

PAPER • OPEN ACCESS

Supplementation of deep neural networks with simplified physics-based features to increase accuracy of plate fundamental frequency predictions

To cite this article: Nicholus R Clinkinbeard and Nicole N Hashemi 2024 Phys. Scr. 99 056010

View the article online for updates and enhancements.

You may also like

- Fast and accurate robotic optical detection of exfoliated graphene and hexagonal boron nitride by deep neural networks Young Jae Shin, Wheemyung Shin, Takashi Taniguchi et al.
- Improving robustness of a deep learningbased lung-nodule classification model of CT images with respect to image noise Yin Gao, Jennifer Xiong, Chenyang Shen et al.
- <u>An unsupervised convolutional neural</u> network method for estimation of intravoxel incoherent motion parameters Hsuan-Ming Huang

Physica Scripta

CrossMark

OPEN ACCESS

RECEIVED 5 December 2023 REVISED

29 March 2024

ACCEPTED FOR PUBLICATION 9 April 2024

PUBLISHED 18 April 2024

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Supplementation of deep neural networks with simplified physicsbased features to increase accuracy of plate fundamental frequency predictions

Nicholus R Clinkinbeard n and Nicole N Hashemi*

Department of Mechanical Engineering, Iowa State University, Ames, IA, 50011, United States of America * Author to whom any correspondence should be addressed.

E-mail: nclink@iastate.edu and nastaran@iastate.edu

Keywords: machine learning, deep neural networks, simplified physics, scarce data, generalizability, physics-informed

Abstract

PAPER

To improve predictive machine learning-based models limited by sparse data, supplemental physicsrelated features are introduced into a deep neural network (DNN). While some approaches inject physics through differential equations or numerical simulation, improvements are possible using simplified relationships from engineering references. To evaluate this hypothesis, thin rectangular plates were simulated to generate training datasets. With plate dimensions and material properties as input features and fundamental natural frequency as the output, predictive performance of a datadriven DNN-based model is compared with models using supplemental inputs, such as modulus of rigidity. To evaluate model accuracy improvements, these additional features are injected into various DNN layers, and the network is trained with four different dataset sizes. When evaluated against independent data of similar features to the training sets, supplementation provides no statisticallysignificant prediction error reduction. However, notable accuracy gains occur when independent test data is of material and dimensions different from the original training set. Furthermore, when physics-enhanced data is injected into multiple DNN layers, reductions in mean error from 33.2% to 19.6%, 34.9% to 19.9%, 35.8% to 22.4%, and 43.0% to 28.4% are achieved for dataset sizes of 261, 117, 60, and 30, respectively, demonstrating potential for generalizability using a data supplementation approach. Additionally, when compared with other methods-such as linear regression and support vector machine (SVM) approaches-the physics-enhanced DNN demonstrates an order of magnitude reduction in percentage error for dataset sizes of 261, 117, and 60 and a 30% reduction for a size of 30 when compared with a cubic SVM model independently tested with data divergent from the training and validation set.

1. Introduction

Owing to its ability—in the presence of sufficient data—to predict new outcomes by determining complex relationships among dataset parameters [1], machine learning (ML) enjoys increasing application for simulating physical phenomena in science and engineering [2, 3]. This is in addition to its use as a data mining and human behavioral study tool [4]. Unfortunately, models driven solely by data may not be conducive to generalization, particularly when such data are limited [5–7], which can lead to large prediction errors for input parameters outside the range of those used for model development [8]. Also limiting are inaccuracies from uncertainties inherent to empirical measurements. Finally, a purely data-driven ML-based model can act as a 'black box' where inputs and outputs are known, but the underlying process for deriving relationships among these remains a mystery [9]. This is an issue when a fundamental understanding of the principles that cause certain features to drive others is crucial.

Alternatively, popular analytical and numerical approaches—such as the finite element method [10] and computational fluid dynamics [11]—are leveraged significantly in science and engineering to make behavioral predictions. Unlike machine learning, these techniques work by solving the governing partial differential equations often rooted in first principles. Notably, however, while ML suffers from lack of generalization and conceptual insight into physical processes, an oft-overlooked deficiency of analytical approaches is their inability to overcome the presence of unknown physics. Because numerical modeling invariably includes simplifying assumptions, nontrivial phenomena driving input-output relationships in a real system are often left unaccounted [12]. Additionally, physics-based numerical models may require significant time and computational resources to conduct a single simulation and are subject to limitations such as initial conditions and linearity [13], limiting their usefulness for studying an array of design options.

