The University of Sheffield

Doctor of Philosophy Dissertation

Prior-guided Machine Learning for Monitoring Advanced Manufacturing Processes

Mohamed Atwya

2022

The University of Sheffield

Doctor of Philosophy Dissertation

Prior-guided Machine Learning for Monitoring Advanced Manufacturing Processes

---

Author: Mohamed Mohamed Gamaleldin Abouelmaty Atwya

A dissertation submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

The University of Sheffield
Department of Automatic Control and Systems Engineering
Faculty of Engineering

October 13, 2022

Executive Abstract

Abstract

Prior-guided Neural Networks are an emerging class of machine learning models that utilize priors in the form of loss function constraint. Prior-guided neural networks use prior loss functions (e.g., physics laws and empirical relations) in the learning stage to guide the optimizer toward more prior-consistent and generalizable solutions. An interesting challenge posed by prior-guided neural networks is weighing the value of priors relative to the empirical and regularization losses. Further, research to date focuses on parametric optimization rather than structure optimization. We present a new framework to optimize the structure of a neural network using prior knowledge. This is achieved through optimizing the number of hidden units via a line search and cross-validation using the empirical error to eliminate dataset/model-structure application dependency. We propose utilizing the prior errors as part of the cross-validation performance index to improve generalization. We present multi-prior-scalable correlation-based adaptive weighting algorithms to balance the interplay between prior and empirical errors in the loss function and the validation performance index. Results demonstrate that the proposed framework enhances the model's empirical accuracy and prior-consistency for benchmark convex and non-convex data sets. The presented results yield a new understanding of physics-guided neural networks in terms of their structural and parametric optimization and set the agenda for further studies model structure optimization and robust adaptive weighting methods. We further test the proposed methods on experimental data sets of laser powder bed fused high-strength aluminum A205 for in-situ porosity localization. Using exclusively within hatch stripe data, we localize porosity with a spherical equivalent diameter smaller than 50 µm with a classification accuracy of up to $80.67 \pm 2.67\%$. This is the first work in the literature demonstrating in-situ localization of porosities as small as 38.12 µm spherical equivalent diameter and is a more than a five-fold improvement on the work in the literature.

Research Contributions

The contributions of this work are split across three journal manuscripts, of which one is published and two are under review. The first and second manuscripts propose novel prior-guided neural network methods. The third manuscript applies prior-guided neural network methods to an additive manufacturing case study. In this Subsection, we cover the contributions of each manuscript.

Paper one is a published journal methodology paper and has the following contributions toward PGNNs:

1. Model structure optimization via the Hidden Unit Number: A framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency for prior-knowledge guided feedforward neural networks (model PGNN_1).

2. Model structure optimization via Prior-knowledge-guided cross-validation: In addition to using the prior-knowledge in the model training step, for the first time in the literature, we propose utilizing the prior errors as part of the cross-validation performance index. Incorporating the prior-consistency error along with the empirical error in the validation process improves generalizability (model PGNN_2).

3. For the first time in the literature, a rigorous statistical study on the empirical and prior-consistency error performance of prior-guided neural networks under the effects of random model weight initialization, data set size, hyperparameters, and the correlation between the empirical prior-consistency errors.

Paper two is an under-review journal methodology manuscript and has the following contributions:

1. For the first time in the literature, an analysis and comparison of the effects of the prior loss weights on the PGNN performance in the case of single versus multi-prior modeling problems.

2. Correlation-based prior loss adaptive weighting: A new prior-guided neural network computational framework that includes for the first time a scalable multi-prior correlation-based adaptive weighting algorithm to balance the interplay between the competing prior and empirical objectives in the loss function.

3. Weighted correlation-based model validation: For the first time in literature, we propose to adaptively weigh the prior errors in the validation performance index of prior-knowledge-guided neural networks. We propose a scalable multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step to balance the interplay between the different error terms in the cross-validation error function.

Paper three is an application-focused under-review journal manuscript and has the following contributions:

1. For the first time in the literature, in-situ detection and localization of porosities using a standardization-friendly multi-layer perceptron neural network, photodiode data, and machine-independent physics-driven features.

2. For the first time in the literature, a framework that given a three-dimensional sintering coordinate can make in-situ porosity localization predictions via (1) for the first time in literature, a local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe and within-layer sensory data and (2) a local 3-dimensional neighborhood of inter-layer sensory data.

3. For the first time in the literature, a framework based on prior-guided neural networks to utilize the often abundant SLM nominal data in the form of a prior training loss and a prior validation error.

4. To the author's knowledge, this is the first work in the literature capable of the localization of porosities as small as 38.118 µm spherical equivalent diameter ($0.029\text{x}10^{-3}$ mm^3), smaller than 47 µm spherical equivalent diameter in [1]. The proposed methods' localization of porosities as small as 38.118 µm spherical equivalent diameter is also a more than a five-fold improvement on the smallest spherical equivalent diameter porosity localization via photodiode sensory data.

The presentation of this research is covered in three publication-driven chapters. Chapter 1 introduces the literature, the literature gaps, and our contributions toward the methodologies and applications covered in this work. In Chapter 2 we present the first publication (methodology-focused) on the Structure optimization of prior-guided neural networks. Chapter 3 presents the second manuscript (methodology-focused) on multi-prior-scalable weighting for

competing prior objectives in prior-guided neural networks. Chapter 4 presents the third manuscript (application-focused) on prior-guided neural networks for in situ porosity localization in laser additive manufacturing. In the Conclusion Chapter 5, we discuss the proposed research work's findings, strengths, and limitations.

Acknowledgements

I would like to thank the PhD supervisor, Prof. George Panoutsos, for his continuous support, guidance, and advice throughout the PhD. I express my gratitude to all of the academic staff of the Department of Automatic Control and Systems Engineering for their support throughout my time at the University.

I am grateful to my grandmother, parents, and brother for their continuous encouragement, support, and love. To my parents, thank you for giving me the opportunity and inspiration to be who I am today. To my grandmother for her limitless love and care. Finally, to my brother for always being there for me.

This work was supported by the UK EPSRC Future Manufacturing Hub - Manufacture using Advanced Powder Processes (MAPP) through grant Grant EP/P006566/1 and the University of Sheffield's EPSRC Impact Acceleration Award, under project 'Machine Learning digital twin for defect-free additive manufacturing', 2022.

Thank you,
Mohamed Atwya

Declaration of Authorship

I, Mohamed Atwya, declare that the work presented in this thesis is my own. All material in this thesis which is not my work has been appropriately accredited and referenced.

Mohamed Atwya

February 18, 2023

Contents

Executive Abstract **iii**

Acknowledgements **vii**

Declaration of Authorship **ix**

Contents **xii**

1 Introduction **1**

1.1 Methodology Introduction 1

1.1.1 Methodology Literature Gaps 5

1.2 Application Introduction 6

1.2.1 Application Literature Gaps 8

1.3 Aims and Objectives 11

1.4 Contributions 12

2 Paper One (methods): Prior-guided Neural Network Model Structure Optimization **14**

2.1 Paper One Summary 14

2.2 Paper One 18

3 Paper Two (methods): Prior-knowledge-guided Neural Networks: Analysis of competing prior objectives and new correlation-based weighting **53**

3.1 Paper Two Summary 53

3.2 Paper Two 57

4 Paper Three (application): In Situ porosity localization in laser additive manufacturing using photodiode sensor measurements: a prior-guided neural network approach **71**

4.1 Paper Three Summary 71

4.2 Paper Three 76

5 Conclusion **116**

5.1 Future work 122

References **1**

Chapter 1

Introduction

1.1 Methodology Introduction

Domain knowledge is available as either physics equations that define physical relationships between variables or physics-based numerical simulations for large scale physical systems. Where a system can be described by an exactly known relation, it would be best to utilize a physical law to model the system. However, in complex systems, several physical relations are necessary to create a model.

Large-scale physics models typically contain unknown or uncertain parameters (e.g. heat transfer coefficients, material constants, stiffness, flow rates) during the model development phase or for new input conditions. The unknown parameters are traditionally estimated via experimentally observed data and deterministic inverse techniques such as the least-squares. However, in practice, sparse and scarce data often co-occurs with large-scale dynamical systems.For example, friction stir welding and additive manufacturing defect detection applications suffer from sparse and scarce data due to experimental expense and imbalanced data sets [2, 3].

Physics simulations such as finite element analysis (FEA) and computational fluid dynamics (CFD) can be computationally-expensive, especially for high-fidelity models. Accordingly, the parameter estimation inverse problem can become intractable when the parameter search space is large. A further limitation is that physics-based models typically utilize approximate forms of physical relationships, which introduces bias. Where physics equations or physics-based numerical simulations are not possible, machine learning methods are often utilized.

Machine learning (ML) is a branch of computational algorithms that are designed to learn through experience. ML is primarily applied for predictions and optimization of practical objectives and applications with large and multi-dimensional data sets. Machine learning methods can be broadly classified into supervised, and unsupervised learning.

Supervised ML involves the learning of a function that maps an input to an output based on example input-output pairs (i.e., labeled data). Supervised ML methods include but are not limited to decision trees, random forest, artificial neural networks, logistic regression, and support vector machines.

A black-box supervised ML model is only as good as the representative quality of the labeled data. However, data is often sparse and scarce in scientific applications due to operational constraints within optimal window settings, high costs and time demand limits. Operating systems outside their optimal window settings in industries such as manufacturing facilities, power plants, and nuclear reactors can also be unsafe. Empirical supervised ML models are also prone to learning false statistical relationships that look deceptively good on training and test sets but are inconsistent with physics laws. Therefore, supervised ML models are, at best, valid only for the range of variable combinations in the training data but are incapable of generalizing to scenarios unseen in the training data (i.e. extrapolation) and are not suitable for subsequent scientific developments.

Theory-guided data-driven modeling (TGDM) is a new paradigm from 2017 for integrating physics-based models and data-driven models to improve the model's empirical performance and consistency with theory [4]. TGDM includes five categories: (1) choosing a suitable model design, (2) refining data model outputs, (3) augmenting theory-based models, (4) hybrid models, and (5) navigating the search space of candidate models.

For clarification, we note that in the literature of theory-guided data-driven modeling, several terminologies are used to refer to the application-dependent domain knowledge utilized (theory, prior, physics, etc.). In this work we choose the term prior-guided neural networks to refer to TGDM.

Choosing a suitable model design is essential for data-driven models to capture the distribution and relationships between the input and target data. The model design process should consider the link function and model architecture choices. For example, if the target is skewed, then a Gumbel distributions would be more suitable than the standard Gaussian distribution. The model architecture involves specifying the connections between nodes (e.g. recurrent NNs for time-series data and convolutional NNs for image data). A possible research direction is to investigate how domain-knowledge-based variable-dependencies can be mapped onto a NN architecture (connections, nodes, and layers).

Augmenting theory-based models includes data assimilation methods and calibrating theory-based models using observational data. Refining data model

outputs looks at constraining and refining data-driven model outputs using theory-based constraints and theory-based models, respectively. Within the field of material science, Geoffroy et al. [5] discovered 209 new ternary oxide compounds with desirable properties that were previously unknown using traditional approaches via theory-guided refinement of data-driven model outputs. The conventional approach for predicting crystal structures and properties is to use first-principle calculations (e.g. density functional theory). However, it is impractical to estimate the structure and properties of all the possible materials due to the computationally-expensive theory-based calculations. Geoffroy et al. used a machine learning approach to estimate the most likely crystal structure candidates and the used a theory-based energy model (DFT) to determine which crystal structures are stable [5].

Hybrid models involve either using theory-based model outputs as an input to the data-driven models or using data-driven models to estimate missing/inaccurate variables in theory-based models. For example, in [6] the authors use the output of two theory-based models (binary result of inland and coastal flooding occurrence) as an input to a cycle-consistent adversarial network which visually transform images of houses under normal weather conditions to flooded houses.

In [7] the authors developed a multi-step method to estimate theory-based equations from synthetic data sets. The first step is to mathematically transform the original features to a library of base functions in an iterative process (e.g. cross/power, hyperbolic, and trigonometric). The second step involves creating new basis functions that combine the top 10-20 original basis functions with the highest correlation to the output in the synthetic data set. The third step is pick 2 to 6 of the highest correlated base functions using principal component analysis (PCA) in an iterative process, and create random combinations of the functions. They then use the random function combinations as input features to train multiple NNs and estimate the output from the synthetic data. The models with the lowest prediction error (40 in one of the examples), are then used to identify the most basis functions present in the unknown theory-based equation. They demonstrate this method on three equations: (1) motion under constant acceleration, (2) heat transfer coefficient for forced convection, and (3) the angle of deflection of supersonic air of a given velocity affected by a shock-wave at a given shock angle. Their proposed iterative correlation method is closely related to evolutionary algorithms and provides an interpretable framework that facilities scientific discovery. However, the methodology is not systematic and has several ambiguous hyper-parameters.

The fifth category of TGDM is navigating the search space of candidate models via prior knowledge using four different approaches: (1) model parameter initialization, (2) probabilistic models, (3) regularization, and (4) constrained optimization.

Model parameters are typically assigned random values during initialization. Random weight initialization in machine learning methods can lead to getting stuck at local minima, saddle points, and flat regions in the loss function.

Prior-guided parameter initialization methods can help guide machine learning (ML) algorithms towards generalizable and prior-consistent solutions [8, 9]. Prior-guided parameter initialization methods include the matrix completion algorithm [10] and the use of computational simulation data for pre-training [9, 11].

The matrix completion approach tackles the challenge of incomplete sparse data. Filling the missing entries in sparse data can significantly improve the quality of the data set and the empirical performance of data-driven modeling approaches [10]. Many matrix completion algorithms are iterative and use fixed or random values to initialize the matrix. Where prior knowledge is available, the initialization of the missing values can be prior-guided, for example, by using the mean of every attribute as initial values in the matrix completion process [8].

TGDM regularization methods include the use and development of new Lasso variants such as the sparse group Lasso to impose theory-specific structures on the parameters of DS models. Chatterjee et al. [12] utilized sparse group Lasso to group climate variables based on their spatial location. Grouping the climate variables observed at specific spatial locations ensured that only the relevant features are considered. By studying the spatially coherent features, physics-consistent relationships in climate science can be identified for individual regions. In the filed of genetics, domain-knowledge (linkage disequilibrium) suggest that genetic markers are typically located in close proximity on the genome. Accordingly, [13] Liu et al. introduced a smoothed min-max concave penalty to Lasso to capture the linkage disequilibrium between adjacent markers. Many physics laws, especially in additive manufacturing (AM), are described as ordinary differential equations and PDEs to capture relationships between physical quantities. A possible research direction would be to develop a Lasso variant that impose the initial and boundary conditions of PDEs.

Constrained optimization methods include developing constrained optimization methods for PDEs/non-linear transformations and integrating

linear equalities and inequalities into existing optimization techniques.

In prior-guided neural networks (PGNN), linear equalities and inequalities are added into existing optimization techniques in the for of prior loss functions. The aim of prior loss functions is to constraint the optimization search algorithm according to the prior knowledge. In the following subsection we discuss the literature gaps in prior-guided neural networks.

1.1.1 Methodology Literature Gaps

The number of hidden units dictates the NN/PGNN model's complexity and is commonly between 5 and 100 [14]. The number of units should be sufficiently large to capture the data non-linearity but not too large to avoid over-fitting and maintain the model's smoothness between the in-sample training data points. Typically, the number of hidden units is chosen via application-specific expert knowledge or empirical experimentation. The work in [9, 15–18] among several other works either sets the number of hidden units based on expert knowledge or reports utilizing sensitivity studies without providing a systematic framework or a statistical analysis on the empirical and prior-consistency performance. The authors in [16] note that the interplay between the physics-guided NN architecture/training procedure and the complexity of the modeling data is poorly understood and propose a Bayesian approach to monitor the variance of the predictive posterior distribution as possible future work.

The authors in [19] note that their method requires systematic searching and sampling procedures to find the optimal physics-guided NN architecture and set it as future work. The work in [16] states that designing the correct physics-guided NN architecture is an open research question and confirms that they have observed that a specific physics-guided NN architecture that yields accurate results for one data set can fail for another data set. Therefore, there is a need for a systematic framework along with statistical analysis.

Another gap in the PGNN literature is a lack of statistical analysis of PGNNs, including:

- A lack of statistical analysis on the mean and variance of the empirical and prior-consistency errors across different random weight initialization of PGNN models.
- A lack of investigation with statistical tests on the effect of the data set size on the mean of the empirical and prior-consistency errors.

- A lack of investigation with statistical tests on the correlation between the empirical and prior-consistency errors.
- A lack of investigation with statistical tests on the effect of the loss weight hyperparameters on the mean of the empirical and prior-consistency errors.

Although several studies have examined and verified the empirical performance effects of prior-consistency loss functions on PGNNs, researchers have not treated the competitive relationship between prior-consistency loss functions in much detail. Research on the subject has been mostly restricted to limited comparisons of vanilla NNs and PGNN variants (e.g., with/without labeled data, different model architectures, and with/without the empirical loss) with a fixed number of priors. Previous studies of PGNNs have not examined the impact of weighting competitive prior loss functions using single priors versus using 3+ combinations of priors in PGNNs. Similar to the prior loss analysis gap, there is no analysis on using single priors in the literature versus using 3+ combinations of priors in the validation performance index (PI) of PGNNs. There is also no attempt in the literature to adaptively or empirically weigh the prior errors in the validation performance index of PGNNs. Another gap in the PGNN literature is the lack of generalizability of the proposed methods. Most PGNN techniques presented in the literature are tailored to specific applications, use several sensitivity studies to set hyperparameters, and suffer from scalability limits. The prior-consistency adaptive weighting gaps in PGNNs include an analysis of using single versus multiple priors, methods to weigh prior errors in the validation performance index, application independence, and scalability.

To summarize, PGNNs often outperform vanilla NNs in terms of empirical and prior-consistency error metrics when application-specific expert knowledge and empirical experimentation are applied to set the model structure and hyperparameters. However, due to the reliance on application-specific expert knowledge and empirical experimentation, PGNN methods in the literature are often not generalizable. Therefore, there is a need for a generalizable PGNN framework and statistical analysis to understand and validate the performance attained by PGNNs.

1.2 Application Introduction

Manufacturers must deliver new products to the market across shorter time intervals due to several reasons including globalization, the rapid introduction

of new technologies, and shorter product life cycles [20–22]. The increasing demand for rapid, flexible, and cost-effective manufacturing results in a driving force towards novel rapid manufacturing technologies such as AM [23, 24]. Academic and industrial investments in advancing the AM technology continue to rise with the aim of revolutionizing manufacturing.

At present, AM technologies generally encounter the same challenges regardless of the specific process employed. The main challenges include relatively high unit costs due to expensive raw materials and machinery, part-part and machine-machine repeatability, and reject rates for finished parts [24, 25]. Improving the cost-related AM limitations and ensuring repeatable sufficient AM part quality (mechanical properties and geometry), require improving both the underlying AM technology and the quality control framework of the process [26]. In the literature, process modeling, monitoring, and control strategies have been deployed to overcome some of these drawbacks.

An important feature of the AM process is the resultant mechanical properties of the manufactured parts. In-situ monitoring and controlling the microstructure and defect formations in AM is necessary as it dictates the mechanical properties of the sintered parts. The microstructure and defect formation in AM processes is very sensitive to slight deviations in process parameters [27]. Therefore, AM processes have tight operating windows for different materials. Sintering outside the experimentally-validated operating windows is either unexplored or associated with defects such as lack of fusion, porosity, and key-holing [28].

