bacarreza, O., and Aliabadi, M. H. F., "A Novel Multi-Fidelity Modelling-Based Framework for Reliability-Based Design Optimisation of Composite Structures," *Engineering with Computers*, Vol. 38, No. 1, 2020, pp. 595–608.
https://doi.org/10.1007/s00366-020-01084-x

[49] Marques, A. N., Opgenoord, M. M. J., Lam, R. R., Chaudhuri, A., and Willcox, K. E., "Multifidelity Method for Locating Aeroelastic Flutter Boundaries," *AIAA Journal*, Vol. 58, No. 4, 2020, pp. 1772–1784.
https://doi.org/10.2514/1.j058663

[50] Soneda, K., Yokozeki, T., Imamura, T., and Tsushima, N., "Multifidelity Aeroelastic Model for Corrugated Morphing Structures," *Journal of Aircraft*, Vol. 60, No. 1, 2022, pp. 120–129.
https://doi.org/10.2514/1.c036692

[51] Rumpfkeil, M. P., and Beran, P. S., "Multifidelity Sparse Polynomial Chaos Surrogate Models Applied to Flutter Databases," *AIAA Journal*, Vol. 58, No. 3, 2020, pp. 1292–1303.
https://doi.org/10.2514/1.j058452

[52] Li, K., Kou, J., and Zhang, W., "Deep Learning for Multifidelity Aerodynamic Distribution Modeling from Experimental and Simulation Data," *AIAA Journal*, Vol. 60, No. 7, 2022, pp. 4413–4427.
https://doi.org/10.2514/1.j061330

[53] Pinti, O., Oberai, A. A., Healy, R., Niemiec, R. J., and Gandhi, F., "Multi-Fidelity Approach to Predicting Multi-Rotor Aerodynamic Interactions," *AIAA Journal*, Vol. 60, No. 6, 2022, pp. 3894–3908.
https://doi.org/10.2514/1.j060227

[54] Absi, G. N., and Mahadevan, S., "Simulation and Sensor Optimization for Multifidelity Dynamics Model Calibration," *AIAA Journal*, Vol. 58, No. 2, 2020, pp. 879–888.
https://doi.org/10.2514/1.j058485

[55] Huang, C., Anderson, W. E., Merkle, C. L., and Sankaran, V., "Multifidelity Framework for Modeling Combustion Dynamics," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2055–2068.
https://doi.org/10.2514/1.j057061

[56] 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, Jan. 2020, Paper 109020.
https://doi.org/10.1016/j.jcp.2019.109020

[57] He, L., Qian, W., Zhao, T., and Wang, Q., "Multi-Fidelity Aerodynamic Data Fusion with a Deep Neural Network Modeling Method," *Entropy*, Vol. 22, No. 9, 2020, Paper 1022.
https://doi.org/10.3390/e22091022

[58] Meng, X., Babaee, H., and Karniadakis, G. E., "Multi-Fidelity Bayesian Neural Networks: Algorithms and Applications," *Journal of Computational Physics*, Vol. 438, Aug. 2021, Paper 110361.
https://doi.org/10.1016/j.jcp.2021.110361

[59] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.
https://doi.org/10.1109/5.726791

[60] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," *SIAM Review*, Vol. 60, No. 3, 2018, pp. 550–591.
https://doi.org/10.1137/16m1082469

[61] Raissi, M., Perdikaris, P., and Karniadakis, G., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 378, Feb. 2019, pp. 686–707.
https://doi.org/10.1016/j.jcp.2018.10.045

[62] Chakraborty, S., "Transfer Learning Based Multi-Fidelity Physics Informed Deep Neural Network," *Journal of Computational Physics*, Vol. 426, Feb. 2021, Paper 109942.
https://doi.org/10.1016/j.jcp.2020.109942

[63] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., "SU2: An Open-Source Suite for Multiphysics Simulation and Design," *AIAA Journal*, Vol. 54, No. 3, 2016, pp. 828–846.
https://doi.org/10.2514/1.J053813

[64] Ren, J., Leifsson, L. T., Koziel, S., and Tesfahunegn, Y., "Multi-Fidelity Aerodynamic Shape Optimization Using Manifold Mapping," *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 2016-0419, 2016.
https://doi.org/10.2514/6.2016-0419

[65] Padron, A. S., Alonso, J. J., and Eldred, M. S., "Multi-Fidelity Methods in Aerodynamic Robust Optimization," *18th AIAA Non-Deterministic Approaches Conference*, AIAA Paper 2016-0680, 2016.
https://doi.org/10.2514/6.2016-0680

[66] Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil Rae 2822: Pressure Distributions, and Boundary Layer and Wake Measurements," Experimental Data Base for Computer Program Assessment, AGARD Rept. 138, 1979.

[67] Ronch, A. D., Panzeri, M., Drofelnik, J., and d'Ippolito, R., "Sensitivity and Calibration of Turbulence Model in the Presence of Epistemic Uncertainties," *CEAS Aeronautical Journal*, Vol. 11, No. 1, 2019, pp. 33–47.
https://doi.org/10.1007/s13272-019-00389-y

[68] Halton, J. H., "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals," *Numerische Mathematik*, Vol. 2, No. 1, 1960, pp. 84–90.
https://doi.org/10.1007/bf01386213

[69] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," Preprint, Submitted 22 Dec. 2014, https://arxiv.org/abs/1412.6980.

D. Allaire
Associate Editor