Fortunately, deficiency for the physics-based model is where ML has an advantage; data derived from experimentation will-within measurement limitations and uncertainties-reflect the true physical state of a system [12]. Therefore, benefit may be gained by successful integration of ML with modeling, potentially leading to improved designs from more efficient computation of possible solutions. Indeed, studies such as those undertaken by Zhu et al [14, 15] for predicting total organic carbon (TOC) content of a reservoir, as well as Lu and Zhang [16] with the assessment of seismic velocity inversion, have successfully integrated physics models with ML, particularly neural networks. Nevertheless, available time and resources to develop a working predictive model may still be less than needed for integration of ML with numerical simulation. Consequently, the incorporation of *simplified* physical relations with ML becomes an attractive alternative [17]. And while still a rich field of research, integration of physics into ML models has been investigated using a variety of approaches, such as tailored loss functions [18] or reduced-order models [19]. This leads to the level of physics integration investigated herein, namely, simplified theories. This methodology-presented by Pawar et al [17]-takes advantage of fundamental physical relationships among features to form the basis of parameter augmentation. For example, the authors increased neural network-based model accuracy for predicting flow around an airfoil by considering fundamental fluid dynamic quantities—such as Reynolds number and angle of attack—and calculating these characteristics from existing features. Once computed, the physics-based (as opposed to purely measured) datapoints were inserted into an intermediate hidden neural network layer to supplement the original data. This approach ultimately showed increased prediction accuracy within the limitations of the underlying theory.

While perhaps not as accurate as may be found through integration with numerical-based or reduced-order models, a DNN implementation strategy that combines simplified physics with machine learning has the potential to provide solutions that are better than purely data-driven techniques, and—once a model has been built—results may compute faster than for physics-only approaches. This is important where time to build a computational model or availability of suitable software is lacking. Therefore, establishment of a simplified approach will prove invaluable for the quick design and implementation of a system.

In order to advance the use of simplified physics within a machine learning framework for sparse datasets, the study detailed herein shows that not only can supplementing datasets with additional features generated from simple physical relations improve DNN-based model predictions, but such physics-based information has varying impact to model predictive accuracy based on size of training dataset. Furthermore, this investigation shows that improvements are to be gained by reintegrating the same physical features into multiple DNN layers rather than simply using them once, and that different combinations of augmented layers produce notably different results, with multiple layers often outperforming those generated with physics introduced only into a single layer. Finally, infusion of a DNN with physics-based information is shown to have the greatest impact on model accuracy when used to predict behavior for data not within the scope of the original training sets. The end result is a methodology that shows at least modest improvement to DNN model accuracy and generalizability when an abundance of datasets is lacking, and one that may be enhanced through further study and hyperparameter tuning.

2. Methods

The objective of this study is to examine the effects on model predictive accuracy when integrating simplified physics into a sparsely-trained deep neural network (DNN) via data augmentation. Four specific aspects of this are investigated, as follows.

(1) *Type and number of physics relations added*. Three different simplified physics relations are used for data augmentation. The goal is to observe whether certain individual terms or combinations thereof provide more improvement than others to DNN-based model accuracy.



Figure 1. Schematic of a thin rectangular plate with simply-supported boundary conditions along all four edges, where (a) shows orthogonal (top) and planar (bottom) views of the plate having dimensions length, *l*, width, *w*, and thickness, *t*, and (b) demonstrates the mode shape for the simply-supported thin plate during excitation of its fundamental resonance.

- (2) *Effects of including physics at different and/or multiple layers of the DNN.* Different cases are developed to examine how inclusion of physics affects model development when injected into each layer singularly or into multiple layers for a given DNN. The purpose is to understand if focusing on a specific DNN layer (or layers) for injecting physics has a greater influence on predictive power of the resulting model.
- (3) *Effect of training dataset size on model accuracy improvement.* The third facet is of this study is to examine the impact of augmentation with simplified physics on increasing model accuracy when progressively smaller training datasets are available.
- (4) *Effect of simplified physics on predictive accuracy for inputs outside the original domain.* This final aspect is geared toward examining whether or not augmenting a DNN with simplified physics data can improve predictions when new input data is largely outside of the domain of the original, i.e., does the enhancement procedure developed herein create a model that can generalize better than the original? To understand the answer to this question, two basic datasets are evaluated for every combination of (1), (2), and (3) previously discussed—one having parameters within the original training dataset and the other having parameters that lie outside.

The physical phenomenon chosen for this investigation is resonance of a thin rectangular plate with four simply-supported edges, shown in figure 1(a). Using a DNN framework, a predictive model is developed for processing previously-measured data to accurately predict the fundamental resonant frequency of a new plate design using three easily measurable dimensional parameters (length, width, and thickness) and three fundamental material properties (density, Young's modulus, and Poisson's ratio). Figure 1(b) provides a schematic of the mode shape for such a simply-supported thin rectangular plate during excitation of its fundamental resonance.

Once predictive performance of the baseline DNN is established for four different sizes of training datasets, supplementary datapoints are generated based on the six input features (i.e., three dimensions and three material properties), and injected into the same DNN code, albeit at different layers. To assess performance of any one combination of augmenting variables/layer(s) of insertion, each DNN-based model is used to predict natural frequency of plates containing various combinations of input features for which the models were not trained. The final performance evaluation is based on the mean and median error between the predicted and actual natural frequencies. Each model iteration is assessed with respect to additional input data using similar materials and dimensions to the original, as well as for a material type and dimensions not part of the DNN training parameters.