A single AM metal part can have several microstructural features and porosity types that dictate its mechanical properties. The microstructure and defects are formed in-situ and primarily depend on the process parameters and material used [29]. For example, a higher laser scanning speed can lead to an increase of the HAZ, which can lead to lack of fusion porosities and the formation of larger grain sizes and more α generations [30, 31]. It is possible to perform to use design of experiment (DOE) methods to estimate the process-property-performance (PPP) linkage and find the optimal parameter settings that minimize defects and achieve the desired microstructre and mechanical properties. However, AM is a multiple-input-multiple-output multi-physics process which makes it costly and time-consuming to rely on DOE alone. Additionally, the highly non-linear dynamics of AM results in a large uncertainty of the DOE surface response mapping in areas that are untested (i.e. extrapolation) [32].

To overcome the challenges associated with DOE, numerical and data-driven

models can be utilized to provide insights and allow real-time process monitoring and control of the process parameters. Real-time process monitoring and control are desirable to achieve the desired microstructure, higher manufacturing repeatability, reduce defects, and potentially achieve functionally-graded AM capabilities.

In the literature, data-driven models have been applied to the following parts of AM: design, in situ monitoring, and the process–property–performance linkage. Machine learning NNs have been applied to estimate one of following properties at a time: dimensional accuracy, surface roughens, shrinkage, density, compressive strength, and porosity [33–36].

Many efforts in the literature have been made towards in-situ monitoring the defect formation in additive manufacturing towards controlling the mechanical properties of the sintered parts. However, the current work in the literature is limited by the need for inter-layer three-dimensional neighborhoods of sensory data, the need for several sensors, and poor experimental and data-driven modeling validation. In the following Subsection 1.2.1 we discuss the literature gaps in data-driven in-situ defect monitoring and localization.

1.2.1 Application Literature Gaps

1. There is a need for terminology standardization regarding defect detection, localization, and quantification. The following terms are often used with vague clarity: defect detection, in-situ/online defect detection, defect prediction, correlation maps, and layer-wise defect detection. Specifically, the word detection is often used as an umbrella term to cover a variety of different research outcomes making it difficult to appraise the literature.

2. There is a need for a defect detection/localization spatio-temporal granularity scale. In the literature, various frameworks for sensory data collection and defect 'detection' exist and vary from one to three-dimensional space. Some work involves collecting and making layer-wise predictions, whereby the prediction of defect status is performed after the layer is sintered [37]. Other work collects data from multiple layers before making a defect prediction, so for a given coordinate, data from the preceding/succeeding layers are considered[1, 38]. The type of sensor utilized also affects the granularity and frequency of defect detection within a layer and across multiple layers. After sintering a particular coordinate, there needs to be clarity on (1) how quickly can a priority status be predicted, (2) is there a need for further data from

following hath paths or succeeding/preceding layers, and (3) with regards to porosity, what is the smallest detectable porosity volume?

3. There is a reoccurring demand for more compatible and standardized frameworks that are transferable between sintering systems/materials and can flexibly accommodate the lack of or addition of sensors.

4. The existing literature on defect detection in AM mostly focuses on layer-wise detection and correlations between process parameters and defect descriptive statistics (e.g., total defect volume and mean defect volume/radius).

5. The experimental validation of ML defect detection in AM is often performed via single-track prints and thin walls which have simple geometries and relatively simple heat transfer dynamics.

6. Defects are often simulated via mixing different powders. However, the defect formation heat dynamics that occur by mixing powders are not necessarily the same defect formation heat dynamics that occur when a single powder is used.

7. There is a broad miss-conception in the use of ML for AM where it is viewed often viewed that a data-driven ML algorithm is a decision maker. The standard data-driven ML algorithms often used in AM are only inference models that predict a regression value such as the melt-pool temperature or a classification value such as the probability of a porosity. Regression and classification predictions are separate from decisions.

8. The existing literature on ML defect detection does not facilitate capturing the effects of the laser spot overlap and the melt pool remelting zone in-situ. The capability to continue monitoring and update the porosity status per sintered coordinate from surrounding hatch paths and preceding/succeeding layers is important for real-time feedback control and correction systems.

9. The existing literature on ML defect detection that does capture the effects of the laser spot overlap and the melt pool remelting zone often need the collection of data from multiple layers before making a prediction.

10. The ML model validation in AM defect detection is often limited due to limited experimental data, poor training/testing/validation data splits,

and the lack of use of methods such as repeated cross validation, multiple random model weight initialization, and prior-guided cross-validation.

1.3 Aims and Objectives

This work aims to develop prior-consistent and generalisable neural network models of dynamical systems via theory-guided machine learning and data-driven modeling. The application of this work investigates the use of the proposed theory-guided ML methods for in-situ detection and localization of porosities in selective laser melting additive manufacturing. These aims raise the following project objectives:

1. Develop a new understating of the impact of hyperparameters and the prior/empirical performance correlation via rigorous statistical analysis and benchmark examination.

2. Develop a novel model structure optimization framework for prior-guided neural networks towards improving the model's prior-consistency and the empirical performance.

3. Develop a novel generalizable multi-prior-scalable weighting framework to balance the interplay between the empirical and prior errors in the loss and validation functions.

4. Develop a novel prior-guided neural networks framework for in situ porosity localization in laser additive manufacturing.

1.4 Contributions

The contributions of this work are split across three journal manuscripts, of which one is published [39] and two are under-review. The first and second manuscripts propose novel prior-guided neural network methods. The third manuscript is an application of prior-guided neural network methods to an additive manufacturing case-study. In this Subsection we cover the contributions of each manuscript.

Paper one is a published journal methodology paper and has the following contributions towards PGNNs [39]:

1. Model structure optimization via the Hidden Unit Number: A framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency for prior-knowledge guided feedforward neural-networks (model PGNN_1).

2. Model structure optimization via Prior-knowledge-guided cross-validation: In addition to using the prior-knowledge in the model training step, we propose utilizing the prior-errors as part of the cross-validation performance index. Incorporating the prior-consistency error along with the empirical error in the validation process improves generalizability (model PGNN_2).

3. A rigorous statistical study on the empirical and prior-consistency error performance of PGNNs under the effects of random model weight initialization, data set size, hyperparameters, and the correlation between the empirical prior-consistency errors.

Paper two is an under-review journal methodology manuscript and has the following contributions:

1. An analysis and comparison of the effects of the prior loss weights on the PGNN performance in the case of single versus multi-prior modeling problems.

2. Correlation-based prior loss adaptive weighting: A new prior-guided neural network computational framework that includes for the first time a scalable multi-prior correlation-based adaptive weighting algorithm to balance the interplay between the competing prior and empirical objectives in the loss function.

3. Weighted correlation-based model validation: We propose a scalable multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step to balance the interplay between the different error terms in the cross-validation error function.This is the first study to adaptively weigh the prior errors in the validation performance index of prior-knowledge-guided neural networks.

Paper three is an application-focused under-review journal manuscript and has the following contributions:

1. A standardization-friendly approach to in-situ detection and localization of porosities using vanilla NNs, photodiode data, and machine-independent physics-driven features.

2. A framework that given a three-dimensional sintering coordinate can make in-situ porosity localization predictions via (1) for the first time in literature, a local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe and within-layer sensory data and (2) a local 3-dimensional neighborhood of inter-layer sensory data.

3. For the first time in the literature, a framework based on prior-guided neural networks to utilize the often abundant SLM nominal data in the form of a prior training loss and a prior validation error.

4. To the author's knowledge, this is the first work in the literature capable of the localization of porosities as small as 38.118 µm spherical equivalent diameter (SED), smaller than 47 µm SED in [1]. The proposed methods localization of porosities as small as 38.118 µm SED is also a more than a five-fold improvement on the smallest SED porosity localization via photodiode sensory data [38].

The presentation of this research is covered in three publication-driven chapters. In Chapter 2 we present the first publication (methodology-focused) on the Structure optimization of prior-guided neural networks. Chapter 3 presents the second manuscript (methodology-focused) on multi-prior-scalable weighting for competing prior objectives in prior-guided neural networks. Chapter 4 presents the third manuscript (application-focused) on prior-guided neural networks for in situ porosity localization in laser additive manufacturing. In the Conclusion Chapter 5, we discuss the proposed research work's findings, strengths, and limitations.

Chapter 2

Paper One (methods): Prior-guided Neural Network Model Structure Optimization

2.1 Paper One Summary

Paper one focuses on the structure optimization of a multi-layer perceptron neural network using prior knowledge [39]. This work is the first rigorous investigation of the merit of prior-guided neural networks in terms of empirical and prior-consistency performance. We test and validate the empirical and prior-consistency error performance of PGNNs across 14 regression benchmark functions and examine the hyperparameter effects' via various statistical studies. The 14 regression benchmark problems include convex data sets with a unique minimum and non-convex multi-modal data sets. The proposed methods are assessed through empirical root-mean-square-error (RMSE) accuracy and prior-knowledge RMSE consistency. Below is a summary of the paper's contributions.

- **Model structure optimization via the Hidden Unit Number:** A framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency for prior-knowledge guided feedforward neural networks (model PGNN_1).
  - Optimizing the network architecture (PGNN_1) with repeated cross-validation statistically significantly improves the empirical and prior-consistency RMSE means relative to the reference models by selecting the number of hidden units with the smallest empirical error. However, a limitation of PGNN_1 is the possibility of a larger hidden unit quantile range across random model weight initialization.

The $PGNN_1$ limitation is hypothesized to be due to the data set size versus the function's complexity and the un-tuned hyperparameters.

- **Model structure optimization via Prior-knowledge-guided cross-validation:** In addition to using the prior-knowledge in the model training step, for the first time in the literature, we propose utilizing the prior errors as part of the cross-validation performance index. Incorporating the prior-consistency error along with the empirical error in the validation process improves generalizability (model $PGNN_2$).

  - Using the prior knowledge and the empirical error in the validation process, $PGNN_2$, statistically improves the empirical and prior-consistency RMSE mean and variance relative to the reference models. By incorporating the prior-consistency error in the validation performance index, $PGNN_2$ re-enforces prior-consistency. $PGNN_2$ outperforms (empirical/prior error mean and variance) $PGNN_2$ relative to the reference models by selecting a suitable number of hidden units to avoid under-fitting or over-fitting depending on the training/prior data landscape versus the data set size. Further, the positive correlation between the empirical RMSE and the prior-consistency RMSEs supports utilizing the prior-consistency error in the validation performance index of PGNNs ($PGNN_2$).

- **A rigorous statistical study on the empirical and prior-consistency error performance of PGNNs under the effects of random model weight initialization, data set size, hyperparameters, and the correlation between the empirical prior-consistency errors.**

  - The data set size investigation has shown that PGNNs result in a 1-dimensional or 2-dimensional statistically significant difference ($P<0.05$) in the multivariate empirical and prior-consistency means for convex and non-convex modeling problems up to 2000 sample data set size. The univariate analysis has shown a statistically significant difference ($P<0.05$) in the empirical and prior-consistency errors for up to sample data set size 4000 and 2000, respectively. The functions' complexity, the prior value, and the correlation between the empirical and the prior-consistency errors dictate the magnitude of the multivariate difference between data set sizes. In the

hyperparameter sensitivity investigation, we show that the proposed methods are not affected by the loss weight hyperparameters and consistently improve the empirical and prior-consistency RMSE means relative to the reference models.

The significance of this work in the field of prior-guided neural networks includes the following findings:

- Intuitively and as appraised in the literature, adding a prior to a neural network learning framework in the form of a prior loss should improve the NN's inference capabilities. However, for the first time in the literature, we show a lack of a statistically significant difference in the empirical and prior-consistency error means and variances between a vanilla NN and a PGNN with 100 hidden units (NN and PGNN-100) for 14 benchmark functions. The empirical evidence from this study shows that given an identical network architecture, a PGNN model is unlikely to statistically significantly outperform (empirical and prior-consistency error-wise) a vanilla NN.

- Prior to this study, it was difficult to make predictions about how the model structure impacts the PGNN empirical/prior-consistency performance. By optimizing the network architecture, out of the 14 benchmark functions and across 24 random model weight initializations, our proposed PGNN_1 statistically significantly outperformed the reference NN and PGNN-100 models for up to 6 functions in terms of the empirical and prior-consistency error mean and variance.

- Using priors in the model validation process advances the field of prior-guided neural networks and is a step towards more generalizable data-driven ML models. By incorporating the prior-consistency error in the validation performance index, out of the 14 benchmark functions and across 24 random model weight initializations, our proposed PGNN_2 statistically significantly outperformed the reference NN, and PGNN-100 models for up to 8 functions in terms of the empirical and prior-consistency error mean and variance.

  - The discrepancy in the error improvements between PGNN_1 and PGNN_2 is due to PGNN_2 validating the prior-consistency of the model in addition to its empirical generalizability. We note that PGNN_2 has a larger hidden unit quantile range $[0.25, 0.75]$ than

$PGNN_1$ for 12 out of the 14 benchmark functions. $PGNN_2$ outperforms (empirical/prior error mean and variance) $PGNN_1$ relative to the reference models by selecting a suitable number of hidden units to avoid under-fitting or over-fitting depending on the training/prior data landscape versus the data set size.

- Through a hyperparameter sensitivity investigation using the two-sample Hotelling's T^2 test, we show that (1) when using the proposed methods to optimize the model structure, the empirical and prior-consistency errors are not affected by the loss weight hyperparameters and (2) for a given set of hyperparameters, the proposed methods consistently improve the empirical and prior-consistency RMSE means relative to the reference models.

- The performance of $PGNN_1$ and $PGNN_2$ and the finding of the hyperparameter investigation shed new light on the importance of model structure optimization as opposed to the re-occurring literature focusing on loss function weighting sensitivity studies.

- The experimental work presented here provides one of the first statistically significant ($P< 0.05$) evidence of a positive correlation between the empirical RMSE and the prior-consistency RMSE for the three prior scenarios of two benchmark function case studies. The positive correlation between the empirical RMSE and the prior-consistency RMSEs demonstrates the importance of developing prior-consistent NNs. The positive correlation between the empirical RMSE and the prior-consistency RMSEs further supports using the prior-consistency error in the validation performance index of PGNNs ($PGNN_2$).

A limitation of this work is the assumption that the prior-consistency error and the empirical error are linearly-related (equally weighted) in the cross-validation performance index in terms of training a neural network. However, (1) the prior-consistency and empirical errors may have a non-linear correlation or a statistically insignificant correlation, and (2) the importance of prior-consistency relative to the empirical error performance is application dependent. A natural progression of this work is to investigate and develop adaptive weighting metrics for the empirical and prior-consistency errors in the cross-validation performance index. Further research should focus on defining the relationship between the prior type/value and the modeling problem landscape complexity to develop a complete picture of the importance of priors in the context of PGNNs.

2.2 Paper One

Structure optimization of prior-knowledge-guided neural networks

Mohamed Atwya[1,*], George Panoutsos[1]

Abstract

Prior-knowledge use in neural networks, for example, knowledge of a physical system, allows network training to be tailored to specific problems. Literature shows that prior-knowledge in neural network training enhances predictive performance. Research to date focuses on parametric optimization rather than structure optimization. We present a new framework to optimize the structure of a neural network using prior-knowledge. This is achieved through optimizing the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency for prior-knowledge guided neural-networks. In addition to using the prior-knowledge in the model training step, we propose utilizing the prior-errors as part of the cross-validation performance index to improve generalization. Results demonstrate that the proposed training framework enhances the model's prediction accuracy and prior-knowledge consistency for convex data sets with a unique minimum and non-convex multi-modal data sets. The presented results yield new understanding on physics-guided neural networks in terms of their structural and parametric optimization.

Keywords: Prior-guided neural network, Machine learning, Constrained optimization, Structure optimization

1. Introduction

When modeling physical systems, prior knowledge is typically not exploited in machine learning frameworks. Prior information, such as first principle physical laws and empirically supported relationships, could guide a learning algorithm towards a more theory-consistent solution. Model consistency to any underlying physical laws can improve the model's generalization ability, especially when only a few training examples are available, which is a common challenge when dealing with problems where data generation is very resource-intensive.

Using prior-information in model training is not a new challenge in the literature. In 2017, Karpatne et al. [1] classified theory-informed data-driven models into the following five approach categories: (1) choosing a suitable model design, (2) refining data model outputs, (3) augmenting theory-based models [2], (4) hybrid models [3, 4], and (5) Navigating the search space of candidate models. Navigating the search space of candidate models can be influenced by prior knowledge using four different approaches: (1) model parameter initialization, (2) probabilistic models, (3) theory-based regularization, and (4) constrained optimization [1].

Theory-guided parameter initialization methods such as the matrix completion algorithm and pre-training using computational simulation data can help guide ML algorithms towards generalizable and theory-consistent models [5, 6]. Probabilistic models in the context of TGDM involve including theory-guided data distributions as Bayesian priors [7]. Theory-guided probabilistic models have shown success in applications such as predicting subsurface aquifers connectivity and electrical activity within the heart wall [8, 9]. TGDM regularization methods include the use and development of new Lasso variants such as the sparse group Lasso to impose theory-specific structures on the parameters of DS models [10, 11].

Constrained optimization methods include (1) developing constrained optimization methods for PDEs/non-linear transformations and (2) integrating theory-based constraints with existing optimization techniques. In this work for the

*Corresponding author.
Email address: Matwya1@sheffield.ac.uk (Mohamed Atwya)
[1]Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

first time in the literature, we focus on integrating theory-based constraints within optimization methods to navigate the search space of candidate feedforward neural network models.

Note that in the literature of theory-guided machine learning, several terminologies are used to refer to the application-dependent domain knowledge utilized (theory, prior, physics, etc.). In theory-based constrained optimization, the methods developed can be applied to any domain-knowledge that can be expressed in the form of a constraint. In this work we choose the term prior-knowledge-guided neural networks.

In [12] they construct physics-guided neural networks with a hybrid-physics-data model that uses a physics-based model's simulated output as an additional feature when training the neural network (NN). The authors introduce a method to add two forms of physics relationships, equality and inequality constraints, to the loss function. They use the mean square error (MSE) for the empirical loss, a rectified linear unit (ReLU) function for the physics-consistency loss, and elastic net regularization for model complexity control. The model structure was set to 3 hidden layers with 12 hidden units each. They demonstrate the prediction error and physics-consistency error improvements of the physics-guided NN on a spatio-temporal problem using labeled and unlabeled data.

Similar to [12], the authors in [13] also add equality and inequity physics-based constraints to the NN loss function, and use $\mathcal{L}_2$ regularization. The network architecture consists of two hidden layers, with the first layer having 64 hidden units and the second layer having $128 units$. They set the $\mathcal{L}_2$ regularization weight value $\lambda_R = 1$ and empirically optimize the physics-based constraint loss weight λ_p. The authors use a $60-20-20$ split to train, validate, and test the models. The results show empirical error percentage improvements on noise-free and noisy synthetic (Bohachevsky) and real data sets.

In [6], Jia et al. use recurrent NNs (RNN) and advance the work of [12] to estimate the spatio-temporal lake temperature. Their first contribution is the use of theory-based simulation data to pre-train the RNN. They also propose running a theory-based energy balance model in parallel with the RNN. If the temporal regression of the temperature violates the energy balance constraint, a penalty is added to the loss function. They perform sensitivity studies on the impact of the physics-based constraint weights by varying one weight and keeping the second weight constant.

Wu et al. [14] demonstrated how incorporating a novel statistical covariance-based constraint in the optimization loss function of deep convolutional generative adversarial networks (DCGANs) can improve training stability and convergence properties. They demonstrated the performance improvement of their GANs-based physical system emulator on the Rayleigh-Benard convection.