Figure 2. Finite element representation of a rectangular plate with simply-supported boundary conditions along all four edges. The view in (a) shows the finite element mesh, which was developed using 4-node quad (SHELL181) elements, which have bending and membrane stiffness. The view in (b) is fringe plot of the fundamental mode shape, which is similar for all combinations of dimensions and materials discussed herein.

Table 1. Worst-case mesh metrics for all plates modeled using the finite
element method. All evaluated quantities are near the ideal value, indicating
a high mesh quality.

Metric	Actual value	Ideal value	Range
Aspect Ratio	1.07	1	≥1
Element Quality	0.980	1	0 to 1
Jacobian Ratio (Corner Nodes)	0.975	1	-1 to 1
Jacobian (Gauss Points)	0.985	1	-1 to 1
Max. Corner Angle	92.1	90°	0° to 180°

The following paragraphs briefly describe the generation and nature of the baseline data, theory behind the development of physics-based data, and the DNN implemented to study the effects of including physics-based information.

2.1. Training and testing dataset generation

The training and testing datasets for this study were generated through normal modes analysis using the finite element software ANSYS Version 2021, Release 1 [20]. Thin rectangular plates with four simply-supported edges having varying dimensions and material types were modeled, and their fundamental natural frequencies calculated. Figure 2 shows an example finite element mesh for the plates under study, as well as a sample contour plot of the fundamental mode shape corresponding to the frequency of interest. The mesh itself was scaled for each individual plate size and adjusted to minimize discretization and other element approximation errors. Table 1 provides relevent mesh metrics used to assess the finite element models.

As well as serving as inputs to the finite element models, the minimal dataset input parameters used for all ML-based model generation comprise three plate dimensions (thickness, width, and length), weight density, and two elastic properties (Young's modulus Poisson's ratio), for a total of six features. To provide a somewhat diverse—but limited—set of data used to train the models, plates were simulated using aluminum, FR-4, copper, magnesium, and stainless steel. Intrinsic and elastic properties for these materials are provided in table 2, while the diversity of dimensions used is given in table 3.

Using the aforementioned parameters, a total of 500 solutions were calculated. From this, 261 datapoints are reserved for model generation (training and validation), while the remaining 239 datapoints are retained for independently assessing model predictive accuracy.

4

Table 2. Materials used in the generation of training data, with their intrinsic and elastic property values. Note the wide range of material properties chosen.

Material	Weight density (kg/m ³)	Young's modulus (GPa)	Poisson's ratio
Aluminum	2,700	68	0.33
FR-4	1,900	14	0.12
Copper	8,940	110	0.343
Magnesium	1,800	45	0.35
Stainless Steel	7,900	200	0.27

Table 3. Dimensions of simply-supported rectangular plates used in the generation of training data. In total, five different combinations of planar dimensions were combined with each increment of plate thickness to give a total of 500 datasets. These 500 datasets were split into 261 for training/ testing and 239 for independent testing.

Dimension	Set	Value
	i.	0.0508 m×0.0508 m
	ii.	$0.0476 \text{ m} \times 0.0762 \text{ m}$
Planar length and width, $w \times l$	iii.	$0.133 \text{ m} \times 0.178 \text{ m}$
,	iv.	$0.152 \text{ m} \times 0.0672 \text{ m}$
	v.	$0.267 \text{ m} \times 0.222 \text{ m}$
Plate Thickness, t:		0.762 to 3.18 mm; increments of 0.127 mm

One final note regarding the training/testing dataset is the application of uncertainty. Since the data generated for this study are computational rather than physical, to simulate a possible 2% (+/-1%) variation due to meaurement uncertainty and imperfect physical sample, the final output feature—i.e., natural frequency —is adjusted as follows:

$$f_{n.adi} = f_n (0.99 + 0.02R) \tag{1}$$

where *R* is a random number between the values of 0 and 1 having an approximately Gaussian distribution, and $f_{n,adj}$ is the final adjusted natural frequency used at the output parameter. (For convenience, the adjusted natural frequency is simply referred to as f_n for the remainder of this study.) This addition of uncertainty was simply to emulate measured data to a degree and analysis of its effect is not a focus of the study.