Liu and Wang [15] developed a multi-fidelity physics-constrained NN (MF-PCNN) and propose an adaptive weighting scheme for regularization. They apply their method to PDEs from two-dimensional heat transfer, phase transition, and dendritic growth problems. The authors add a physics-equation-based constraint and initial condition (IC) and boundary condition (BC) constraints to the cost function. They successfully train several MF-PCNN variants with four layers but with different hidden unit numbers identified by conducting sensitivity studies. The authors acknowledge that their method requires searching and sampling procedures to find the optimal architecture and set it as future work. Even though their proposed method does no use model complexity control (e.g., $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization), they claim that the sensitivity studies avoid over-fitting. However, in the results, an increase in the MSE of one of the MF-PCNN variants is attributed to over-fitting. Their proposed adaptive weighting scheme results in higher prediction accuracy and shorter training time than standard NNs.

In [16] the authors compute data-driven solutions to PDEs using physics-guided neural networks (PGNN). Their method involves training two NNs in two steps. Step one is to train a 9-layer NN with 20 units per layer to model a data set extracted from the initial and boundary conditions. Step two is to compute the partial derivatives of the first NN, substitute the partial derivatives into the PDE to define the second 9-layer NN with 20 units per layer (i.e., the PGNN), and train it on collocation points. They demonstrate this method on Burger's equation with Dirichlet boundary conditions in one space dimension.

The more recent work in [17] proposes a new PGNN using the system's variational energy as the loss function combined with the transfer learning method. The authors argue that the proposed loss function is easier to minimize, and hence, the proposed PGNN performs better relative to PGNNs with residual loss functions such as in [16, 18, 19]. They demonstrate the effectiveness of the proposed method on six fracture mechanics problems. The neural network had three hidden layers in the six fracture mechanics problems, each consisting of 50 neurons.

In [20] the authors propose incorporating prior knowledge in neural network models using genetic algorithms for a metallurgy application. The authors train and test a vanilla NN on a continuous cooling transformation data set

with seven inputs and six outputs. They find that the resulting NN does not conform to metallurgical engineering prior-knowledge. Accordingly, they propose a genetic-algorithm-based multi-objective cost function constrained via metallurgy prior-knowledge. The vanilla NN and the proposed NN models both have one hidden layer with twenty neurons. The proposed model is reported to improve the mean-square-error and almost completely remove the metallurgy prior-knowledge inconsistencies studied.

Jagtapa et al. [21] introduce an optimize-able adaptive hyperparameter that changes the slope of the activation function to tune the PGNN architecture. They show that the proposed adaptive activation function results in a faster loss function decay and a smaller $\mathcal{L}_2$ regularization error. Even though the proposed method is tune-able for any number of hidden layers, the authors choose different numbers of hidden layers and units for each case study with no explanation. A more comprehensive study would include a sensitivity study on the effects of the number of hidden layers, number of hidden units, and training data set size on the empirical performance of the proposed method.

The work in [22] proposes a Probabilistic Physics-guided Neural Network (PPgNN) with a novel model architecture to extend the standard neural network approach for fatigue data analysis. Among other methods, the proposed PPgNN utilizes a custom loss function, physics-constrained loss optimization, and a custom partially connected neural network architecture. Their proposed PPgNN variant has two hidden layers. Hidden layer one consists of 20 hidden units using the tanh activation function. Hidden layer two consists of two units; unit one uses the linear activation function, and unit two uses the exponential linear unit activation function. The output layer has one neuron which uses the linear activation function. The model's final outputs are the output layer neuron and the second hidden unit of the second hidden layer. The proposed PPgNN is tested on fatigue data analysis data and is more consistent with the domain knowledge relative to a neural network without physics-guidance. With further validation, the authors claim that the proposed framework is not limited to fatigue data analysis and can be applied to other survival data analyses via adjusting the constraints and the network architecture according to physics knowledge.

In [23] the authors propose novel methods to integrate semantic knowledge into the learning to improve the accuracy of convolutional neural networks (CNNs) on image classification tasks. The work presents a novel use of knowledge distillation, collective classification, training phases, and hyperparameter cross-validation for prior guided CNNs. They use cross-validation as a heuristic hyperparameter tuning method to tune the penalty of each predicate. The authors test different (1) modeling case studies, (2) data set sizes, and (3) CNN architectures. The data set size ablation study shows that the smaller the training data set size, the more significant the reduction of the empirical error of the proposed methods. The work in [23] has no investigation on the random model weight initialization effects on the empirical error mean and variance.

The number of hidden units dictates the NN/PGNN model's complexity and is commonly between 5 and 100 [24]. The number of units should be sufficiently large to capture the data non-linearity but not too large to avoid over-fitting and maintain the model's smoothness between the in-sample training data points. Typically, the number of hidden units is chosen via application-specific expert knowledge or empirical experimentation. The work in [12, 6, 16, 13, 25] among several other works either sets the number of hidden units based on expert knowledge or reports utilizing sensitivity studies without providing a systematic framework or a statistical analysis on the empirical and prior-consistency performance. The authors in [16] note that the interplay between the physics-guided NN architecture/training procedure and the complexity of the modeling data is poorly understood and propose a Bayesian approach to monitor the variance of the predictive posterior distribution as possible future work.

The authors in [15] note that their method requires systematic searching and sampling procedures to find the optimal physics-guided NN architecture and set it as future work. The work in [16] states that designing the correct physics-guided NN architecture is an open research question and confirms that they have observed that a specific physics-guided NN architecture that yields accurate results for one data set can fail for another data set. Therefore, there is a need for a systematic framework along with statistical analysis.

Another gap in the PGNN literature is a lack of statistical analysis of PGNNs, including:

- A lack of statistical analysis on the mean and variance of the empirical and prior-consistency errors across different random weight initialization of PGNN models.
- A lack of investigation with statistical tests on the effect of the data set size on the mean of the empirical and prior-consistency errors.
- A lack of investigation with statistical tests on the correlation between the empirical and prior-consistency

errors.

- A lack of investigation with statistical tests on the effect of the loss weight hyperparameters on the mean of the empirical and prior-consistency errors.

For more detailed literature and surveys on PGNN gaps and methods of incorporating prior knowledge into machine learning, we direct the reader to [26], [27], and [28]. In this work, we summarize the literature review by highlighting the need for (1) a systematic framework to set the number of hidden units for PGNNs, (2) statistical analysis on the performance of PGNNs under varying settings, and (3) methods to verify if and when a PGNN model can be trusted. In this work, we aim to address the gaps discussed via the following contributions towards prior-knowledge guided feedforward neural networks (PGNNs):

1. Model structure optimization via the Hidden Unit Number: A framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency for prior-knowledge guided feedforward neural-networks (model PGNN_1).
2. Model structure optimization via Prior-knowledge-guided cross-validation: In addition to using the prior-knowledge in the model training step, we propose utilizing the prior-errors as part of the cross-validation performance index (Eq. 3). Incorporating the prior-consistency error along with the empirical error in the validation process improves generalizability (model PGNN_2).
3. Statistical study on the empirical and prior-consistency error performance of PGNNs under the effects of random model weight initialization, data set size, hyperparameters, and the correlation between the empirical prior-consistency errors.

The remainder of the paper is organized as follows. The Methodology section provides the problem formulation and methods contributions 1 and 2. The Evaluation section covers the methods used to test and benchmark the proposed methods against reference models. The Results and Analysis Sections introduce the benchmark testing data, demonstrate the empirical and physics-consistency errors of the NN/PGNN models, and discuss the results. The conclusion covers the main benefits, limitations, and future work for the proposed framework.

2. Methodology

In this work, we utilize prior knowledge constraint losses in the cost function of a multi-layer perceptron feedforward neural network model. The first PGNN model, PGNN_1, differs from the NN model by incorporating the prior-based losses in the weight optimization function. This work's contributions are: (1) PGNN_1 differs from the work in the literature on PGNN models by optimizing the number of hidden units, and (2) PGNN_2 incorporates the same methods in PGNN_1, but additionally incorporates the prior-error in the cross-validation process.

In the following Subsections, the problem formulation and preliminaries are provided in Subsection 2.1. Subsections 2.3 and 2.2 will discuss the methodologies of contributions 1 and 2, respectively. Finally, Subsection 3.1 provides the model performance testing methods.

2.1. Preliminaries

A typical NN optimization criterion typically has two loss functions: an empirical loss function $\mathcal{L}_\text{e}$ and a regularization loss function $\mathcal{L}_\text{r}$ (Eq. 1). To incorporate prior knowledge, we add the prior knowledge-based loss function $\mathcal{L}_\text{p}$ to the optimization criterion as in literature (Eq. 2) [15, 13, 21]. The regularization and prior-based loss functions have the weights $\rho_\text{r} \geq 0$ and $\rho_\text{p} \geq 0$, respectively (Eq. 1 and 2). Accordingly, the framework proposed leads to three hyper-parameters: regularization loss weight (ρ_r), prior-based loss weights (ρ_p), and the hidden units number (j).

$$\mathcal{L}_{NN} = \mathcal{L}_\text{e} + \rho_\text{r}\mathcal{L}_\text{r} \tag{1}$$

where $\mathcal{L}_\text{e}$ is the empirical loss , $\mathcal{L}_\text{r}$ is the regularization loss, and ρ_r is the regularization loss weight.

$$\mathcal{L}_{PGNN} = \mathcal{L}_\text{e} + \rho_\text{r}\mathcal{L}_\text{r} + \rho_\text{p}\mathcal{L}_\text{p}, \tag{2}$$

where $\mathcal{L}_{\mathrm{p}}$ is the prior loss and ρ_{p} is the prior-based loss weight.

The cost function is non-linear and multi-modal (Eq. 2). Therefore, the networks initial random weights determine the cost function solution (a local minimum), and a training algorithm does not guarantee to find the global minimum. Furthermore, if the training data set is randomly sampled from an available data set, then the training data set might adversely be biased (i.e., not a representative of the physics phenomenon of study). Therefore, data models must be trained and validated with reliable empirical evidence. It is also necessary to ensure that performance differences between any proposed methods and reference models are due to the proposed methods and not effects such as biased data sets and random weight initialization. Accordingly, to validate the models, repeated k-fold cross-validation (CV) is applied for the following reasons: (1) make efficient use of the data to do training and validation, (2) reduce the chance of bias and variance in the validation performance index, and (3) address the local-minimum problem. The number of CV folds was chosen to be $k = 5$ to reduce the variance in the validation performance index. Finally, for every set of m value (number of hidden units) and random initial weights, the 5-fold CV is repeated 3 times using re-divided data subsets to minimize the occurrence of misleading results due to bias dominated data subsets. In the literature, the validation performance index for vanilla NNs and PGNN model has been the empirical error.

2.2. Model Structure Optimization via the Hidden Unit Number

In contribution (1), we train the PGNN model across a linearly spaced vector of 8 values within $[30, 100]$ to estimate the best number of hidden unit values m^* value based on the empirical error. A second fine search is then performed with a finer vector of 10 linearly spaced points around the best-estimated value $m^* \pm 10$. The model (PGNN_1) with the best cross-validation empirical error is chosen. Optimizing the number of hidden units via a line search eliminates the data-set/model-structure application dependency for prior-knowledge-guided feedforward neural-networks.

2.3. Model Structure Optimization via Prior-knowledge-guided Cross-validation

In contribution (2), we propose utilizing the prior error as part of the cross-validation performance index to optimize the best number of hidden units (PGNN_2, Eq. 3). Note that the prior-based errors $\mathcal{E}_{\mathrm{p}}$ are application dependent and would therefore have different means and standard deviations per application. If the empirical error ($\mathcal{E}_{\mathrm{e}}$) and the prior-based errors differ by orders of magnitude, then it is possible for the cross-validation error function to be dominated by either errors. Accordingly, the empirical and prior errors in Eq. 3 have been re-scaled via min-max normalization.

$$\underline{\mathcal{E}} = \frac{\underline{\mathcal{E}}_{\mathrm{e}} - min(\underline{\mathcal{E}}_{\mathrm{e}})}{max(\underline{\mathcal{E}}_{\mathrm{e}}) - min(\underline{\mathcal{E}}_{\mathrm{e}})} + \frac{\underline{\mathcal{E}}_{\mathrm{p}} - min(\underline{\mathcal{E}}_{\mathrm{p}})}{max(\underline{\mathcal{E}}_{\mathrm{p}}) - min(\underline{\mathcal{E}}_{\mathrm{p}})}. \tag{3}$$

3. Evaluation

The half sum squared error (HSSE) metric was chosen for the empirical loss function in Eq. 1 and 2. The regularization loss function was selected to be $\mathcal{L}_2$ regularization to penalize large weights in the model and reduce data over-fitting.

$$\mathcal{L}_{\mathrm{e}} = \frac{1}{2} \sum_{i=1}^{n} \left(\underline{\hat{y}}^{(i)} - \underline{y}^{(i)} \right)^2, \tag{4}$$

where n is the number of data points and $\hat{y}_i$ and y_i are the prediction and target for the k^{th} input vector.

$$\mathcal{L}_{\mathrm{r}} = \frac{1}{2} \underline{w}^T \underline{w}, \tag{5}$$

where $\underline{w}$ is a vector of the model weights.

We chose to use the boundary condition constraint (scenario 1) in Eq. 6 as the prior loss function to demonstrate the proposed methods in the main result in Subsection 4.1. We also investigate using the initial condition constraint (Scenario 2) and a horizontal symmetry constraint (Scenario 3) as prior loss functions (Eq. 7 and 8) to assess the impact of using different priors on the proposed methods (Subsection 4.2). The prior losses are separately applied to the reference and proposed models, where scenarios 1 to 3 are the boundary, initial, and symmetry prior, respectively. To quantify the prior-knowledge violations, we utilize the HSSE metric (Eq. 6, 7 8).

$$\mathcal{L}_{\mathrm{p}} = \frac{1}{2}\sum_{i=1}^{p}\left(\underline{\hat{y}}^{(i)}\left(\underline{X}_{\mathrm{b}}^{(i)}\right) - \underline{y}^{(i)}\left(\underline{X}_{\mathrm{b}}^{(i)}\right)\right)^2, \tag{6}$$

where p is the number of samples in the boundary dataset.

$$\mathcal{L}_{\mathrm{p}} = \frac{1}{2}\sum_{i=1}^{1}(\hat{y}^{(i)}(0,0) - y^{(i)}(0,0))^2. \tag{7}$$

$$\mathcal{L}_{\mathrm{p}} = \frac{1}{2}\sum_{i=1}^{n}(\underline{\hat{y}}^{(i)}(-\underline{x}_1, \underline{x}_2) - \underline{y}^{(i)}(\underline{x}_1, \underline{x}_2))^2. \tag{8}$$

The $\mathcal{L}_2$ regularization loss weight (ρ_{r}) and the prior loss weight (ρ_{p}) dictate the importance of model smoothness (complexity) and prior-consistency, respectively. Since ρ_{r} and ρ_{p} are scalars, it is possible to train NN/PGNNs with different combinations of ρ values and test the NN/PGNNs on a new data sample to identify the best model (empirical error wise). A typical NN example of complexity control is a one-grid search to find the optimal regularization weight (ρ_{r}), which results in the best PI. Similarly, a two-grid search is used to find the optimal function width and regularization weight of a Gaussian radial basis function model. State-of-the-art methods involve online adaptive methods to update the weights (ρ_{r} and ρ_{p}) during training. Optimizing the weights is outside the scope of this work; therefore we chose to set them as constants for the main result and perform a hyperparameter sensitivity study in Subsections 4.1 and 4.6, respectively.

The scaled conjugate gradient (SCG) back-propagation method was used to minimize the loss function (Eq. 1 and 2) and find the optimal weights. Note that each input feature from each case study was transformed separately by z-score standardization (i.e. zero mean and unit standard deviation). The optimization algorithm has two stopping criteria; the precision of the objective function and the weights at the solution. The two criteria were set to 10^{-4}. The precision in the line search parameter space was set to Netlab's default value (10^{-3}) [29]. The optimization algorithm was allowed a maximum of 1000 iterations to find the solution.

The root mean square error (RMSE) metric was chosen for the empirical and prior-based CV error functions (Eq. 9, 10, 11, and 12). Note that the summation stop values in Eq. 9, 10, and 12 are divided by k to take into account the CV folds.

$$\mathcal{E}_{\mathrm{e}} = \sqrt{\frac{\sum_{i=1}^{n/k}(\underline{\hat{y}}^{(i)} - \underline{y}^{(i)})^2}{n/k}}. \tag{9}$$

$$\mathcal{E}_{\mathrm{p}} = \sqrt{\frac{\sum_{i=1}^{p/k}\left(\underline{\hat{y}}^{(i)}(\underline{X}_{\mathrm{b}}^{(i)}) - \underline{y}^{(i)}(\underline{X}_{\mathrm{b}}^{(i)})\right)^2}{p/k}}. \tag{10}$$

$$\mathcal{E}_{\mathrm{p}} = \sqrt{(\hat{y}(0,0) - y(0,0))^2}. \tag{11}$$

$$\mathcal{E}_{\mathrm{p}} = \sqrt{\frac{\sum_{i=1}^{n/k}\left(\underline{\hat{y}}^{(i)}\left(-\underline{x}_1^{(i)}, \underline{x}_2^{(i)}\right) - \underline{y}^{(i)}\left(\underline{x}_1^{(i)}, \underline{x}_2^{(i)}\right)\right)^2}{n/k}}. \tag{12}$$

Table 1: Empirical RMSE mean and standard deviation range across 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark Function	NN−100	PGNN−100	$PGNN_1$	$PGNN_2$
McCormick	0.11 ± 0.01	0.11 ± 0.02	0.11 ± 0.04	0.10 ± 0.01
Griewank	0.14 ± 0.03	0.15 ± 0.03	0.13 ± 0.01	0.13 ± 0.01
Ackley	0.59 ± 0.00	0.59 ± 0.01	0.58 ± 0.01	0.58 ± 0.01
Branin	0.88 ± 0.13	0.88 ± 0.16	0.84 ± 0.14	0.88 ± 0.14
Styblinski–Tang	0.90 ± 0.24	0.92 ± 0.22	0.93 ± 0.22	0.93 ± 0.24
Booth	1.36 ± 0.12	1.41 ± 0.41	1.30 ± 0.11	1.31 ± 0.12
Three-hump Camel	2.13 ± 0.63	2.01 ± 0.43	1.82 ± 0.30	1.87 ± 0.31
Bukin N.6	2.17 ± 0.12	2.21 ± 0.17	2.16 ± 0.14	2.16 ± 0.14
Himmelblau's	3.57 ± 1.74	3.72 ± 1.71	2.95 ± 0.38	3.18 ± 0.71
Zakharov	15.11 ± 5.06	14.76 ± 3.07	13.11 ± 2.48	13.90 ± 3.00
Bohachevsky	21.11 ± 3.67	21.74 ± 4.08	19.06 ± 2.96	19.21 ± 2.85
Rosenbrock	620.93 ± 279.05	644.92 ± 274.50	487.82 ± 136.64	538.26 ± 215.55
Beale	1252.28 ± 313.79	1250.56 ± 359.89	1146.71 ± 286.44	1086.34 ± 252.09
Goldstein–Price	19110.69 ± 6111.54	18938.97 ± 5574.39	15712.85 ± 3940.58	15916.60 ± 3574.84

The reference and proposed models consist of two input neurons ($d = 2$), a hidden layer with a non-linear ReLu activation function, an output layer with one neuron ($c = 1$) and a linear activation function. Below is a list of the differences between the reference and proposed models.

Reference Models:

- NN−100: A vanilla NN with 100 hidden units (no prior usage in the loss or validation performance index).
- PGNN−30: A PGNN with 30 hidden units (no prior usage in the validation performance index).
- PGNN−65: A PGNN with 65 hidden units (no prior usage in the validation performance index).
- PGNN−100: A PGNN with 100 hidden units (no prior usage in the validation performance index).

Proposed Models:

- $PGNN_1$: A PGNN network where the number of hidden units is optimized via a line search as detailed in Subsection 2.2 (no prior usage in the validation function).
- $PGNN_2$: A PGNN network where the number of hidden units is optimized via a line search (Subsection 2.2) and the prior error is utilized in the validation performance index (Subsection 2.3).