2.2. Simplified physics: natural frequency of a thin plate

Simplified physics used for integration with the deep learning model is partly based on small-deflection plate theory, adapted from Steinberg [21]. Using this approach, the fundamental natural frequency, f_n , of the rectangular plate simply-supported along four edges shown in figure 1 is as follows:

$$f_n = \frac{\pi}{2} \left(\frac{Dt}{\rho} \right)^{1/2} \left[\frac{1}{w^2} + \frac{1}{l^2} \right]$$
(2)

where t is the plate thickness, ρ is the material mass density, w and l are the plate width and length, respectively, and D is flexural rigidity. Flexural rigidity is computed from thickness, Young's modulus, E, and Poisson's ratio, ν :

$$D = \frac{Et^3}{12(1-\nu^2)}$$
(3)

Note that this term—calculated independently from planar dimensions and boundary conditions becomes the primary physics-based relation used to augment DNN-based model generation. This is important because while the plate equation is simple enough to use to calculate fundamental frequencies for our current configuration, for the majority of real-world applications, physical boundary conditions are not easily classifiable and may be a mix of simple, clamped, or elastic, to name a few. However, flexural rigidity is relatively easy to estimate for any plate based on material elastic properties and plate thickness. Therefore, this approach can be adapted to any dimensional and boundary condition configuration.



Figure 5. Examples of deep neural networks used to generate predictive models for fundamental resonance of a rectangular plate simply-supported along all edges. Note the following common characteristics of these DNNs: six input features (plate thickness, *t*, length, *l*, width, *w*, density, ρ , Young's modulus, *E*, and Poisson's ratio, ν); four hidden layers with several neurons each; and one output layer with a single feature, i.e., fundamental natural frequency, *f*_n. (a) Basic DNN used to predict natural frequency of thin rectangular plates simply-supported along all four edges. (b) DNN with augmented data set generated from physical relationships among original data. Here the physics-based parameters are simply added to the baseline inputs to give up to nine total input features. (c) DNN with physics integrated into a single layer (Layer 3 shown). In this architecture, physics-based features are added to a hidden layer as additional inputs for that layer only. (d) DNN with physics integrated into multiple layers (Layers 2 through 4 shown). With this final example, physics-based features are inserted into the first and/or an early hidden layer and re-inserted at one or more subsequent layers.

Since flexural rigidity is only dependent on three features—plate thickness, Young's modulus, and Poisson's ratio—an attempt is made to involve the remaining dimensions and material density into simplified physics by considering the plate weight, *W*, calculated as

$$W = t \times w \times l \times \rho \tag{4}$$

Shear modulus, G, is used as a third and final supplemental term and is calculated as:

$$G = E/[2(1+\nu)]$$
(5)

Since equation (2) is based on thin plate theory for small deflections, shear modulus has no direct bearing on natural frequency calculations where such assumptions are present; rather, it is chosen purely for its inclusion of both Young's modulus and Poisson's ratio, providing a relationship between the two. Particularly, its presence is intended as an additional datapoint to inform the DNN of the relationship between elastic parameters.

2.3. Neural network architecture

Based on early experimentation, the basic predictive model chosen and developed for this study is a deep neural network comprising four hidden layers and one output layer, as diagrammed in figure 3(a). The optimization

IOP Publishing

constant throughout the study.		
Hyperparameter	Description/Value	
Hidden layer activation function	Rectified Linear Unit (ReLU)	
Output layer activation function	Linear	
Learning rate, α_t	0.01	
Batch size	80	
Epochs	500	
Number of neurons per layer	40	
Numerical stability parameter, $\hat{\epsilon}$	1e-6	

Table 4. Selected hyperparameters and their values. Note that for performance comparison purposes, hyperparameters remained constant throughout the study.

algorithm chosen for this study is *Adam*, introduced by Kingma and Ba [22]. Implementation of each DNN was conducted using Python 3.8.8 64-bit through the Spyder 4.2.5 interface, as provided by Anaconda Navigator 2.3.2 [23]. The code used for implementation was initially patterned after that developed by Pawar *et al* [17].

Loss function

Data used for training and validation of the neural networked-based models were configured into commaseparated-value(s) (CSV) format. The resulting data files contain labeled column headings corresponding to the six input parameters (plate thickness, length, width, density, Young's modulus, and Poisson's ratio) and one output parameter (fundamental resonance). For iterations where augmented data were used, additional columns were added, as necessary. Specifically, these include flexural rigidity (*D*), plate weight (*W*), and material shear modulus (*G*). In all cases, data rows—each representing a different plate configuration—were randomized within the CSV files in order to remove biasing and allow representative diverse sampling to be used for validation.

Prior to model generation, preprocessing was conducted to condition the data for use with the neural network. Specifically, values for all parameters (input and output) were normalized to a range of 0 to 1 using min-max scaling, as follows [24]:

$$X_{SC} = (X - X_{max})/(X_{max} - X_{min})$$
 (6)

Mean-Squared Error (MSE)

where X is the datapoint undergoing scaling for a given parameter, X_{max} is the maximum value for the given parameter within the dataset, X_{min} is the minimum parameter value for the given dataset, and X_{SC} is the final scaled value.

Hyperparameter selection was decided by trial and error rather than through a rigorous optimization process. For consistency, all DNN-based model generation is conducted using the parameters provided in table 4. Note that the training/testing data was parsed such that 20% was reserved for validation, with the balance used for model training.