3.1. Model Testing

Following training and validation, an out-of-sample (OOS) empirical data set with 1000 data points was used to test the models RMSE empirical performance (Eq. 9 with $k = 1$). The OOS empirical data set was also utilized to test the symmetry prior-consistency RMSE (Eq. 12 with $k = 1$). Similarly, an out-of-sample (OOS) boundary data set with 4000 data points was used to test the models RMSE empirical performance ((Eq. 10 with $k = 1$). The initial prior-consistency RMSE performance was tested via Eq. 11.

The OOS data sets were reserved and only used to test the models after they were trained and validated. Note that each PGNN/NN model was trained and validated 24 times using randomly initialized weights, and the averaged results are provided in Subsection 4.

Table 2: Boundary prior-consistency RMSE mean and standard deviation range across 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark Function	NN−100	PGNN−100	$PGNN_1$	$PGNN_2$
McCormick	0.21 ± 0.03	0.23 ± 0.04	0.22 ± 0.09	0.20 ± 0.02
Griewank	0.31 ± 0.05	0.32 ± 0.05	0.28 ± 0.03	0.27 ± 0.04
Ackley	1.01 ± 0.02	1.00 ± 0.02	0.99 ± 0.02	0.97 ± 0.02
Branin	2.50 ± 0.47	2.50 ± 0.51	2.33 ± 0.54	2.31 ± 0.48
Styblinski–Tang	2.91 ± 0.75	2.85 ± 0.81	2.99 ± 0.83	2.78 ± 0.74
Booth	2.24 ± 0.31	2.35 ± 0.25	2.19 ± 0.21	2.18 ± 0.19
Three-hump Camel	12.16 ± 3.48	11.08 ± 2.72	10.34 ± 1.78	10.19 ± 1.80
Bukin N.6	2.94 ± 0.21	2.99 ± 0.27	2.90 ± 0.19	2.90 ± 0.23
Himmelblau's	9.40 ± 2.59	9.68 ± 2.38	8.17 ± 1.05	8.41 ± 1.32
Zakharov	42.21 ± 15.65	42.04 ± 13.16	35.87 ± 6.48	36.36 ± 7.83
Bohachevsky	40.42 ± 8.17	40.57 ± 7.29	34.73 ± 7.50	33.73 ± 7.91
Rosenbrock	1992.64 ± 548.91	2056.07 ± 600.60	1869.55 ± 403.32	1921.96 ± 451.53
Beale	4262.21 ± 831.87	4339.32 ± 1025.90	4020.72 ± 851.43	3843.76 ± 761.80
Goldstein–Price	66484.35 ± 16126.71	67140.18 ± 13743.65	59491.66 ± 11529.53	60214.83 ± 10469.91

4. Results and Analysis

4.1. Benchmark function testing

To test the empirical and prior-consistency error performance of the proposed methods, we test 14 benchmark case studies varying from convex functions with a unique minimum to non-convex multi-modal functions (Table Appendix A). Further, we analyze why the proposed methods improve the empirical and prior-consistency error performance for some functions but not others and what dictates the level of improvement. For each modeling case study a boundary data set $[X_b, y(X_b)]$ of $p = 4000$ uniformly distributed data points was used for the prior-knowledge as discussed in Subsection 2.1. Tables 1 and 2) show the empirical and prior-consistency RMSE mean and standard deviation using a data set size of 1000, regularization weight of $1e-3$ for the NN model, and regularization and a prior weigh of $1e-5$ and $1e-6$ for the PGNN models, respectively. The median and $[0.25, 0.75]$ quantiles of the optimal number of hidden units selected by the proposed methods are shown in Table 7.

The percentage change in the empirical and prior-consistency RMSEs between the proposed methods and the reference models (Tables 3 and 4). The cells highlighted in Tables 3 and 4 indicate a statistically significant percentage change ($P< 0.05$) based on an independent two-sample multivariate Hotelling's test (Appendix Appendix B Tables B.11 and B.12). Similarly, the percentage change in the variance of the empirical and prior-consistency RMSEs between the proposed methods and the reference models (Tables 5 and 6). The cells highlighted in Tables 5 and 6 indicate a statistically significant percentage change ($P< 0.05$) based on a two-sample F-test for equal variance (Appendix Appendix B Tables B.13, B.14, B.15, and B.16).

The first result to note is the lack of any statistically significant difference in the empirical and prior-consistency error means and variances between the vanilla NN and PGNN−100 for the 14 benchmark functions (Tables 3, 5, and 6). The vanilla NN and PGNN−100 performance show that given an identical network architecture, a PGNN model may not out-perform a vanilla NN if the loss function weight hyperparameters are not tuned. There is also no statistically significant difference between $PGNN_1$ and PGNN2 empirical and prior-consistency mean errors except for the Ackley function (Table 4). However, the effect of the different number of hidden units between $PGNN_1$ and $PGNN_2$ is seen in the performance improvements of $PGNN_1$ and $PGNN_2$ relative to the reference models.

By optimizing the network architecture, $PGNN_1$ outperformed (empirical and prior-consistency error-mean-wise) the reference NN and PGNN-100 models for 3 and 6 functions, respectively (Tables 3 and 4). $PGNN_1$ reduced the empirical error variance relative to the reference NN and PGNN-100 for 6 and 3 functions, respectively (Table 5). $PGNN_1$ also reduced the prior-consistency error variance relative to the reference NN and PGNN-100 for 4 and 3 functions, respectively (Table 5).

Table 3: Percentage change in the empirical and prior-consistency RMSEs for the 14 benchmark functions comparing the proposed methods to the other tested methods using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively (24 samples per model variant since we test 24 random weight initialization).

Benchmark	[NN,PGNN−100]		[NN,PGNN$_1$]		[NN,PGNN$_2$]	
Function	Empirical	Prior	Empirical	Prior	Empirical	Prior
McCormick	1.54	4.78	2.74	2.11	−5.63	−7.89
Griewank	1.73	2.55	−12.09	−11.41	−11.84	−12.17
Ackley	−0.07	−0.45	−0.44	−1.86	−0.37	−3.93
Branin	0.09	0.02	−4.56	−6.96	−0.31	−7.88
Styblinski–Tang	1.46	−1.84	2.41	2.96	3.10	−4.51
Booth	3.48	5.00	−4.37	−2.39	−3.77	−2.64
Three-hump Camel	−5.80	−8.85	−14.62	−14.91	−12.11	−16.16
Bukin N.6	1.75	1.74	−0.44	−1.32	−0.75	−1.09
Himmelblau's	4.41	3.02	−17.40	−13.10	−10.7	−10.51
Zakharov	−2.36	−0.41	−13.26	−15.03	−8.04	−14.10
Bohachevsky	2.96	0.38	−9.73	−14.08	−9.04	−16.54
Rosenbrock	3.86	3.18	−21.44	−6.18	−13.31	−3.55
Beale	−0.14	1.81	−8.43	−5.67	−13.25	−9.82
Goldstein–Price	−0.90	0.99	−17.78	−10.52	−16.71	−9.43

For the McCormick function, PGNN$_1$ degraded the percentage change in the empirical and prior-consistency errors variance relative to NN−100 and PGNN−100. The performance degradation in PGNN$_1$ with the McCormick function can be associated with the hidden unit results in Table 7 since PGNN$_1$ had the largest quantile range $[0.25, 0.75]$ of 39 hidden units. A larger hidden unit quantile range is likely to correlate with the variance in the empirical and prior-consistency errors positively. The reason for PGNN$_1$'s performance with the McCormick function is not clear, but it is likely to do with the data set size relative to the complexity of the function and the untuned hyperparameters.

PGNN$_2$ outperformed (empirical and prior-consistency error-wise) the reference NN and PGNN-100 models for 6 and 8 functions, respectively (Tables 3 and 4). PGNN$_2$ reduced the empirical error variance relative to the reference NN and PGNN-100 for 5 and 3 functions, respectively (Table 5). PGNN$_2$ also reduced the prior-consistency error variance relative to the reference NN and PGNN-100 for 5 and 3 functions, respectively (Table 5).

The discrepancy in the error improvements between PGNN$_1$ and PGNN$_2$ is due to PGNN$_2$ validating the prior-consistency of the model in addition to its empirical generalizability. In Table 7 we see that PGNN$_2$ has resulted in a smaller, larger, and equal number of hidden units median for 7, 5, and 2 functions, respectively. We also note that PGNN$_2$ has a larger hidden unit quantile range $[0.25, 0.75]$ than PGNN$_1$ for 12 out of the 14 benchmark functions. By incorporating the prior-consistency error in the validation performance index, PGNN$_2$ re-enforces prior-consistency. PGNN$_2$ outperforms (empirical/prior error mean and variance) PGNN$_1$ relative to the reference models by selecting a suitable number of hidden units to avoid under-fitting or over-fitting depending on the training/prior data landscape versus the data set size.

The discrepancy in the error improvements of the proposed methods between the benchmark functions is likely to be related to (1) discrepancy in the value of the prior knowledge for each benchmark function, (2) training data set size, (3) prior data set size. The training data set size effects are examined in Subsection 4.2. The prior data set size effects are not examined in this work and are left as future work.

A possible explanation for the discrepancy in the error improvements is that the prior knowledge has a different value for each benchmark function depending on the training/prior data landscape. For example, the boundary prior is likely to be less valuable if, for a given 3-dimensional function, the target values across a line on the 2-dimensional landscape are highly non-linear and non-monotonic. We propose a simple measure of boundary/training mean output ratio to give insight on the value of the boundary prior. The prior output mean, training output mean, and boundary/training output mean ratio for each benchmark function are shown in Table 8). Using the proposed measure, we measure Spearman's correlation, and the statistical significance between the empirical and prior-consistency RMSE

Table 4: Percentage change in the empirical and prior-consistency RMSEs for the 14 benchmark functions comparing the proposed methods to the other tested methods using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively (24 samples per model variant since we test 24 random weight initialization).

Benchmark	[PGNN−100,PGNN$_1$]		[PGNN−100,PGNN$_2$]		[PGNN$_1$,PGNN$_2$]	
Function	Empirical	Prior	Empirical	Prior	Empirical	Prior
McCormick	1.17	−2.55	−7.06	−12.09	−8.14	−9.79
Griewank	−13.58	−13.61	−13.34	−14.36	0.28	−0.86
Ackley	−0.37	−1.42	−0.30	−3.50	0.07	−2.11
Branin	−4.65	−6.98	−0.40	−7.90	4.45	−0.99
Styblinski–Tang	0.93	4.89	1.61	−2.72	0.67	−7.25
Booth	−7.59	−7.04	−7.01	−7.28	0.63	−0.26
Three-hump Camel	−9.36	−6.65	−6.70	−8.02	2.94	−1.47
Bukin N.6	−2.15	−3.02	−2.46	−2.78	−0.31	0.24
Himmelblau's	−20.89	−15.64	−14.54	−13.14	8.03	2.97
Zakharov	−11.17	−14.68	−5.82	−13.75	6.02	1.09
Bohachevsky	−12.33	−14.40	−11.65	−16.86	0.77	−2.86
Rosenbrock	−24.36	−9.07	−16.54	−6.52	10.34	2.80
Beale	−8.30	−7.34	−13.13	−11.42	−5.26	−4.40
Goldstein–Price	−17.03	−11.39	−15.96	−10.31	1.30	1.22

Table 5: Percentage change in the variance of the Empirical RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark	Percentage change (%)					
Function	[NN−100, PGNN−100]	[NN−100, PGNN$_1$]	[NN−100, PGNN$_2$]	[PGNN−100, PGNN$_1$]	[PGNN−100, PGNN$_2$]	[PGNN$_1$, PGNN$_2$]
McCormick	46.31	951.52	−20.86	618.71	−45.91	−92.47
Griewank	−6.32	−73.56	−70.02	−71.77	−68.00	13.36
Ackley	83.12	26.99	10.48	−30.65	−39.67	−13.00
Branin	40.35	6.99	5.81	−23.77	−24.61	−1.10
Styblinski–Tang	14.35	−16.23	−0.31	−2.19	16.40	19.00
Booth	31.33	−19.60	−2.87	−38.78	−26.04	20.81
Three-hump Camel	−54.14	−76.99	−75.72	−49.83	−47.07	5.50
Bukin N.6	106.06	29.75	31.38	−37.03	−36.24	1.25
Himmelblau's	−3.36	−95.20	−83.63	−95.03	−83.06	241.16
Zakharov	−35.13	−75.87	−64.91	−62.81	−45.91	45.45
Bohachevsky	23.80	−34.73	−39.50	−47.27	−51.13	−7.31
Rosenbrock	−3.23	−76.02	−40.33	−75.22	−38.34	148.86
Beale	31.54	−16.67	−35.46	−36.65	−50.93	−22.55
Goldstein–Price	−16.81	−58.43	−65.79	−50.03	−58.87	−17.70

percentage change and the boundary/training mean output ratio (Table 9). From Table 9 we see that as the boundary/training mean output ratio increases, the percentage change between the NN and the proposed methods (PGNN$_1$ and PGNN$_2$) decreases with a strong negative correlation ($\rho < -0.60$). Similarly, as the boundary/training mean output ratio increases, the percentage change between the PGNN−100 and the proposed methods decreases with a moderate negative correlation ($\rho < -0.40$). The negative correlation indicates that the prior is more valuable (empirical and prior-consistency error-wise) when the prior target and training data-target means differ more. This result does not rule out the influence of other factors in the impact of PGNN$_1$ and PGNN$_2$ on the empirical and prior-consistency error performance relative to the reference models.

This study has shown that optimizing the network architecture (PGNN$_1$) statistically significantly improves the

empirical and prior-consistency RMSE means relative to the reference models but can result in a larger empirical and prior-consistency variance across randomly initialized model weights. Using the prior knowledge and the empirical error in the validation process, $PGNN_2$, results in a more informed performance index for model candidate selection and statistically significantly improves the empirical and prior-consistency RMSE mean and variance relative to the reference models. $PGNN_2$ statistically significantly improves the empirical and prior-consistency RMSE mean and variance for more benchmark functions than $PGNN_1$ relative to the reference models. Finally, The results invite further research on (1) the relationship between the empirical and prior data in the context of physics guided loss and validation PI functions, (2) training data landscape versus prior knowledge value, and (3) adaptive-weighting of the prior knowledge in PGNNs.

4.2. Data set size multivariate sensitivity study

The objective of this study is to investigate if the data set size affects the proposed methods' multivariate empirical and prior-consistency RMSE performance. To the best of the authors' knowledge, this is the first study to quantitatively examine the effect of the data set size on the multivariate empirical and prior-consistency error variance for PGNNs. Five training data sets were created for the Bohachevsky and the Ackley functions; sample sizes: 200, 600, 1000, 2000, and 4000 samples. The samples in each data set are uniformly distributed and bounded by the input ranges in Table A.10. We set a regularization weight of $1e-3$ for the NN model and regularization and a prior weight of $1e-5$ and $1e-6$ for the PGNN models. Three prior case studies were tested per benchmark function: (1) boundary prior, (2) initial prior, and (3) symmetry prior. The empirical and prior-consistency RMSEs are presented in Appendix Appendix C Tables C.17 and C.18.

We perform a one-way Multivariate Analysis of Variance (MANOVA) for comparing the multivariate means of the empirical and prior-consistency errors, grouped by data set size. The dimension result is an estimate of the dimension of the space containing the group means. The P values test whether the means lie in the space of dimension 0, 1, or 2 (critical P-value 0.05). The null hypothesis is that the multivariate means of all data set sizes are equal and that any difference observed in the multivariate means is random.

From Table C.19, $PGNN_1$ and $PGNN_2$ MANOVA results for the Bohachevsky case study show that there is a 1-dimensional statistically significant difference ($P < 0.05$) in the multivariate means for the three prior scenarios. However, the multivariate means may lie on the same line. From Table C.20, $PGNN_1$ and $PGNN_2$ MANOVA results for the Ackley case study show that there is a 2-dimensional statistically significant difference ($P < 0.05$) in the multivariate means for the three prior scenarios. The multivariate means may lie on the same plane in 2-dimensional space but not on the same line.

Qualitatively, the canonical analysis of the Bohachevsky case for $PGNN_1$ in Fig. C.1a, C.2a, and C.3a shows that: (a) data set size 200 is separate from and does not overlap the other data set sizes, (b) data set size 600, 1000, and 2000 are overlapping but with distinct centers, (c) data set sizes 2000 and 4000 are not clearly separable, (d) the first canonical variable approximately separates data set sizes 200, 600, 1000, and 2000, and (e) the second canonical variable, does not show clear separation between the data set sizes. The canonical analysis of the Bohachevsky case for $PGNN_2$ in Fig. C.4a, C.5a, and C.6a approximately has the same results as the Canonical analysis for $PGNN_1$.

Quantitatively, the Mahalanobis distance matrix of the Bohachevsky case for $PGNN_1$ and $PGNN_2$ in Fig. C.1b, C.2b, C.3b, C.4b, C.5b, and C.6b confirms the canonical analysis findings and demonstrates that the distance between the 2000 and 4000 is negligible relative to the distances between other data set sizes. To assess if the univariate distance between data set sizes is statistically significant we perform a one-way analysis of variance (ANOVA) and a multiple comparison test in Subsection Appendix D.

Qualitatively, the canonical analysis of the Ackley case for $PGNN_1$ in Fig. C.7a, C.8a, and C.9a shows that: (a) all data set sizes overlap with one or more other data set sizes but each has a distinct center, (d) the first canonical variable approximately separates the data set sizes into three groups (1) 200, (2) 600 and 1000, and (3) 2000 and 4000, and (e) the second canonical variable does not show clear separation between the data set sizes. The canonical analysis of the Ackley case for $PGNN_2$ in Fig. C.10a, C.11a, and C.12a approximately has the same results as the Canonical analysis for $PGNN_1$.

Quantitatively, the Mahalanobis distance matrix of the Ackley case for $PGNN_1$ and $PGNN_2$ in Fig. C.7b, C.8b, C.9b, C.10b, C.11b, and C.12b confirms the canonical analysis findings and demonstrates that the data set sizes can be separated into three groups (1) 200, (2) 600 and 1000, and (3) 2000 and 4000. Note that the distance between the

Table 6: Percentage change in the variance of the boundary prior-consistency RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark Function	Percentage change (%)					
	[NN−100, PGNN−100]	[NN−100, $PGNN_1$]	[NN−100, $PGNN_2$]	[PGNN−100, $PGNN_1$]	[PGNN−100, $PGNN_2$]	[$PGNN_1$, $PGNN_2$]
McCormick	23.88	660.26	−49.92	513.69	−59.58	−93.41
Griewank	−10.40	−59.27	−46.13	−54.54	−39.88	32.25
Ackley	1.67	11.52	−30.41	9.68	−31.55	−37.59
Branin	21.93	36.66	8.42	12.08	−11.08	−20.67
Styblinski–Tang	15.57	19.48	−3.31	3.39	−16.34	−19.08
Booth	−35.58	−52.19	−60.70	−25.79	−38.99	−17.79
Three-hump Camel	−39.24	−74.01	−73.29	−57.22	−56.04	2.75
Bukin N.6	63.33	−21.47	14.05	−51.92	−30.17	45.23
Himmelblau's	−15.62	−83.69	−73.86	−80.67	−69.03	60.21
Zakharov	−29.31	−82.85	−74.98	−75.74	−64.60	45.93
Bohachevsky	−20.26	−15.66	−6.08	5.77	17.79	11.36
Rosenbrock	19.72	−46.01	−32.34	−54.90	−43.48	25.33
Beale	52.09	4.76	−16.14	−31.12	−44.86	−19.95
Goldstein–Price	−27.37	−48.89	−57.85	−29.62	−41.97	−17.54

Table 7: The median and $[0.25, 0.75]$ qauntiles of the optimal number of hidden units selected by the proposed methods across the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark Function	$PGNN_1$ Median (quantile)		$PGNN_2$ Median (quantile)	
McCormick	80.00	(59.00 – 98.00)	83.00	(60.00 – 100.00)
Griewank	56.50	(42.00 – 70.00)	51.50	(38.50 – 66.50)
Ackley	77.00	(62.00 – 91.50)	64.00	(46.00 – 80.00)
Branin	96.00	(86.00 – 104.00)	84.50	(73.00 – 98.50)
Styblinski–Tang	95.50	(86.50 – 106.00)	97.00	(89.00 – 107.00)
Booth	102.00	(99.00 – 107.00)	105.00	(101.00 – 110.00)
Three-hump Camel	98.50	(86.00 – 103.00)	90.00	(81.50 – 100.00)
Bukin N.6	100.00	(90.50 – 105.00)	100.50	(93.00 – 103.00)
Himmelblau's	103.00	(94.50 – 110.00)	101.00	(92.00 – 107.00)
Zakharov	103.00	(94.00 – 108.50)	95.00	(83.00 – 101.50)
Bohachevsky	97.00	(90.00 – 100.50)	97.00	(88.00 – 105.00)
Rosenbrock	92.00	(90.00 – 96.00)	90.50	(87.00 – 98.00)
Beale	90.00	(78.50 – 100.00)	90.00	(77.00 – 100.00)
Goldstein–Price	92.50	(69.00 – 99.00)	94.00	(70.00 – 100.50)

2000 and 4000 data set sizes is smaller than the distance between the 600 and 1000 data set sizes. To assess if the univariate distance between data set sizes is statistically significant we perform a one-way ANOVA and a multiple comparison test in Subsection Appendix D.