2.4. Integration of physics into NN architecture

As previously discussed, the methodology chosen for integrating physics into the process of machine learning is to insert physics-based quantities as augmentative data into various DNN layers. Four basic approaches are taken, represented in figure 3:

- 1. Neural network with no physics terms (figure 3(a)),
- 2. insertion of physics into Layer 1 (figure 3(b)),
- 3. insertion of physics into Layers 2, 3, 4, or 5 individually (figure 3(c)), and
- 4. insertion of physics into multiple layers for each run (figure 4(c)).

These four strategies represent nine total DNNs: one basic network without additional physics terms and eight different ways of integrating physics into the baseline DNN.

Along with varying locations of physics integration within the DNN model, six different combinations of calculated parameters are examined, listed in table 5. The purpose behind this is to observe whether single or multiple variables show the most improvement when integrated into the DNN.

The final consideration for model development is amount of training data available, for which the following dataset sizes are used: 261 datapoints, 117 datapoints, 60 datapoints, and 30 datapoints. The largest dataset containing 261 points serves as a superset from which the 117, 60, and 30 are derived. The goal in examining



Figure 4. Example of averaging used for final model predictions. Note that as the number of averages increases, the final natural frequency value tends to stabilize at about 245 Hz, while the standard deviation exhibits a steadily reducing trend. This approach is used to lessen the amount of variation within natural frequency predictions. (Results shown are for baseline DNN-based model predictions of the natural frequency for a stainless-steel plate with dimensions 5.25 in x 7.00 in x 0.045 in. The DNN-based model for this example was trained with a dataset size of 261.).

Model	Physics	Code file name
Baseline Deep Neural Network	None	NNML.py
Physics-Guided Deep Neural Network 1	W, D	PGML.py
Physics-Guided Deep Neural Network 2	W	PGMLa.py
Physics-Guided Deep Neural Network 3	D	PGMLb.py
Physics-Guided Deep Neural Network 4	G	PGMLc.py
Physics-Guided Deep Neural Network 5	W, D, G	PGMLd.py
Physics-Guided Deep Neural Network 6	D, G	PGMLf.py

Table 5. Various models developed to predict natural frequency. Specifically, this entails the injection of different combinations of weight (*W*), flexural modulus (*D*), and shear modulus (*G*) at each DNN layer.

different training dataset sizes is to determine how much—if any—impact physics has on improving accuracy of the DNN when only scarce data are available.

Based on the various combinations described herein, the total number of models generated for this study is

(4 baseline DNN Models) + (9 insertion options) $\times (6 physics combinations) \times (4 training data sizes) =$ **220 models**(7)

Each of the final 220 models is used to compute multiple predictions of fundamental natural frequency while initialized with different random seeds, where the final output value is realized as an average of all computed values (each having equal weight). The purpose for this is to reduce variation in model predictions. In total, fifty values of natural frequency are computed for each model. This number of averages was chosen based on early trials demonstrating that—while not strictly true—the overall average would tend toward a quasi-stable value between 20 and 30 iterations with the baseline (non-augmented) neural network and for all four training dataset sizes. However, since not all datasets in the final study are reviewed for this particular trend, a factor of approximately 2x was chosen as the standard number of averages to better ensure stability. An example of the stability trend using the training dataset size of 261 is shown in figure 4.

Table 6. Independent Dataset 2 input

 features. Material elastic properties taken

 from Steinberg [21], while density is

 estimated. Planar length (*l*), planar width

 (*w*), and plate thickness (*t*) for the 101

 different plates are generated at random

 within the bounds shown.

Feature	Value
Density	$4,200 \text{ kg m}^{-3}$
Young's modulus	21 GPa
Poisson's ratio	0.18
Planar length, l	0.0518 to 0.250 m
Planar width, w	0.0512 to 0.250 m
Plate Thickness, t:	0.610 to 5.49 mm

2.5. Error

The final term of high importance computed for assessment of model accuracy is percentage error. Once a solution of fifty averages is attained, the error for each resultant datapoint is calculated as

$$Error = \frac{|f_{n,a} - f_{n,p}|}{f_{n,a}} \times 100\%$$
(8)

where $f_{n,a}$ is the actual natural frequency of the plate as determined through finite element modeling and $f_{n,p}$ is the natural frequency predicted by the DNN-based model. Based on this information, several basic quantities are available, such as mean error, median error, standard deviation of error. Thus, these are investigated for each predictive model.

The error terms for each physics-guided DNN model are evaluated against those for the baseline DNN without integrated physics. This provides insight into the overall improvement a particular physics-guided DNN has in predicting plate natural frequency. To ensure any conclusions are statistically significant, the two-sample t-Test assuming Gaussian distribution but unequal variances is conducted between the baseline DNN and each physics-enhanced DNN individually. Results are considered statistically significant for a two-tail p-value of 0.05 (5%) or less.

2.6. Independent assessment of physics-guided DNN model accuracy

Following development of predictive models through training and testing with original data having parameters from tables 2 and 3, each is tested using two independent datasets in order to assess prediction accuracy. The first dataset independently tested with each model—henceforth referred to as *Test Dataset* 1—contains the remaining 239 datapoints developed from finite element modeling of the aluminum, copper, FR-4, magnesium, and stainless steel plates. The second independent dataset—*Test Dataset* 2— consists of a series of 101 plates constructed from composite printed wiring board (PWB) material having the isotropic property values and random dimensions within the extreme parameters provided in table 6.

Selection of printed wiring board material properties for independent Test Dataset 2 was undertaken for two primary reasons. First, systems containing PWBs are often subjected to vibration environments in practice. As such, their use in this study provides a real-world rationale for development of a methodology to accurately predict natural frequencies of plates. Second, PWBs are often fabricated from a layup of copper and FR-4 material, both used in generation of the training dataset. As seen in a comparison of tables 2 and 6, PWB material properties lie between those for FR-4 and copper and therefore lie within the values used to develop the predictive DNN-based model.

3. Results

3.1. Natural frequency predictions using data similar to training

Figure 5 shows mean predictive error for each physics-guided DNN model when compared with the baseline DNN containing no physics. Test Dataset 1 is used for independent testing of each model. Note that mean error for the baseline DNN without physics is 2.1% for the largest training size of 261 datapoints, which is approximately equivalent to the range of uncertainty assumed for plate natural frequency measurements. As expected, baseline DNN models generated with decreased training dataset sizes of 117, 60, and 30 datapoints exhibit increasingly higher mean error, with values of 5.0%, 6.4%, and 19.3%, respectively. Figure 6 shows the corresponding median predictive error and variability for each physics-guided DNN model when compared



samples, (b) 117 training samples, (c) 60 training samples, and (d) 30 training samples. For each layer of augmentation, several different combinations of plate weight (W), flexural rigidity (D), and material shear modulus (G) are evaluated. Note that (d) is scaled to a larger ordinate value for clarity.

with the baseline DNN containing no physics. Note that median error for the baseline DNN without physics is 0.95% for the largest training size of 261 datapoints. As with mean error, baseline DNN models generated with decreased training dataset sizes of 117, 60, and 30 datapoints exhibit increasingly higher median error, with values of 2.0%, 3.6%, and 8.3%, respectively.

3.2. Natural frequency predictions using data dissimilar from training

Figure 7 compares mean predictive error for the physics-guided DNN models to the baseline DNN when independently tested with Test Dataset 2. Baseline mean error values are 33.2%, 34.9%, 35.8%, and 43.0% for dataset sizes of 261, 117, 60, and 30 points, respectively. Figure 8 shows the corresponding median predictive error and variability for the physics-guided DNN models when independently tested with Test Dataset 2. Baseline median error values are 30.4%, 30.8%, 32.0%, and 36.5% for dataset sizes of 261, 117, 60, and 30 points, respectively.

3.3. Comparison with other regression methods

One further evaluation of the methodology and models developed section 2 was conducted to assess their performance against various other regression approaches. Using the Matlab 2024a Regression Modeler application [25], the following models were trained validated with the same datasets of size 261, 117, 60, and 30:



- Linear Regression
- Cubic Support Vector Machine (SVM)
- · Trilayered Neural Network

These models were subsequently tested with independent Test Dataset 1 (materials and dimensions similar to the training/validation set) and Test Dataset 2 (the printed wiring board material dissimilar to the training/validation set). Figure 9 shows mean prediction error for each of these three models when compared with mean error for the baseline DNN of figure 3(a) and the best performing physics-enhanced DNNs (figures 3(b)–(d)).

4. Discussion

4.1. Natural frequency predictions using data similar to training

One major item immediately stands out from the data. The first observation is that for data of similar dimensions and materials as the training set, insertion of physics into the DNNs results in no statistically-relevant improvement for models generated with 261, 117, and 60 datapoints. In fact, for these simulations, the



t-Test shows that the only statistically-significant changes in predictive power are detrimental. This effect actually becomes more apparent for models generated using scarcer training data. For example, figure 5 demonstrates that, in general, inclusion of the weight term in the model (alone or in combination with modulus of rigidity and shear modulus) results in significantly worse predictive power than the simple non-enhanced DNN. Overall, therefore, no notable or consistent benefit is evident from the inclusion of physics-based parameters when testing with data similar to that used for training.

4.2. Natural frequency predictions using data dissimilar from training

In contrast to independent Test Dataset 1, although less accurate overall, Test Dataset 2 sees moderate but clear improvement to natural frequency predictions for certain combinations of physics parameters and insertion layers covering all four training dataset sizes. In particular, solutions where flexural rigidity is included as a physical parameter almost universally demonstrate better accuracy of natural frequency prediction over the baseline DNN without physics. For example, the greatest increase in accuracy occurs with inclusion of weight and flexural modulus in DNN Layers 1 through 4, exhibiting a total predictive error reduction of about 15% (from 33.2% to 19.6%) when the training dataset contains 117 points.