The MANOVA study shows a 1-dimensional statistically significant difference ($P < 0.05$) in the multivariate empirical and prior-consistency means for the three prior scenarios in the Bohachevsky case study. However, the Ackley case study has a 2-dimensional statistically significant difference ($P < 0.05$) in the multivariate means. The change in the dimensional difference of the multivariate means between the Bohachevsky and Ackley functions is likely due to (1) the functions' complexity and (2) the value of the prior per function.

A prior is likely to significantly impact the empirical and prior-consistency errors as the function complexity increases with a fixed data set size. The value of a prior in a given case study depends on (1) the training data set

Table 8: The mean of the boundary data and the training data of the 14 benchmark functions.

Benchmark Function	Boundary output mean	Training output mean	Boundary/training output mean ratio
McCormick	13.79	7.63	1.81
Griewank	0.94	1.00	0.94
Ackley	12.05	9.68	1.24
Branin	88.15	54.93	1.60
Styblinski–Tang	108.12	−7.59	−14.24
Booth	747.39	389.24	1.92
Three-hump Camel	1139.11	255.57	4.46
Bukin N.6	149.14	121.38	1.23
Himmelblau's	357.00	137.34	2.60
Zakharov	8221.47	3695.84	2.22
Bohachevsky	199945.39	9940.28	2.01
Rosenbrock	316950.15	125540.62	2.52
Beale	41572.18	8614.62	4.83
Goldstein–Price	197803.51	50959.70	3.88

Table 9: Spearman's rho correlation and P value between the percentage change in RMSEs and the boundary/training mean output ratio across the 14 benchmark functions using a training data set size of 1000 samples.The model hyperparameters for this test include a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively. P-values are shaded to indicate a statistically significant correlation (critical P value 0.05).

	$\rho(P)$					
	[NN−100, PGNN−100]	[NN−100, $PGNN_1$]	[NN−100, $PGNN_2$]	[PGNN−100, $PGNN_1$]	[PGNN−100, $PGNN_2$]	[$PGNN_1$, $PGNN_2$]
Empirical	−0.24 (0.40)	−0.67 (0.01)	−0.76 (0.00)	−0.54 (0.05*)	−0.57 (0.03)	0.35 (0.23)
Prior	0.09 (0.77)	−0.53 (0.06)	−0.42 (0.14)	−0.47 (0.09)	−0.34 (0.24)	0.28 (0.33)

size and (2) the correlation between the empirical and the prior-consistency errors. If the data set size is small relative to the function complexity, then a prior is likely to be more useful in training and validating PGNNs, and vise versa. Similarly, if there is a strong correlation between the empirical and the prior-consistency errors, then a prior is likely to be more useful in training and validating PGNNs. In Subsection 4.4 we report and analyze the correlation between the empirical and the prior-consistency errors for the Bohachevsky and the Ackley functions.

For the Bohachevsky and Ackley functions, the canonical analysis and Mahalanobis distance matrix for the proposed methods with all three priors separated the 200 data set size results from the other data sets for $PGNN_1$ and $PGNN_2$. However, for both functions, the 2000 and 4000 data set sizes were not separable. The separation of the 200 data set size, and lack of separation between the 2000 and 4000 data set sizes are in line with previous studies and show that as the data set size increases, the impact on the empirical and prior-consistency errors reduces.

The canonical analysis and Mahalanobis distance matrix for the proposed methods with all three priors approximately separated the multivariate errors of the 600 and 1000 data set sizes for the Bohachevsky, but not the Ackley function. A possible explanation for the difference in multivariate error variance between the Bohachevsky and Ackley functions for the 600 and 1000 sample sizes is that the Ackley function is more complex than the Bohachevsky function. Since the Ackley function is more complex, smaller changes in the data set size are likely to impact the multivariate error performance less.

We conclude that the data set size can result in a 1-dimensional or 2-dimensional statistically significant difference ($P < 0.05$) in the multivariate empirical and prior-consistency means for convex and non-convex modeling problems up to a given data set size. The functions' complexity, the prior value, and the correlation between the empirical and the prior-consistency errors dictate the magnitude of the multivariate difference between data set sizes.

4.3. Data set size univariate sensitivity study

This study examines if and how the data set size and the modeling problem complexity affect the proposed methods' effect on the univariate empirical and prior-consistency RMSE performance. We perform an ANOVA test (omitted but available upon request) followed by a multiple comparison test using Tukey's honestly significant difference procedure to determine which data set sizes statistically differ (P> 0.05) from one another (Table D.21). The ANOVA and multiple comparison tests are performed on the Bohachevsky and the Ackley function results in Subsection 4.2.

For the Bohachevsky case study, the proposed methods ($PGNN_1$ and $PGNN_2$) have a statistically significant impact on the empirical RMSE mean for the three prior scenarios up to a data set size of 600 samples. The prior-consistency error means for initial, and symmetry priors are statistically significantly different up to a data set size of 600 samples. The boundary prior-consistency error mean is statistically significantly different up to a data set size of 1000 samples.

For the Ackley case study, the proposed methods ($PGNN_1$ and $PGNN_2$) have a statistically significant impact on the empirical RMSE means for all the prior scenarios up to a data set size of 4000 samples. The proposed methods ($PGNN_1$ and $PGNN_2$) also statistically significantly impact the prior-consistency RMSE means for the initial prior scenario up to a data set size of 4000 samples. For the boundary and symmetry prior scenarios, the proposed methods ($PGNN_1$ and $PGNN_2$) have a statistically significant impact on the prior-consistency RMSE means for data set sizes up to 2000 samples (Table D.21). Now that we have confirmed that the proposed methods' have a statistically significant impact on the empirical and prior-consistency RMSEs for some data set sizes, we will assess if there is a correlation between the data set size and the empirical/prior-consistency RMSEs.

The ANOVA study and multiple comparison test results align with previous studies; as the data set size increases, PGNN models are less likely to impact the empirical and prior-consistency errors. The proposed methods ($PGNN_1$ and $PGNN_2$) have a statistically significant impact on the empirical RMSE mean for the three prior scenarios up to a data set size of 600 samples for the Bohachevsky function and up to 4000 samples for the Ackley function. Similarly, the prior-consistency error means for the Bohachevsky function do not statistically differ for data set sizes larger than 600 for the initial and symmetry priors and 1000 samples for the boundary prior. However, for the Ackley function, the prior-consistency error means do not statistically differ for data set sizes larger than 4000 for the initial prior and 2000 for the boundary and symmetry priors. The Bohachevsky function is convex with a unique minimum and therefore requires a smaller training data set size relative to the Ackley function, which is non-convex and multi-modal. The difference in complexity between the two functions explains why the empirical and prior-consistency error variance saturates at a smaller data set size for the Bohachevsky function.

For data set size 1000 and larger, $PGNN_1$ for the Bohachevsky function achieves the same empirical error mean across all the three prior scenarios, indicating that with the proposed framework, the number of hidden units set by $PGNN_1$ dictates the performance, instead of the prior type/value. $PGNN_1$ empirical performance with the Bohachevsky function for data set sizes 1000, and larger is also likely because of the use of a constant prior weight instead of a tuned/adaptive weight in the loss function (Eq. 2).

Note that the initial prior is less beneficial than the boundary and symmetry priors for the Bohachevsky function since the Bohachevsky function is convex. Hence, there is no statistically significant impact on the empirical and prior-consistency RMSE means for data sets larger 600 samples.

We conclude this section by noting that the proposed methods ($PGNN_1$ and $PGNN_2$) affect the empirical and prior-consistency RMSEs depending on the modeling problem complexity and data set size. As the modeling problem complexity decreases and the data set size increases, the proposed methods are less likely to have a statistically significant impact (P> 0.05) on the empirical and prior-consistency RMSE.

4.4. Data set size and performance correlation study

The objective of this study is to determine the strength and direction of correlation between the data set size and the empirical and prior-consistency errors. We use Superman's rank correlation to measure the strength and direction of monotonic association between the data set size and the empirical RMSE in the results in Subsection 4.2. In Table E.22 we see a strong ($\rho > 0.60$) to very strong ($\rho > 0.80$) statistically significant (P< 0.05) negative correlation between the empirical RMSE and the data set size for the three prior scenarios of the Bohachevsky and Ackley case studies.

Similarly, we use Superman's rank correlation to measure the strength and direction of monotonic association between the data set size and the prior-consistency RMSE. From Table E.23 we see a very strong ($\rho > 0.80$) statistically significant (P< 0.05) negative correlation between the boundary and symmetry prior RMSE and the data

set size for the three prior scenarios of the Bohachevsky and Ackley case studies. For the initial prior, there is a weak ($\rho > 0.30$) statistically significant (P< 0.05) negative correlation for the Bohachevsky case study and a strong ($\rho > 0.60$) statistically significant (P< 0.05) negative correlation for the Ackley case study.

The statistically significant (P< 0.05) negative correlation between the data set size, and the empirical and prior-consistency RMSE are in line with previous studies. The weaker correlation between the initial prior-consistency error and data set size for the Bohachevsky function further supports the results obtained in Subsection 4.3 which show that the initial prior is less beneficial than the boundary and symmetry priors for the Bohachevsky function since the function is convex.

4.5. Empirical and prior-consistency error correlation study

This study aims to determine the strength and direction of correlation between the empirical and prior-consistency errors across the data set sizes examined in this work. A prior may not have value or correlate to the empirical error where a sufficiently representative training data set is available. Where the prior is valuable, the relationship between the empirical and prior-consistency errors can be a positive linear or monotonic correlation. The empirical and prior-consistency errors may not be strongly correlated if the prior is not significantly informative relative to the training data (e.g., the initial prior scenario with the Bohachevsky function). We use Superman's rank correlation to measure the strength and direction of monotonic association between the empirical RMSE and the prior-consistency RMSEs from the data set size sensitivity study results in Subsection 4.2. The result in Table F.24 shows the correlation and P-value for the three prior case studies in the Bohachevsky and the Ackley functions.

In Table F.24, for the Bohachevsky function, the reference and proposed models have a very strong ($\rho > 0.8$) statistically significant (P< 0.05) positive correlation between the empirical RMSE and the prior-consistency RMSE for the boundary and symmetry prior scenarios. For the initial prior scenario with the Bohachevsky function, the proposed methods have a statistically significant (P< 0.05) weak ($[0.20, 0.39]$) to moderate ($[0.40, 0.59]$) positive correlation between the empirical RMSE and the prior-consistency RMSE. For the initial prior scenario with the Bohachevsky function, the reference models NN-100, PGNN-65, and PGNN-100 have a statistically significant (P< 0.05) weak ($[0.20, 0.39]$) positive correlation between the empirical RMSE and the initial prior-consistency RMSE. The reference model PGNN-30 has no statistically significant (P< 0.05) correlation between the empirical RMSE and the initial prior-consistency RMSE.

In Table F.24, for the Ackley function, the reference and proposed models have a very strong ($\rho > 0.8$) statistically significant (P< 0.05) positive correlation between the empirical RMSE and the prior-consistency RMSE for the boundary and symmetry prior scenarios. For the initial prior scenario with the Ackley function, the reference and proposed models have a statistically significant (P< 0.05) strong positive correlation ($\rho > 0.6$) between the empirical RMSE and the initial prior-consistency RMSE.

The correlation between the empirical RMSE and the prior-consistency RMSE for the Bohachevsky and Ackley functions mostly differ in the initial prior case study. The Bohachevsky function has a weak to moderate positive correlation, while the Ackley function has a strong positive correlation. The correlation difference for the initial prior between the two functions is consistent with the findings regarding the smaller benefit of the initial prior especially for the Bohachevsky function discussed in Subsections 4.4 and 4.3.

The data set size and correlation between the empirical and the prior-consistency errors are also linked. For example, a boundary prior is likely more beneficial for small data sets if a strong symmetrical two-dimensional monotonic relation between the function's output and input data exists over the input data range up to the boundary condition (e.g., Bohachevsky, Ackley, or Griewank). On the other hand, the boundary prior can be less valuable where the data set size is small, and the correlation between the empirical and prior-consistency errors is weak (e.g., Zakharov or Three-hump Camel). The association between the data set size on the value of the boundary prior also accords with our earlier observations in Subsection 4.1, which showed better empirical/prior-consistency RMSE mean error improvements for the Bohachevsky, Ackley, and Griewank relative to the Zakharov and Three-hump Camel functions.

In summary, these results show a statistically significant (P< 0.05) positive correlation between the empirical RMSE and the prior-consistency RMSE for the three prior scenarios of the Bohachevsky and Ackley case studies. The positive correlation between the empirical RMSE and the prior-consistency RMSEs demonstrates the importance of developing prior-consistent NNs. The positive correlation between the empirical RMSE and the prior-consistency RMSEs further supports using the prior-consistency error in the validation performance index of PGNNs (PGNN_2).

4.6. Hyperparameter sensitivity investigation

The objective of this investigation is to determine whether the regularization loss weight and the prior loss weight hyperparameters have an impact on the RMSE performance of the proposed methods and the multivariate empirical and prior-consistency RMSE performance difference between the reference and the proposed models. To perform this investigation, we train the reference and proposed models with different combinations of empirical and prior-consistency weights and compare the multivariate empirical and prior-consistency RMSEs via a two-sample Hotelling's T^2 for independent samples.

Throughout the results in this Subsection, the NN−100 reference model was trained with a regularization weight $\rho_r = 1e-3$. The PGNN−30, PGNN−65, PGNN−100, $PGNN_1$, and $PGNN_2$ models were trained with the following $[\rho_r, \rho_p]$ loss weights: $[1e-3, 1e-4]$, $[1e-3, 1e-6]$, $[1e-3, 1e-6]$, $[1e-5, 1e-4]$, and $[1e-7, 1e-4]$ (Table G.25 and G.27). All the models were trained 24 times with randomly initialized weights for the Bohachevsky function using a training data set size of 1000.

The two-sample Hotelling's T^2 test is sensitive to violations of the assumption of equal variance and covariance. Therefore, we have to assess whether variance-covariance matrices are significantly different to decide whether to perform a homoscedastic or a heteroscedastic test. In this work, we use the multivariate statistical test Box's M (assuming multivariate normality) to check the equality of multiple covariance matrices. The Box M test is sensitive to violations of multivariate non-normality [30]. We note that the tested data does not have a univariate or multivariate normal distribution (results omitted). However, instead of taking on a Chi^2 approximation, Box's M test can take on an F-test approximation which is largely robust to normality violations [30, 31]. Although the Chi^2 approximation is more accurate for group sample sizes larger than or equal to 20, we use the F-test approximation due to its robustness to normality violations.

The null hypothesis for the Box M test is that the observed covariance matrices for the dependent variables are equal across groups. In other words, a non-significant test result (P> 0.05) indicates that the covariance matrices are equal. If the covariance matrices are not significantly different (homoscedastic) and the groups' sample size is at least 50, Hotelling's T^2 test takes a Chi^2 approximation; otherwise, it takes an F approximation. If the covariance matrices are significantly different (heteroscedastic), Hotelling's T^2 test takes a Chi^2 approximation.

We perform the Box's M and two-sample Hotelling's T^2 tests on to assess if changing the prior weight or the regularization weight has an effect on the multivariate empirical and prior-consistency RMSEs of the proposed models (Table G.26 and G.28).

The results from the prior weight $1e-4$ and the three regularization weights (72 samples per model variant, 24 random weight initialization by 3 regularization weights) were combined in a group for each model variant, and we then applied the Box's M and two-sample Hotelling's T^2 tests to compare the models (Table G.29). Finally, we combine the results from the three prior weights and the three regularization weights for each model variant (144 samples per model variant, 24 random weight initialization by 3 regularization weights, and 24 random weight initialization by 3 prior weights) and apply the Box's M and two-sample Hotelling's T^2 tests (Table G.30).

We report the Box's M statistic, Box's M statistic P significance level, T^2 statistic (F statistic or chi-square statistic), and T^2 statistic P significance level P significance level. The hyperparameter investigation study results in Table G.26 and G.28 show that $PGNN_1$ and $PGNN_2$ performance for the Bohachevsky function is not statically significantly affected by the prior loss weight or the regularization loss weight. Table G.29 and G.30 show that the proposed methods $PGNN_1$ and $PGNN_2$ have a statistically significant impact on the multivariate empirical and prior-consistency error means relative to the reference models for different prior and regularization loss weights.

In conclusion, (1) the proposed methods are not affected by the loss weight hyperparameters and (2) for a given set of hyperparameters, the proposed methods consistently improve the empirical and prior-consistency RMSE means relative to the reference models (Table G.29 and G.30). The exception to the empirical and prior-consistency RMSE mean improvements is for the initial prior case study, where there is no statically significant difference in the multivariate empirical and prior-consistency RMSE performance between $PGNN_2$ and PGNN−100. The lack of statically significant difference in the multivariate errors is consistent with the findings regarding the smaller benefit of the initial prior especially for the Bohachevsky function discussed in Subsections 4.4, 4.3, and 4.5.

5. Conclusion

The literature on physics-guided NNs (PGNN) shows that when PGNNs are expertly tailored to specific problems, they can perform better than traditional NNs for small and noisy synthetic/experimental data sets. Expertly tailored PGNNs have also been used as computationally-efficient and accurate counterparts to physics-based numerical simulations [16, 32]. However, the need for expert knowledge to tailor the PGNN structure (number of hidden layers and units) limits the applications and objectivity of PGNNs [15, 16].

In this investigation, the aim was to develop and statistically test a systematic framework to optimize the structure of PGNNs while ensuring empirical generalizability and prior-consistency. We propose a framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency ($PGNN_1$). In addition to using the prior-knowledge in the model training step, we propose utilizing the prior errors as part of the cross-validation performance index to reinforce prior-consistency ($PGNN_2$). The third contribution of this work was to perform statistical analysis on the empirical and prior-consistency error performance of PGNNs under varying settings to determine when and why the proposed methods are effective.

Optimizing the network architecture ($PGNN_1$) with repeated cross-validation statistically significantly improves the empirical and prior-consistency RMSE means relative to the reference models by selecting the number of hidden units with the smallest empirical error. However, a limitation of $PGNN_1$ is the possibility of a larger hidden unit quantile range across random model weight initialization. The $PGNN_1$ limitation is hypothesized to be due to the data set size versus the function's complexity and the untuned hyperparameters.

Using the prior knowledge and the empirical error in the validation process, $PGNN_2$, statistically improves the empirical and prior-consistency RMSE mean and variance relative to the reference models. By incorporating the prior-consistency error in the validation performance index, $PGNN_2$ re-enforces prior-consistency. $PGNN_2$ outperforms (empirical/prior error mean and variance) $PGNN_1$ relative to the reference models by selecting a suitable number of hidden units to avoid under-fitting or over-fitting depending on the training/prior data landscape versus the data set size. Further, the positive correlation between the empirical RMSE and the prior-consistency RMSEs supports utilizing the prior-consistency error in the validation performance index of PGNNs ($PGNN_2$).

The data set size investigation has shown that PGNNs result in a 1-dimensional or 2-dimensional statistically significant difference ($P < 0.05$) in the multivariate empirical and prior-consistency means for convex and non-convex modeling problems up to a 2000 sample data set size. The univariate analysis has shown a statistically significant difference ($P < 0.05$) in the empirical and prior-consistency errors for up to sample data set size 4000 and 2000, respectively. The functions' complexity, the prior value, and the correlation between the empirical and the prior-consistency errors dictate the magnitude of the multivariate difference between data set sizes. In the hyperparameter sensitivity investigation, we show that the proposed methods are not affected by the loss weight hyperparameters and consistently improve the empirical and prior-consistency RMSE means relative to the reference models.

Future work includes investigating practical applications, multi-prior loss functions, prior adaptive weighing, and the boundary data set size effect on PGNN performance. Additional studies should focus on the relationship between the prior type/value and the modeling problem landscape complexity to develop a complete picture of the importance of priors in the context of PGNNs.

Acknowledgment

This work was supported by The University of Sheffield and the UK EPSRC Future Manufacturing Hub - Manufacture using Advanced Powder Processes (MAPP) through grant Grant EP/P006566/1.

References

[1] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data, IEEE Transactions on Knowledge and Data Engineering 29 (10) (2017) 2318–2331.

[2] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, G. Ceder, Finding nature's missing ternary oxide compounds using machine learning and density functional theory, Chemistry of Materials 22 (12) (2010) 3762–3767.

[3] X. Zhao, K. Shirvan, R. K. Salko, F. Guo, On the prediction of critical heat flux using a physics-informed machine learning-aided framework, Applied Thermal Engineering 164 (2020) 114540.

[4] V. Schmidt, A. Luccioni, S. K. Mukkavilli, N. Balasooriya, K. Sankaran, J. Chayes, Y. Bengio, Visualizing the consequences of climate change using cycle-consistent adversarial networks, arXiv preprint arXiv:1905.03709.
[5] F. Schrodt, J. Kattge, H. Shan, F. Fazayeli, J. Joswig, A. Banerjee, M. Reichstein, G. Bönisch, S. Díaz, J. Dickie, et al., Bhpmf –a hierarchical bayesian approach to gap-filling and trait prediction for macroecology and functional biogeography, Global Ecology and Biogeography 24 (12) (2015) 1510–1521.
[6] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, V. Kumar, Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 558–566.
[7] L. von Rueden, S. Mayer, J. Garcke, C. Bauckhage, J. Schuecker, Informed machine learning-towards a taxonomy of explicit integration of knowledge into machine learning, arXiv preprint arXiv:1903.12394.
[8] H. Denli, N. Subrahmanya, et al., Multi-scale graphical models for spatio-temporal processes, in: Advances in Neural Information Processing Systems, 2014, pp. 316–324.
[9] K. C. Wong, L. Wang, P. Shi, Active model with orthotropic hyperelastic material for cardiac image analysis, in: International Conference on Functional Imaging and Modeling of the Heart, Springer, 2009, pp. 229–238.
[10] S. Chatterjee, K. Steinhaeuser, A. Banerjee, S. Chatterjee, A. Ganguly, Sparse group lasso: Consistency and climate applications, in: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, 2012, pp. 47–58.
[11] J. Liu, K. Wang, S. Ma, J. Huang, Accounting for linkage disequilibrium in genome-wide association studies: A penalized regression method, Statistics and its interface 6 (1) (2013) 99.
[12] A. Karpatne, W. Watkins, J. Read, V. Kumar, Physics-guided neural networks (pgnn): An application in lake temperature modeling, arXiv preprint arXiv:1710.11431.
[13] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, N. Ramakrishnan, Incorporating prior domain knowledge into deep neural networks, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 36–45.
[14] J.-L. Wu, K. Kashinath, A. Albert, D. Chirila, H. Xiao, et al., Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems, arXiv preprint arXiv:1905.06841.
[15] D. Liu, Y. Wang, Multi-fidelity physics-constrained neural network and its application in materials modeling, Journal of Mechanical Design 141 (12).
[16] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.
[17] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, Theoretical and Applied Fracture Mechanics 106 (2020) 102447.
[18] N. Geneva, N. Zabaras, Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks, Journal of Computational Physics 403 (2020) 109056.
[19] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81.
[20] S. Chakraborty, P. P. Chattopadhyay, S. K. Ghosh, S. Datta, Incorporation of prior knowledge in neural network model for continuous cooling of steel using genetic algorithm, Applied Soft Computing 58 (2017) 297–306.
[21] A. D. Jagtap, K. Kawaguchi, G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics 404 (2020) 109136.
[22] J. Chen, Y. Liu, Probabilistic physics-guided machine learning for fatigue data analysis, Expert Systems with Applications 168 (2021) 114316.
[23] S. Roychowdhury, M. Diligenti, M. Gori, Regularizing deep networks with prior knowledge: A constraint-based approach, Knowledge-Based Systems 222 (2021) 106989.
[24] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media, 2009.
[25] T. de Wolff, H. Carrillo, L. Martí, N. Sanchez-Pi, Towards optimally weighted physics-informed neural networks in ocean modelling, arXiv preprint arXiv:2106.08747.
[26] T. Dash, S. Chitlangia, A. Ahuja, A. Srinivasan, Incorporating domain knowledge into deep neural networks, arXiv preprint arXiv:2103.00180.
[27] R. Wang, R. Yu, Physics-guided deep learning for dynamical systems: A survey, arXiv preprint arXiv:2107.01272.
[28] J. Willard, X. Jia, S. Xu, M. Steinbach, V. Kumar, Integrating scientific knowledge with machine learning for engineering and environmental systems (2021). arXiv:2003.04919.
[29] I. Nabney, Netlab Algorithms for Pattern Recognition, Springer, 2004.
[30] R. E. Schumacker, Using R with multivariate statistics, Sage Publications, 2015.
[31] J. P. Stevens, Applied multivariate statistics for the social sciences, Routledge, 2012.
[32] Q. Wang, J. Li, M. Gouge, A. R. Nassar, P. P. Michaleris, E. W. Reutzel, Physics-based multivariable modeling and feedback linearization control of melt-pool geometry and temperature in directed energy deposition, Journal of Manufacturing Science and Engineering 139 (2) (2017) 021013.

Table A.10: The benchmark functions and evaluation ranges tested in this work.

Benchmark Function	Function $y(x_1, x_2)$	x_1 range	x_2 range
McCormick	$\sin x_1 + x_2 + (x_1 - x_2)^2 - 1.50x_1 + 2.50x_2 + 1.00$	$[-1.50, 4.00]$	$[-3.00, 4.00]$
Griewank	${x_1}^2/4000.00 + {x_2}^2/4000.00 - \cos x_1 \cos\left(x_2/\sqrt{2.00}\right) + 1.00$	$[-5.00, 5.00]$	$[-5.00, 5.00]$
Ackley	$-20\exp\left[-0.2\sqrt{0.5(x_1^2 + x_1^2)}\right]$ $-\exp\left[0.5(\cos(2\pi x_1) + \cos(2\pi x_2)\right] + \exp(1) + 20$	$[-5.00, 5.00]$	$[-5.00, 5.00]$
Branin	$\left(x_2 - (5.10/(4.00\pi^2)){x_1}^2 + (5.00/\pi)x_1 - 6.00\right) +$ $10.00(1 - (1/(8.00\pi)))\cos(x_1) + 10$	$[-5.00, 10.00]$	$[0.00, 15.00]$
Styblinski–Tang	$\left({x_1}^4 - 16.00{x_1}^2 + 5.00x_1 + {x_2}^4 - 16.00{x_2}^2 + 5.00x_2\right)/2$	$[-5.00, 5.00]$	$[-5.00, 5.00]$
Booth	$(x_1 + 2.00x_2 - 7.00)^2 + (2.00x_1 + x_2 - 5.00)^2$	$[-10.00, 10.00]$	$[-10.00, 10.00]$
Three-hump Camel	$2.00{x_1}^2 - 1.05{x_1}^4 + x^6/6.00 + x_1x_2 + {x_2}^2$	$[-5.00, 5.00]$	$[-5.00, 5.00]$
Bukin N.6	$100.00\sqrt{\left\lvert x_2 - 0.01{x_1}^2\right\rvert} + 0.01\lvert x_1 + 10.00\rvert$	$[-15.00, -5.00]$	$[-3.00, 3.00]$
Himmelblau's	$\left({x_1}^2 + x_2 - 11.00\right)^2 + \left(x_1 + {x_2}^2 - 7.00\right)^2$	$[-5.00, 5.00]$	$[-5.00, 5.00]$
Zakharov	${x_1}^2 + {x_2}^2 + (0.50x_1 + x_2)^2 + (0.50x_1 + x_2)^4$	$[-5.00, 10.00]$	$[-5.00, 10.00]$
Bohachevsky	$x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	$[-100.00, 100.00]$	$[-100.00, 100.00]$
Rosenbrock	$100.00\left(x_2 - {x_1}^2\right)^2 + (1.00 - x_1)^2$	$[-5.00, 10.00]$	$[-5.00, 10.00]$
Beale	$(1.50 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1{x_2}^2\right)^2 + \left(2.625 - x_1 + x_1{x_2}^3\right)^2$	$[-4.50, 4.50]$	$[-4.50, 4.50]$
Goldstein–Price	$\left[1.00 + (x_1 + x_2 + 1.00)^2(19.00 - 14.00x_1 + 3.00{x_1}^2 - 14.00x_2\right.$ $\left.+6.00x_1x_2 + 3.00{x_2}^2)\right] \times \left[30.00 + (2.00x_1 - 3.00x_2)^2 + (18.00\right.$ $\left.-32.00x_1 + 12.00{x_1}^2 + 48.00x_2 - 36.00x_1x_2 + 27.00{x_2}^2)\right]$	$[-2.00, 2.00]$	$[-2.00, 2.00]$

Appendix A. Benchmark functions

The 14 benchmark functions studied in this work are presented in Table Appendix A.

Table B.11: Independent two-sample multivariate Hotelling's of the empirical and prior-consistency RMSEs for the 14 benchmark functions comparing the proposed methods to the other tested methods using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively (24 samples per model variant since we test 24 random weight initialization).

Benchmark	[NN,PGNN−100]				[NN,PGNN$_1$]				[NN,PGNN$_2$]			
Function	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
McCormick	0.46	0.71	4.97	0.16	9.32	0.00	0.20	0.90	1.76	0.15	4.66	0.11
Griewank	0.90	0.44	0.73	0.70	6.10	0.00	7.90	0.02	4.51	0.00	7.91	0.02
Ackley	1.05	0.37	0.46	0.80	0.43	0.73	8.23	0.02	0.27	0.85	41.95	0.00
Branin	0.43	0.73	0.00	1.00	0.36	0.78	1.43	0.50	0.68	0.56	6.91	0.04
Styblinski–Tang	0.30	0.82	0.64	0.73	0.61	0.61	0.14	0.93	0.66	0.58	2.51	0.30
Booth	0.96	0.41	2.29	0.34	1.09	0.35	3.33	0.21	2.05	0.10	2.23	0.35
Three-hump Camel	2.20	0.09	1.77	0.43	3.89	0.01	5.23	0.07	3.58	0.01	7.29	0.03
Bukin N.6	1.35	0.26	0.96	0.63	2.34	0.07	1.78	0.43	0.97	0.41	0.26	0.88
Himmelblau's	0.41	0.75	0.16	0.92	13.81	0.00	4.71	0.09	5.55	0.00	3.20	0.20
Zakharov	0.34	0.79	0.36	0.84	5.45	0.00	3.39	0.18	3.30	0.02	3.90	0.14
Bohachevsky	0.67	0.57	0.80	0.68	0.54	0.65	6.37	0.05	0.73	0.53	8.67	0.02
Rosenbrock	0.11	0.95	0.15	0.93	3.51	0.01	4.65	0.10	0.60	0.62	1.48	0.49
Beale	1.18	0.32	3.55	0.19	2.07	0.10	2.43	0.31	2.45	0.06	4.59	0.12
Goldstein–Price	0.91	0.44	1.60	0.46	3.27	0.02	11.38	0.00	3.36	0.02	11.62	0.00

Table B.12: Independent two-sample multivariate Hotelling's of the empirical and prior-consistency RMSEs for the 14 benchmark functions comparing the proposed methods to the other tested methods using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively (24 samples per model variant since we test 24 random weight initialization).

Benchmark	[PGNN−100,PGNN$_1$]				[PGNN−100,PGNN$_2$]				[PGNN$_1$,PGNN$_2$]			
Function	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
McCormick	7.73	0.00	3.89	0.14	2.65	0.05*	14.40	0.00	11.12	0.00	1.63	0.44
Griewank	4.79	0.00	12.34	0.00	4.83	0.00	12.67	0.00	0.89	0.45	0.85	0.66
Ackley	0.35	0.79	4.36	0.13	0.93	0.42	35.87	0.00	0.70	0.55	14.93	0.00
Branin	1.13	0.36	1.36	0.52	1.70	0.16	7.56	0.03	0.77	0.51	4.51	0.12
Styblinski–Tang	0.16	0.92	0.96	0.63	0.90	0.45	0.78	0.69	1.75	0.16	3.12	0.23
Booth	0.49	0.69	9.86	0.01	0.50	0.68	8.86	0.02	0.27	0.85	0.14	0.93
Three-hump Camel	1.87	0.13	3.22	0.22	2.04	0.11	1.88	0.41	0.12	0.99	1.81	0.42
Bukin N.6	1.62	0.18	2.41	0.32	0.74	0.53	1.43	0.50	1.93	0.12	0.61	0.74
Himmelblau's	12.80	0.00	8.52	0.01	5.30	0.00	6.11	0.05*	3.23	0.02	2.11	0.35
Zakharov	3.73	0.01	4.32	0.12	1.99	0.11	5.61	0.08	0.90	0.44	1.90	0.40
Bohachevsky	1.29	0.28	8.09	0.03	2.04	0.11	9.70	0.01	0.14	0.94	0.83	0.67
Rosenbrock	3.52	0.01	6.52	0.04	0.71	0.55	2.31	0.33	1.55	0.20	0.95	0.63
Beale	0.70	0.55	1.42	0.50	1.31	0.27	3.63	0.18	0.13	0.95	0.60	0.75
Goldstein–Price	3.42	0.02	6.11	0.05*	3.31	0.02	6.04	0.05*	0.10	0.96	0.08	0.96

* This value is less than 0.05, but has been rounded to two decimal places for consistency.

Appendix B. Benchmark functions statistical tests

Tables B.11 and B.12 present the independent two-sample multivariate Hotelling's test on the percentage change in the empirical and prior-consistency RMSEs between the proposed methods and the reference models (Tables 3 and 4). Similarly, Tables B.13, B.14, B.15, and B.16 present the two-sample F-test for equal variance results for the percentage change in the variance of the empirical and prior-consistency RMSEs between the proposed methods and the reference models (Tables 5 and 6).

Table B.13: Two-sample F-test for equal variances of the Empirical RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark	[NN−100,PGNN−100]		[NN−100,PGNN$_1$]		[NN−100,PGNN$_2$]	
Function	F	P	F	P	F	P
McCormick	0.68	0.37	0.10	0.00	1.26	0.58
Griewank	1.07	0.88	3.78	0.00	3.34	0.01
Ackley	0.55	0.15	0.79	0.57	0.91	0.81
Branin	0.71	0.42	0.93	0.87	0.95	0.89
Styblinski–Tang	1.17	0.71	1.19	0.67	1.00	0.99
Booth	0.76	0.52	1.24	0.61	1.03	0.95
Three-hump Camel	2.18	0.07	4.35	0.00	4.12	0.00
Bukin N.6	0.49	0.09	0.77	0.54	0.76	0.52
Himmelblau's	1.03	0.94	20.84	0.00	6.11	0.00
Zakharov	1.54	0.31	4.14	0.00	2.85	0.02
Bohachevsky	0.81	0.61	1.53	0.31	1.65	0.24
Rosenbrock	1.03	0.94	4.17	0.00	1.68	0.22
Beale	0.76	0.52	1.20	0.67	1.55	0.30
Goldstein–Price	1.20	0.66	2.41	0.04	2.92	0.01

Table B.14: Two-sample F-test for equal variances of the empirical RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark	[PGNN−100,PGNN$_1$]		[PGNN−100,PGNN$_2$]		[PGNN$_1$,PGNN$_2$]	
Function	F	P	F	P	F	P
McCormick	0.16	0.00	2.47	0.03	15.18	0.00
Griewank	2.20	0.06	1.66	0.23	0.76	0.51
Ackley	0.91	0.83	1.46	0.37	1.60	0.27
Branin	0.89	0.79	1.12	0.78	1.26	0.58
Styblinski–Tang	0.97	0.94	1.20	0.67	1.24	0.62
Booth	1.35	0.48	1.64	0.24	1.22	0.64
Three-hump Camel	2.34	0.05*	2.27	0.05	0.97	0.95
Bukin N.6	2.08	0.09	1.43	0.40	0.69	0.38
Himmelblau's	5.17	0.00	3.23	0.01	0.62	0.27
Zakharov	4.12	0.00	2.83	0.02	0.69	0.37
Bohachevsky	0.95	0.89	0.85	0.70	0.90	0.80
Rosenbrock	2.22	0.06	1.77	0.18	0.80	0.59
Beale	1.45	0.38	1.81	0.16	1.25	0.60
Goldstein–Price	1.42	0.41	1.72	0.20	1.21	0.65

Table B.15: Two-sample F-test for equal variances of the boundary prior-consistency RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark	[NN−100,PGNN−100]		[NN−100,PGNN$_1$]		[NN−100,PGNN$_2$]	
Function	F	P	F	P	F	P
McCormick	0.81	0.61	0.13	0.00	2.00	0.10
Griewank	1.12	0.79	2.46	0.04	1.86	0.15
Ackley	0.98	0.97	0.90	0.80	1.44	0.39
Branin	0.82	0.64	0.73	0.46	0.92	0.85
Styblinski–Tang	0.87	0.73	0.84	0.67	1.03	0.94
Booth	1.55	0.30	2.09	0.08	2.54	0.03
Three-hump Camel	1.65	0.24	3.85	0.00	3.74	0.00
Bukin N.6	0.61	0.25	1.27	0.57	0.88	0.76
Himmelblau's	1.19	0.69	6.13	0.00	3.83	0.00
Zakharov	1.41	0.41	5.83	0.00	4.00	0.00
Bohachevsky	1.25	0.59	1.19	0.69	1.26	0.88
Rosenbrock	0.84	0.67	1.85	0.15	1.48	0.36
Beale	0.66	0.32	0.95	0.91	1.19	0.68
Goldstein–Price	1.38	0.45	1.96	0.11	2.37	0.04

Table B.16: Two-sample F-test for equal variances of the boundary prior-consistency RMSE across the model variants for the 14 benchmark functions using a training data set size of 1000 samples, a regularization weight of $1e-3$ for the NN model, and a regularization and a prior weights of $1e-5$ and $1e-6$ for PGNN models, respectively.