Interestingly, while not as effective when inserted into the DNN with larger training datasets, the addition of shear modulus to the *W*-*D* combination exhibits similar performance to just weight and flexural modulus alone for the training dataset of size 60, and the *W*-*D*-*G* combination actually outperforms all others when the model is



trained with only thirty 30 datapoints. Although not conclusive, this may indicate that certain combinations of physics-based parameters are more effective than others when applied to different training dataset sizes during model generation. In particular, those models with a greater number of terms creating parameter relationships appear to be effective in improving predictions.

4.3. Effect of embedding physics into different layers

As illustrated in figure 3, supplementation of physics-driven data was implemented with the neural network in several different layers, both singularly and simultaneously. Similar to results overall, figures 5–7 show no discernable improvement or detriment with any particular combination of layer augmentation for predictions made with using Test Dataset 1 (i.e., input parameters similar to the training dataset). However, when making predictions with Test Dataset 2, notable differences in accuracy are observed. In particular, when modulus of rigidity (D) and plate weight (W) are used as augmentative data parameters, the effect of injecting physics into Layer 1 or Layer 5 appears to be least impactful to accuracy. However, for these cases, the supplementation of a combination of inner layers or inner with Layer 5 provides a lower overall predictive error for the model. While further study is required to better understand this effect, it indicates that the supplementation of a DNN with physics-derived data may be most effective when conducted for multiple layers.



4.4. Loss and overfitting

One aspect of this study not yet discussed in detail is the comparison of training and validation loss. During the course of the investigation, it is observed that as the training dataset sizes decrease, the training and validation loss (based on mean-squared error, as indicated in table 4) increasingly separate from one another, as is evident in the example shown in figure 10. Although expected behavior for the DNN where all hyperparameters are held constant, it likely indicates an increasing degree of overfitting. As such, the number of epochs used for this study is too large and reduction to something less than 100 would improve computational efficiency while not likely deteriorating accuracy. Overall, further investigation to simplify the DNN while optimizing hyperparameters would benefit the approach to reduce the overfitting effect.

4.5. Comparison of approach with other regression techniques

As described in section 3.3, the baseline deep neural network and corresponding physics-enhanced models developed for this study were compared with various other regression approaches, specifically linear regression, cubic SVM, and a trilayered neural network. As figure 9 demonstrates, the baseline DNN significantly outperformed Linear Regression and Cubic SVM for all training/validation dataset sizes when independently tested against Test Dataset 1 and Test Dataset 2. In contrast, the baseline DNN showed similar performance to the Trilayered DNN approach. However, in all comparisons the best-case physics-enhanced models were found to provide the lowest average percentage prediction error.

5. Conclusions

This study serves as a continuation of an earlier investigation that introduced simplified physics-based data into a single internal deep neural network layer [17]. To evolve the approach, physics-enhanced parameters informing the deep neural network are not only injected into each DNN layer one-at-a-time, but reinserted into multiple layers during a single DNN computation. When tested by means of independent data, supplementation with simplified physics-based parameters provides virtually no reduction in prediction error over the baseline for models trained with dataset sizes of 60 and greater, although a small improvement of slightly better than 3% is observed when trained with a sparser size of 30 and physics introduced either to Layer 4 or 5 (but not multiple layers). However, notable gains in accuracy occur when the independent test data is for material and dimensions not conforming to the training set. Particularly, reductions in error from 33.2% to 19.6%, 34.9% to 19.9%, 35.8% to 22.4%, and 43.0% to 28.4% are achieved for training dataset sizes of 261, 117, 60, and 30, respectively.





For these cases, injection of physics into multiple layers per DNN consistently outperforms instances where only one layer is augmented, and this disparity is more apparent for models trained with 30 points. The initial lack of error reduction for similar training/independent test datasets coupled with subsequent greater improvements for dissimilar independent test data suggests that the approach described herein indeed provides improvement to DNN-based model generalizability.

To better understand the benefits of the simplified methodology discussed herein and further evolve the approach, ongoing research is focused on the following:

- *Hyperparameter Optimization*. Since hyperparameter tuning is critical for neural network success, assessment of the effect of optimization of such parameters—with particular attention to characteristics affected by the addition of simplified physics—is expected to further enhance the methodology.
- *Generalization Effect.* Elucidation of the observation that predictions are improved with the simplified physics approach when using dissimilar data to the training set but not with similar data will further clarify the benefit of the methodology.

Acknowledgments

This work was partially supported by the National Science Foundation Award 2014346.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

Conflict of interest

The authors declare no conflict of interest.

ORCID iDs

Nicholus R Clinkinbeard https://orcid.org/0000-0002-5807-2766 Nicole N Hashemi https://orcid.org/0000-0001-8921-7588

References

- Bianco M J, Gerstoft P, Traer J, Ozanich E, Roch M A, Gannot S and Deledalle C-A 2019 Machine learning in acoustics: theory and applications J. Acoust. Soc. Am. 146 3590–628
- [2] Hashemi N and Clark N N 2007 Artificial neural network as a predictive tool for emissions from heavy-duty diesel vehicles in Southern California Int. J. Engine Res. 8 321–36
- [3] Shirsavar M A, Taghavimehr M, Ouedraogo L J, Javaheripi M, Hashemi N N, Koushanfar F and Montazami R 2022 Machine learningassisted E-jet printing for manufacturing of organic flexible electronics *Biosensors and Bioelectronics* 212 114418
- [4] Shokrollahi O, Hashemi N and Dehghani M 2021 Discourse analysis of Covid-19 in persian twitter social networks using graph mining and natural language processing arXiv:2109.00298
- [5] Anuj K, Ankush K, Shyam B and Vipin K 2014 Predictive learning in the presence of heterogeneity and limited Proc. of the 2014 SIAM Int. Conf. on Data Mining (SDM) (https://doi.org/10.1137/1.9781611973440.29)
- [6] Nguyen L T, Zeng M, Tague P and Zhang J 2015 Recognizing new activities with limited training data Proc. of the 2015 ACM Int. Symp. on Wearable Computers (https://doi.org/doi.org/10.1145/2802083.2808388)
- [7] Xi Z and Zhao X 2018 An enhanced copula-based method for battery capacity prognosis considering insufficient training data sets IISE Annual Conf. and Expo 2018, X
- [8] Willard J, Jia X, Xu S, Steinbach M and Kumar V 2022 Integrating scientific knowledge with machine learning for engineering and environmental systems ACM Computing Surveys 55 1–37
- [9] Adadi A and Berrada M 2018 Peeking inside the black-box: a survey on explainable artificial intelligence (XAI) IEEE Access 6 52138–60
- [10] Chandrupatla T R and Belegundu A D 2002 Introduction to Finite Elements in Engineering (Prentice Hall) https://books.google.com/ books?id=okPUAQAACAAJ
- [11] Pletcher R H, Tannehill J C and Anderson D 1997 Computational Fluid Mechanics and Heat Transfer Second Edition (Taylor & Francis) https://books.google.com/books?id=ZJPbtHeilCgC
- [12] Blakseth S S, Rasheed A, Kvamsdal T and San O 2022 Deep neural network enabled corrective source term approach to hybrid analysis and modeling *Neural Netw.* 146 181–99
- [13] Liu G, Wang P and Kwon Y-O 2023 Physical insights from the multidecadal prediction of North Atlantic sea surface temperature variability using explainable neural networks *Geophys. Res. Lett.* 50 e2023GL106278
- [14] Zhu L, Zhang C, Zhang Z, Zhou X, Liu W and Zhu B 2020 A new and reliable dual model- and data-driven TOC prediction concept: a TOC logging evaluation method using multiple overlapping methods integrated with semi-supervised deep learning *Journal* of Petroleum Science and Engineering 188 106944
- [15] Zhu L, Zhang C, Zhang C, Zhang Z, Nie X, Zhou X, Liu W and Wang X 2019 Forming a new small sample deep learning model to predict total organic carbon content by combining unsupervised learning with semisupervised learning *Appl. Soft Comput.* 83 105596
- [16] Lu C and Zhang C 2023 Seismic velocity inversion via physical embedding recurrent neural networks (RNN) Applied Sciences 13 13312
- [17] Pawar S, San O, Aksoylu B, Rasheed A and Kvamsdal T 2021 Physics guided machine learning using simplified theories Phys. Fluids 33 011701
- [18] Daw A, Karpatne A, Watkins W, Read J and Kumar V 2022 Physics-guided neural networks (PGNN): an application in lake temperature modeling *Knowledge Guided Machine Learning* (Taylor & Francis) 15 353–372
- [19] Ahmed S E, Pawar S, San O, Rasheed A, Iliescu T and Noack B R 2021 On closures for reduced order models—a spectrum of firstprinciple to machine-learned avenues Phys. Fluids 33 091301
- [20] ANSYS 2021 ANSYS Mechanical (Version 2021 R1) [Finite Element Analysis]. https://ansys.com/
- [21] Steinberg D S 2001 Preventing Thermal Cycling and Vibration Failures in Electronic Equipment (Wiley) https://books.google.com/ books?id=Kh1TAAAAMAAJ
- [22] Kingma D P and Ba J 2015 Adam: a method for stochastic optimization 3rd International Conference for Learning Representations (San Diego, May 7-9) (https://doi.org/10.48550/arXiv.1412.6980)
- [23] Anaconda Software Distribution 2022 Anaconda Documentation (Anaconda Inc) Retrieved from https://docs.anaconda.com/
- [24] 2024 sklearn.preprocessing.MinMaxScaler Accessed 2 https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing. MinMaxScaler.html
- [25] Regression Learner 2024 Matlab In (Version 2024a). https://matlab.mathworks.com/