Benchmark	[PGNN−100,PGNN$_1$]		[PGNN−100,PGNN$_2$]		[PGNN$_1$,PGNN$_2$]	
Function	F	P	F	P	F	P
McCormick	0.14	0.00	1.85	0.15	13.29	0.00
Griewank	3.54	0.00	3.12	0.01	0.88	0.77
Ackley	1.44	0.39	1.66	0.23	1.15	0.74
Branin	1.31	0.52	1.33	0.50	1.01	0.98
Styblinski–Tang	1.02	0.96	0.86	0.72	0.84	0.68
Booth	1.63	0.25	1.35	0.48	0.83	0.65
Three-hump Camel	1.99	0.11	1.89	0.13	0.95	0.90
Bukin N.6	1.59	0.27	1.57	0.29	0.99	0.98
Himmelblau's	20.14	0.00	5.90	0.00	0.29	0.00
Zakharov	2.69	0.02	1.85	0.15	0.69	0.38
Bohachevsky	1.90	0.13	2.05	0.09	1.08	0.86
Rosenbrock	4.04	0.00	1.62	0.25	0.40	0.03
Beale	1.58	0.28	2.04	0.09	1.29	0.55
Goldstein–Price	2.00	0.10	2.43	0.04	1.22	0.64

Table C.17: Empirical and prior-consistency RMSE data set study for the Bohachevsky case study using using a regularization weight of $1e-5$ and a prior weight of $1e-6$.

Model Type	Prior	Scenario 1		Scenario 2		Scenario 3	
	Weight	Empirical Error	Prior_1	Empirical	Prior_2	Empirical	Prior_3
200	NN−100	66.02 ± 14.75	174.86 ± 39.55	66.02 ± 14.75	66.35 ± 41.32	66.02 ± 14.75	81.64 ± 21.72
	PGNN−30	118.07 ± 24.08	264.62 ± 74.78	118.08 ± 25.20	76.82 ± 70.93	118.58 ± 25.19	141.93 ± 31.46
	PGNN−65	76.58 ± 13.72	191.64 ± 37.77	76.11 ± 13.05	63.31 ± 40.69	76.61 ± 13.52	96.33 ± 20.01
	PGNN−100	65.24 ± 13.26	173.57 ± 33.37	65.24 ± 13.26	70.79 ± 48.33	65.24 ± 13.26	80.64 ± 20.27
	PGNN_1	62.97 ± 11.62	167.72 ± 29.03	60.13 ± 8.43	68.43 ± 40.33	62.02 ± 10.26	76.08 ± 14.98
	PGNN_2	60.66 ± 9.76	159.63 ± 29.69	64.82 ± 12.68	57.92 ± 44.83	59.71 ± 8.37	72.63 ± 12.00
600	NN−100	25.95 ± 4.36	55.03 ± 10.03	25.95 ± 4.36	26.54 ± 24.69	25.95 ± 4.36	32.68 ± 5.39
	PGNN−30	67.25 ± 14.93	124.81 ± 21.03	67.24 ± 14.94	62.38 ± 43.80	67.27 ± 14.93	67.75 ± 18.84
	PGNN−65	35.23 ± 6.79	72.22 ± 18.84	35.32 ± 7.11	40.20 ± 24.10	35.37 ± 7.15	41.24 ± 7.71
	PGNN−100	26.60 ± 4.86	55.44 ± 11.73	26.60 ± 4.86	26.02 ± 20.74	33.30 ± 6.46	33.30 ± 6.46
	PGNN_1	22.36 ± 2.78	46.42 ± 7.80	22.27 ± 3.21	31.36 ± 22.68	22.59 ± 2.92	28.03 ± 3.88
	PGNN_2	22.81 ± 3.08	45.91 ± 6.76	24.44 ± 4.24	19.03 ± 13.58	22.50 ± 3.20	27.60 ± 4.37
1000	NN−100	21.11 ± 3.67	40.42 ± 8.17	21.11 ± 3.67	23.78 ± 16.67	21.11 ± 3.67	27.86 ± 5.55
	PGNN−30	61.78 ± 15.65	111.01 ± 29.81	61.79 ± 15.67	63.19 ± 44.10	61.79 ± 15.67	63.72 ± 24.58
	PGNN−65	27.83 ± 4.40	50.57 ± 10.19	27.83 ± 4.40	28.12 ± 19.00	27.83 ± 4.40	34.48 ± 6.80
	PGNN−100	21.74 ± 4.08	40.57 ± 7.29	21.74 ± 4.08	23.56 ± 17.01	21.74 ± 4.08	28.35 ± 6.19
	PGNN_1	19.06 ± 2.96	34.73 ± 7.50	19.06 ± 2.96	26.88 ± 17.88	19.06 ± 2.96	24.52 ± 4.42
	PGNN_2	19.21 ± 2.85	33.73 ± 7.91	20.95 ± 3.89	27.32 ± 15.56	19.44 ± 3.00	24.58 ± 4.43
2000	NN−100	19.33 ± 3.53	33.42 ± 7.01	19.33 ± 3.53	28.73 ± 17.92	19.33 ± 3.53	24.82 ± 4.71
	PGNN−30	59.33 ± 18.24	100.96 ± 31.15	59.33 ± 18.24	83.02 ± 42.94	59.33 ± 18.24	58.11 ± 25.85
	PGNN−65	25.30 ± 4.84	43.19 ± 7.77	25.30 ± 4.84	29.70 ± 15.25	25.30 ± 4.84	30.30 ± 7.85
	PGNN−100	19.39 ± 3.74	33.88 ± 8.74	19.39 ± 3.74	28.98 ± 16.24	19.39 ± 3.74	25.24 ± 5.16
	PGNN_1	16.16 ± 2.27	26.66 ± 5.36	16.16 ± 2.27	23.20 ± 14.65	16.16 ± 2.27	19.95 ± 3.49
	PGNN_2	16.27 ± 2.34	26.10 ± 5.31	17.97 ± 2.71	17.81 ± 8.10	15.94 ± 1.68	19.63 ± 2.76
4000	NN−100	19.18 ± 3.32	33.35 ± 6.52	19.18 ± 3.32	23.69 ± 11.50	19.18 ± 3.32	24.36 ± 4.41
	PGNN−30	61.54 ± 14.82	104.46 ± 27.65	61.54 ± 14.82	78.11 ± 48.07	61.54 ± 14.82	58.15 ± 21.03
	PGNN−65	24.54 ± 4.83	43.61 ± 8.11	24.54 ± 4.83	27.28 ± 14.90	24.54 ± 4.83	28.41 ± 8.29
	PGNN−100	18.70 ± 3.37	33.31 ± 6.85	18.70 ± 3.37	27.32 ± 16.09	18.70 ± 3.37	23.75 ± 4.19
	PGNN_1	16.29 ± 2.31	27.38 ± 4.21	16.29 ± 2.31	22.87 ± 10.57	16.29 ± 2.31	20.00 ± 3.34
	PGNN_2	16.14 ± 2.16	26.81 ± 4.07	18.23 ± 2.57	17.44 ± 9.63	16.55 ± 2.54	19.99 ± 3.50

Appendix C. Data set size sensitivity MANOVA study

Tables C.17 and C.18 present the empirical and prior-consistency RMSEs for the Bohachevsky and the Ackley functions with different data set sizes. Tables C.19 and C.18 present the one-way Multivariate Analysis of Variance (MANOVA) results for the Bohachevsky and Ackley case studies, respectively. Figures C.1a, C.2a, C.3a, C.4a, C.5a, and C.6a demonstrate the canonical analysis of the Bohachevsky case. Figures C.7a, C.8a, C.9a, C.10a, C.11a, and C.12a demonstrate the canonical analysis of the Ackley case. Figures C.1b, C.2b, C.3b, C.4b, C.5b, and C.6b demonstrate the Mahalanobis distance matrix for the Bohachevsky case. Figures C.7b, C.8b, C.9b, C.10b, C.11b, and C.12b demonstrate the Mahalanobis distance matrix of the Ackley case.

Table C.18: Empirical and prior-consistency RMSE data set study for the Ackley case study using using a regularization weight of $1e-5$ and a prior weight of $1e-6$.

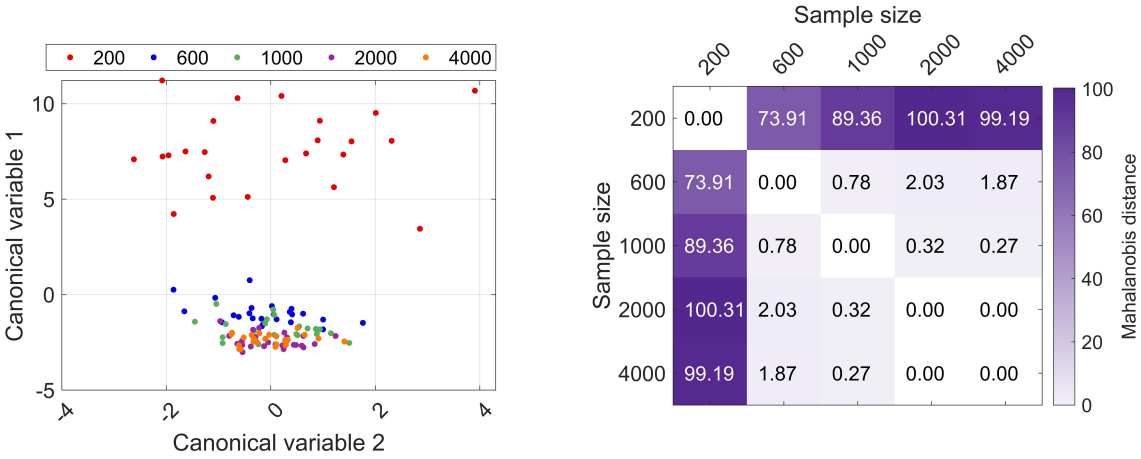
Model Type	Prior	Scenario 1		Scenario 2		Scenario 3	
	Weight	Empirical Error	Prior_1	Empirical	Prior_2	Empirical	Prior_3
200	NN−100	0.65 ± 0.01	1.22 ± 0.04	0.65 ± 0.01	2.93 ± 0.31	0.65 ± 0.01	0.27 ± 0.03
	PGNN−30	0.64 ± 0.01	1.17 ± 0.04	0.64 ± 0.01	3.33 ± 0.22	0.64 ± 0.01	0.26 ± 0.03
	PGNN−65	0.64 ± 0.01	1.18 ± 0.04	0.64 ± 0.01	3.11 ± 0.25	0.64 ± 0.01	0.28 ± 0.03
	PGNN−100	0.65 ± 0.01	1.21 ± 0.04	0.65 ± 0.01	2.91 ± 0.33	0.65 ± 0.01	0.27 ± 0.03
	PGNN_1	0.64 ± 0.01	1.18 ± 0.04	0.64 ± 0.01	3.12 ± 0.31	0.64 ± 0.01	0.27 ± 0.03
	PGNN_2	0.63 ± 0.01	1.13 ± 0.04	0.64 ± 0.01	2.75 ± 0.27	0.64 ± 0.01	0.26 ± 0.02
600	NN−100	0.59 ± 0.01	1.07 ± 0.03	0.59 ± 0.01	2.06 ± 0.29	0.59 ± 0.01	0.21 ± 0.02
	PGNN−30	0.60 ± 0.01	1.04 ± 0.03	0.60 ± 0.01	2.66 ± 0.24	0.60 ± 0.01	0.20 ± 0.02
	PGNN−65	0.59 ± 0.01	1.05 ± 0.04	0.59 ± 0.01	2.25 ± 0.32	0.59 ± 0.01	0.21 ± 0.02
	PGNN−100	0.59 ± 0.01	1.07 ± 0.03	0.59 ± 0.01	2.01 ± 0.29	0.59 ± 0.01	0.21 ± 0.01
	PGNN_1	0.59 ± 0.01	1.04 ± 0.03	0.59 ± 0.01	2.19 ± 0.29	0.59 ± 0.01	0.20 ± 0.01
	PGNN_2	0.59 ± 0.01	1.02 ± 0.02	0.59 ± 0.01	1.96 ± 0.24	0.59 ± 0.01	0.20 ± 0.01
1000	NN−100	0.59 ± 0.00	1.01 ± 0.02	0.59 ± 0.00	1.76 ± 0.27	0.59 ± 0.00	0.18 ± 0.02
	PGNN−30	0.59 ± 0.01	0.99 ± 0.04	0.59 ± 0.01	2.55 ± 0.24	0.59 ± 0.01	0.18 ± 0.02
	PGNN−65	0.59 ± 0.01	0.98 ± 0.02	0.59 ± 0.01	2.08 ± 0.24	0.59 ± 0.01	0.18 ± 0.02
	PGNN−100	0.59 ± 0.01	1.00 ± 0.02	0.59 ± 0.01	1.79 ± 0.31	0.59 ± 0.01	0.19 ± 0.02
	PGNN_1	0.58 ± 0.00	0.99 ± 0.02	0.58 ± 0.00	1.99 ± 0.21	0.58 ± 0.00	0.18 ± 0.01
	PGNN_2	0.58 ± 0.00	0.97 ± 0.02	0.59 ± 0.01	1.71 ± 0.27	0.58 ± 0.00	0.17 ± 0.01
2000	NN−100	0.57 ± 0.01	0.92 ± 0.05	0.57 ± 0.01	1.91 ± 0.22	0.57 ± 0.01	0.15 ± 0.02
	PGNN−30	0.59 ± 0.01	0.92 ± 0.03	0.59 ± 0.01	2.55 ± 0.20	0.59 ± 0.01	0.15 ± 0.03
	PGNN−65	0.58 ± 0.01	0.91 ± 0.03	0.58 ± 0.01	2.19 ± 0.22	0.58 ± 0.01	0.14 ± 0.02
	PGNN−100	0.57 ± 0.01	0.92 ± 0.03	0.57 ± 0.01	1.91 ± 0.28	0.57 ± 0.01	0.14 ± 0.01
	PGNN_1	0.57 ± 0.01	0.91 ± 0.05	0.57 ± 0.01	1.98 ± 0.37	0.57 ± 0.01	0.13 ± 0.01
	PGNN_2	0.57 ± 0.01	0.91 ± 0.05	0.57 ± 0.01	1.88 ± 0.32	0.57 ± 0.01	0.12 ± 0.01
4000	NN−100	0.57 ± 0.02	0.93 ± 0.06	0.57 ± 0.02	1.69 ± 0.35	0.57 ± 0.02	0.14 ± 0.03
	PGNN−30	0.58 ± 0.00	0.91 ± 0.03	0.58 ± 0.00	2.29 ± 0.18	0.58 ± 0.00	0.14 ± 0.02
	PGNN−65	0.57 ± 0.01	0.91 ± 0.03	0.57 ± 0.01	1.88 ± 0.24	0.57 ± 0.01	0.13 ± 0.02
	PGNN−100	0.57 ± 0.02	0.93 ± 0.06	0.57 ± 0.02	1.63 ± 0.33	0.57 ± 0.01	0.14 ± 0.02
	PGNN_1	0.56 ± 0.01	0.89 ± 0.02	0.56 ± 0.01	1.59 ± 0.27	0.56 ± 0.01	0.13 ± 0.02
	PGNN_2	0.56 ± 0.01	0.88 ± 0.02	0.56 ± 0.01	1.48 ± 0.24	0.56 ± 0.01	0.12 ± 0.01

Table C.19: MANOVA results across the five data sizes for the Bohachevsky case study using a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant difference between the means (critical P value 0.05).

Model Type	Scenario 1			Scenario 2			Scenario 3		
	Dimension	P_1	P_2	Dimension	P_1	P_2	Dimension	P_1	P_2
PGNN_1	1	0.00	0.90	1	0.00	0.99	1	0.00	0.35
PGNN_2	1	0.00	0.82	1	0.00	0.35	1	0.00	0.52

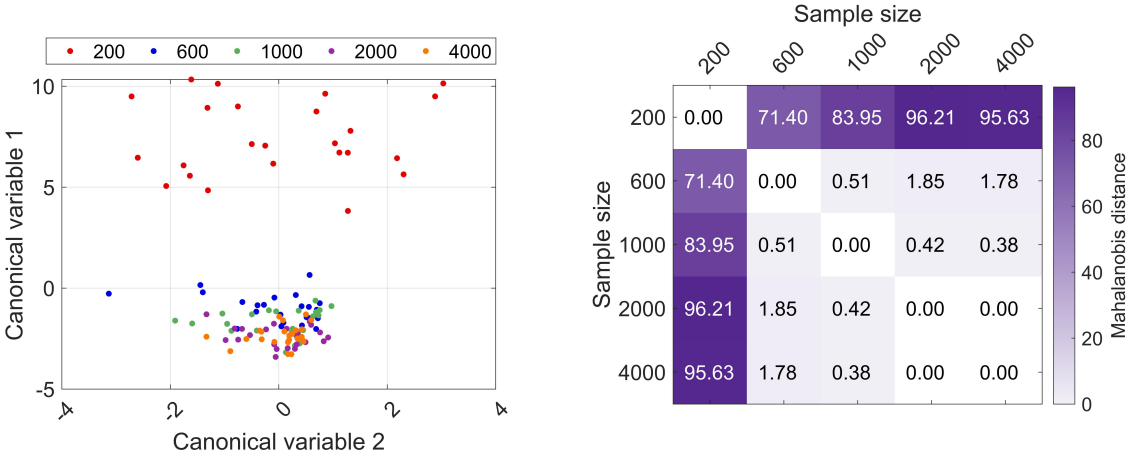
Table C.20: MANOVA results across the five data sizes for the Ackley case study using a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant difference between the means (critical P value 0.05).

Model Type	Scenario 1			Scenario 2			Scenario 3		
	Dimension	P_1	P_2	Dimension	P_1	P_2	Dimension	P_1	P_2
PGNN_1	2	0.00	0.00	2	0.00	0.00	2	0.00	0.00
PGNN_2	2	0.00	0.00	2	0.00	0.00	2	0.00	0.00



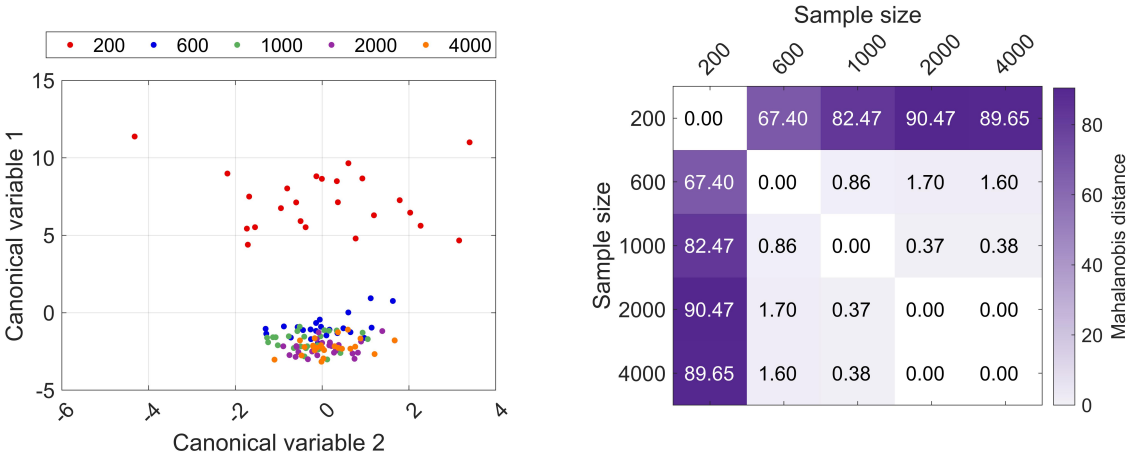
(a) (b)

Figure C.1: MANOVA study results for $PGNN_1$ using the Bohachevsky function and the boundary prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



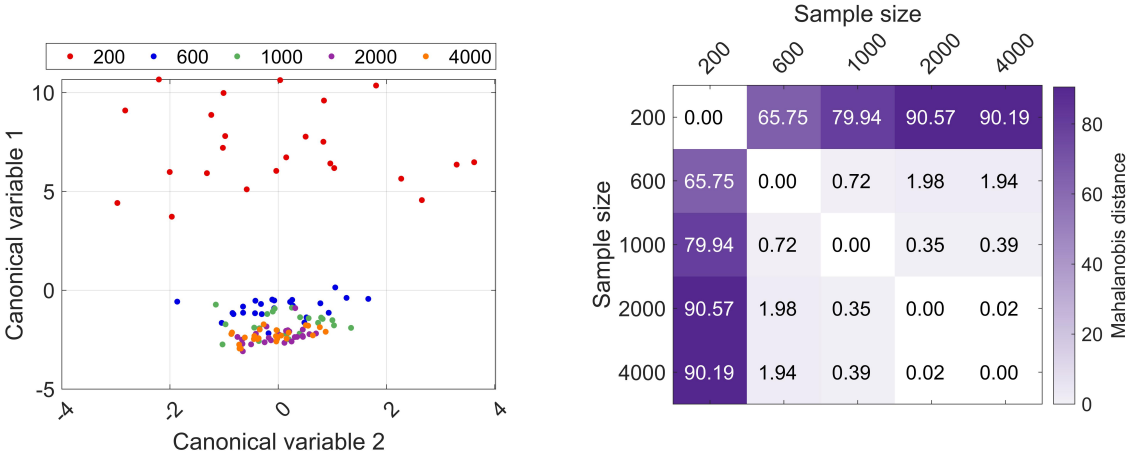
(a) (b)

Figure C.2: MANOVA study results for $PGNN_1$ using the Bohachevsky function and the initial prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



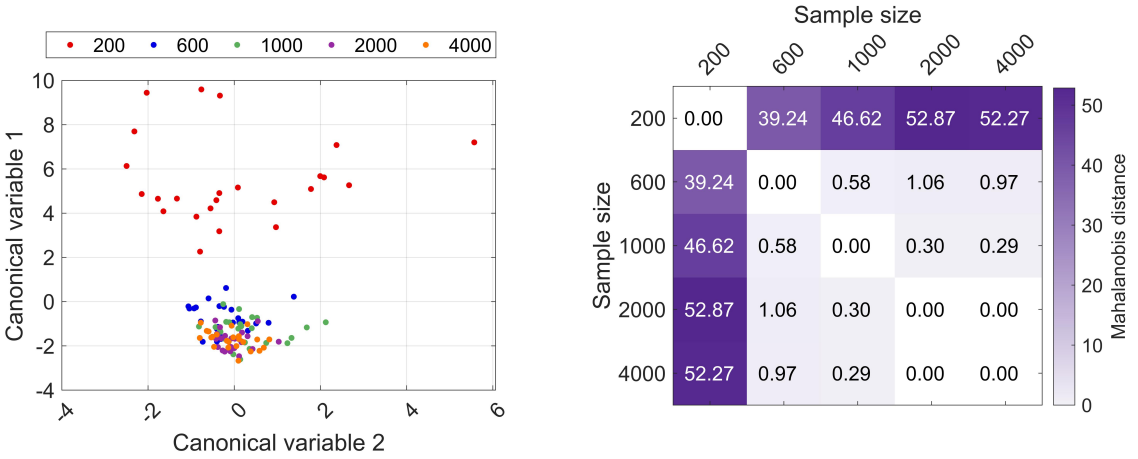
(a) (b)

Figure C.3: MANOVA study results for $PGNN_1$ using the Bohachevsky function and the symmetry prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



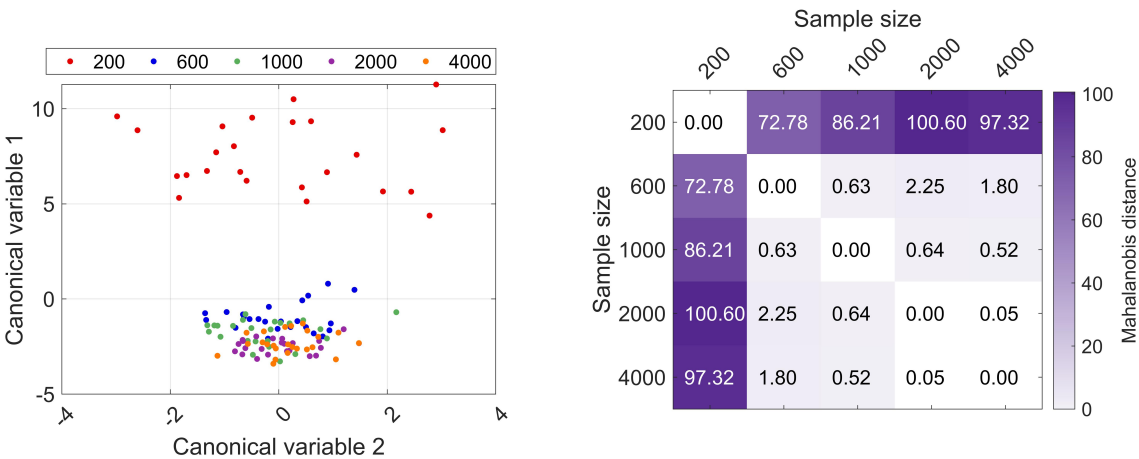
(a) (b)

Figure C.4: MANOVA study results for $PGNN_2$ using the Bohachevsky function and the boundary prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



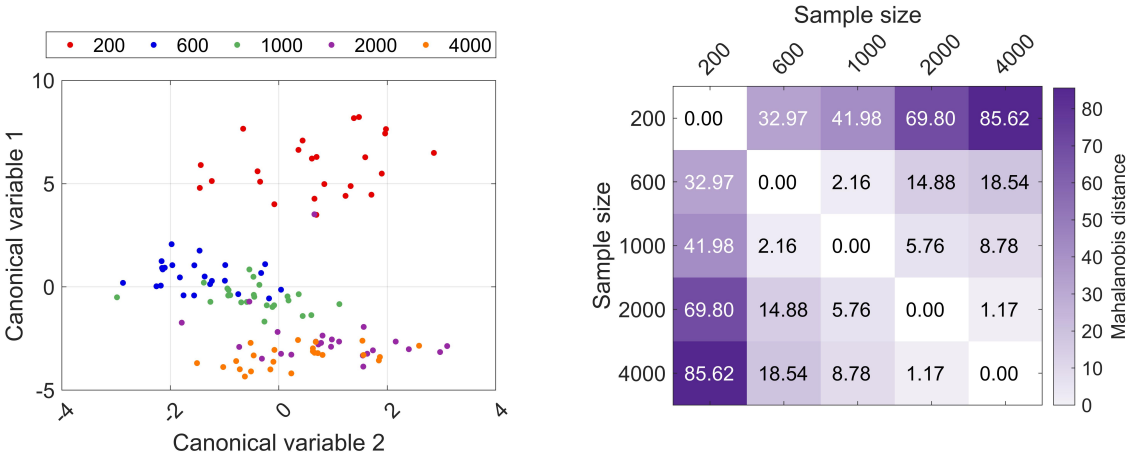
(a) (b)

Figure C.5: MANOVA study results for $PGNN_2$ using the Bohachevsky function and the initial prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



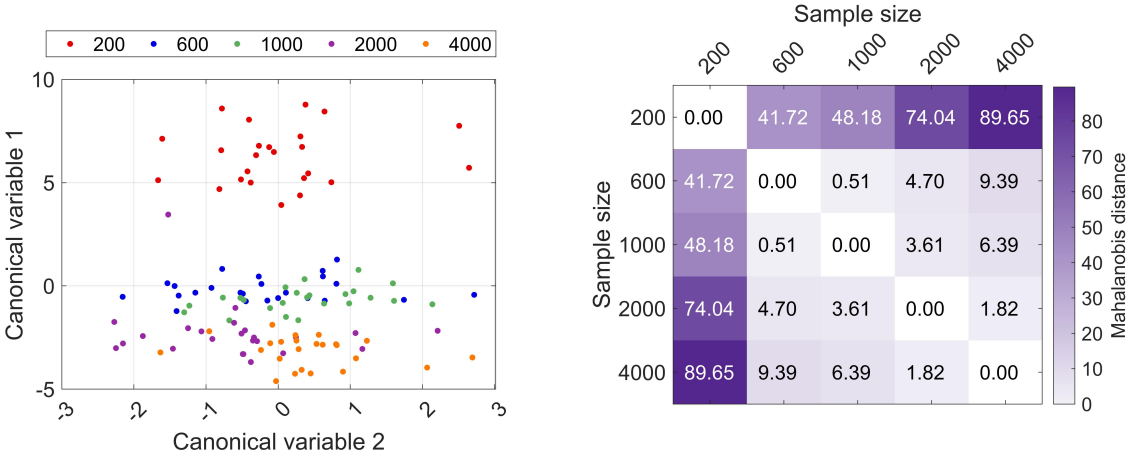
(a) (b)

Figure C.6: MANOVA study results for $PGNN_2$ using the Bohachevsky function and the symmetry prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



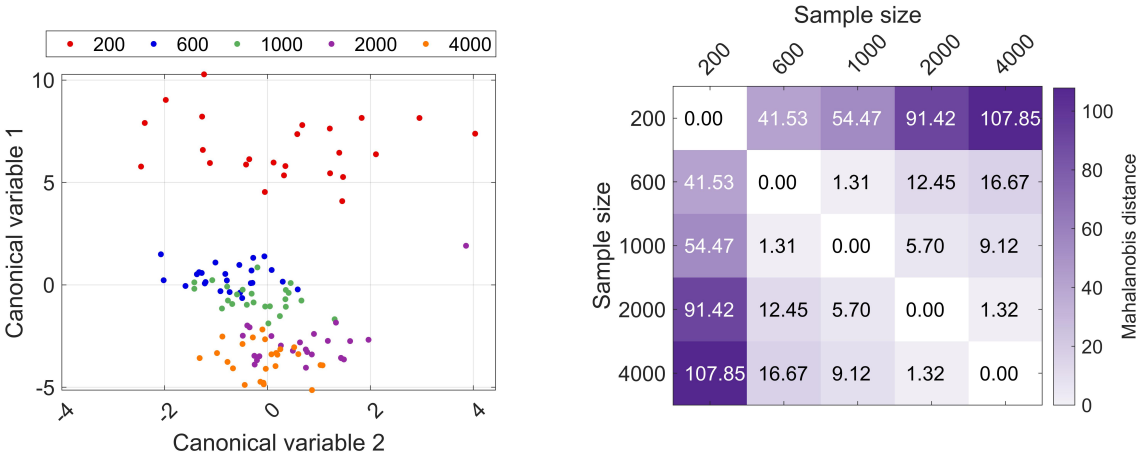
(a) (b)

Figure C.7: MANOVA study results for $PGNN_1$ using the Ackley function and the boundary prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



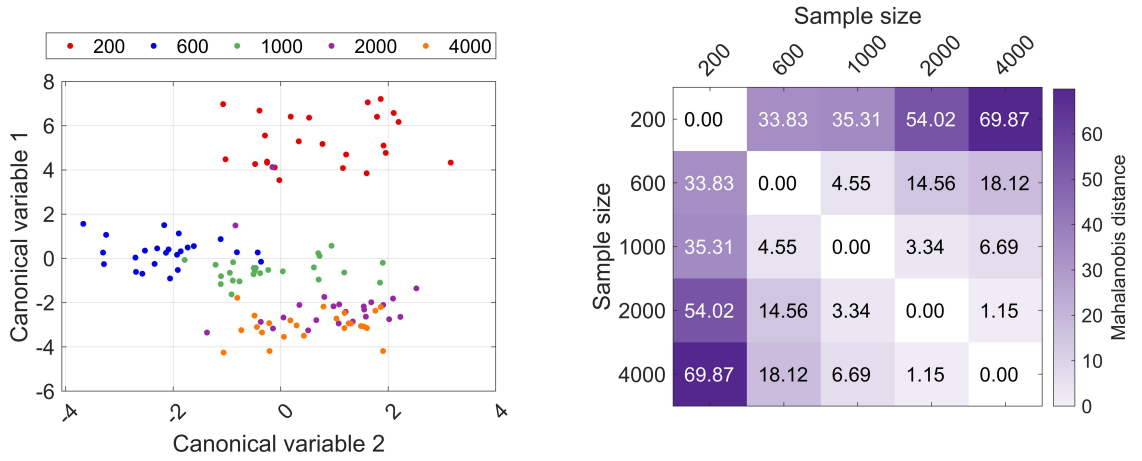
(a) (b)

Figure C.8: MANOVA study results for $PGNN_1$ using the Ackley function and the initial prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



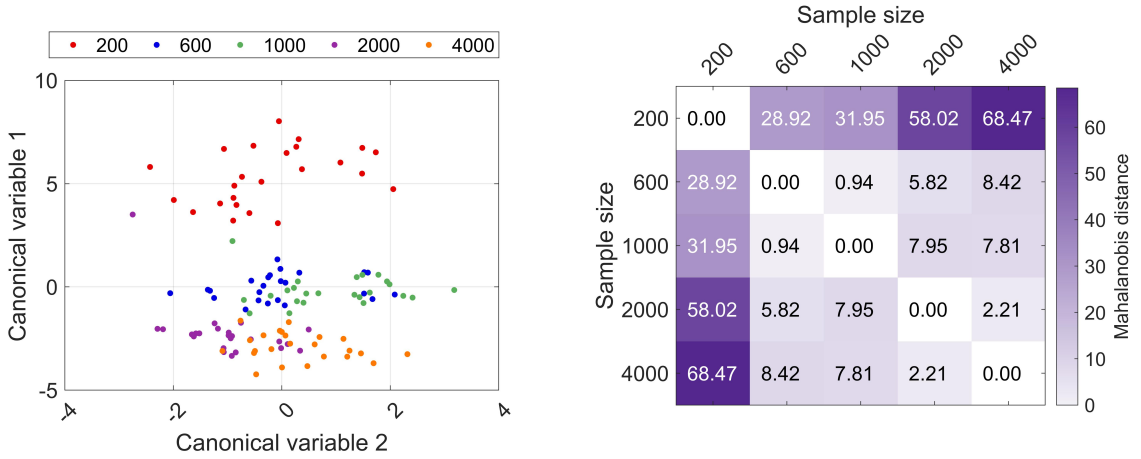
(a) (b)

Figure C.9: MANOVA study results for $PGNN_1$ using the Ackley function and the symmetry prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



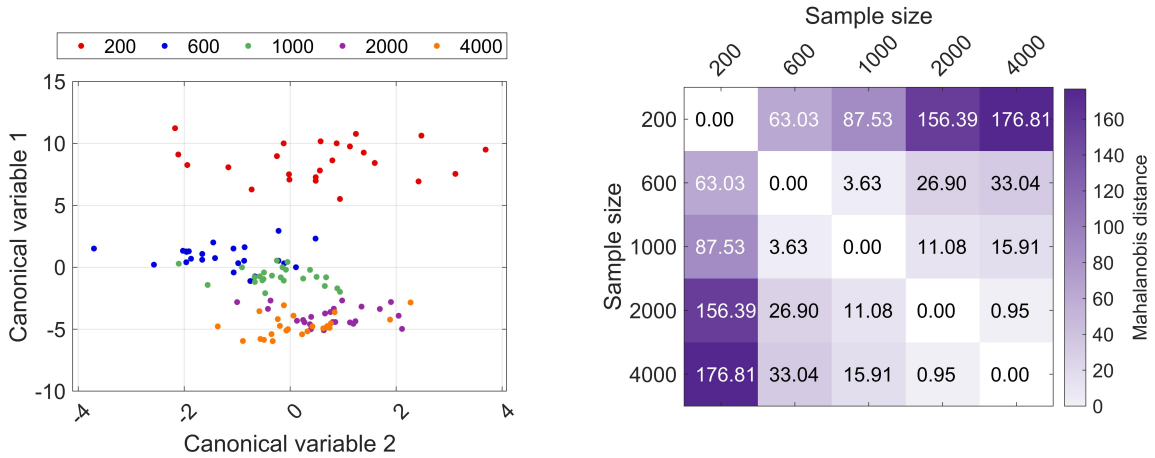
(a) (b)

Figure C.10: MANOVA study results for $PGNN_2$ using the Ackley function and the boundary prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



(a) (b)

Figure C.11: MANOVA study results for $PGNN_2$ using the Ackley function and the initial prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.



(a) (b)

Figure C.12: MANOVA study results for $PGNN_2$ using the Ackley function and the symmetry prior: (a) Canonical analysis scatter plot and (b) Mahalanobis distances between sample size means.

Table D.21: Multiple comparison test of the empirical and prior-consistency RMSEs across the five data sizes for the three prior scenarios of the Bohachevsky and Ackley case studies.The model hyperparameters for this test include a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant difference between the means (critical P value 0.05).

Model Type	Data set size compared	Empirical RMSE P values						Prior-consistency RMSE P values					
		Bohachevsky Scenario			Ackley Scenario			Bohachevsky Scenario			Ackley Scenario		
		1	2	3	1	2	3	1	2	3	1	2	3
	[200, 600]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 1000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 2000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 4000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$PGNN_1$	[600, 1000]	0.27	0.10	0.13	0.22	0.30	0.28	0.04	0.97	0.49	0.00	0.12	0.00
	[600, 2000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.00	0.09	0.00
	[600, 4000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.00	0.00	0.00	0.00
	[1000, 2000]	0.40	0.17	0.30	0.00	0.00	0.00	0.29	0.98	0.22	0.00	1.00	0.00
	[1000, 4000]	0.45	0.21	0.34	0.00	0.00	0.00	0.38	0.98	0.23	0.00	0.00	0.00
	[2000, 4000]	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00	0.21	0.00	1.00
	[200, 600]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 1000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 2000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	[200, 4000]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$PGNN_2$	[600, 1000]	0.09	0.34	0.13	0.43	0.00	0.29	0.03	0.72	0.47	0.00	0.02	0.00
	[600, 2000]	0.00	0.01	0.00	0.00	0.89	0.00	0.00	1.00	0.00	0.00	0.87	0.00
	[600, 4000]	0.00	0.01	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
	[1000, 2000]	0.25	0.50	0.06	0.00	0.00	0.00	0.36	0.60	0.06	0.00	0.20	0.00
	[1000, 4000]	0.21	0.59	0.17	0.00	0.00	0.00	0.46	0.56	0.10	0.00	0.03	0.00
	[2000, 4000]	1.00	1.00	0.99	0.00	0.04	0.01	1.00	1.00	1.00	0.12	0.00	0.99

Appendix D. Data set size ANOVA and multiple comparison test study

Table D.21 presents the multiple comparison test results using Tukey's honestly significant difference procedure on ANOVA test results (omitted but available upon request).

Table E.22: Spearman's rho correlation and P value between the empirical RMSE and the data size for the three prior scenarios of the Bohachevsky and Ackley case studies.The model hyperparameters for this test include a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant correlation (critical P value 0.05).

Model Type	Bohachevsky Case Study						Ackley Case Study					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value
$PGNN_1$	−0.83	0.00	−0.82	0.00	−0.83	0.00	−0.92	0.00	−0.92	0.00	−0.93	0.00
$PGNN_2$	−0.83	0.00	−0.76	0.00	−0.82	0.00	−0.90	0.00	−0.91	0.00	−0.92	0.00

Table E.23: Spearman's rho correlation and P value between the prior-consistency RMSE and the data size for the three prior scenarios of the Bohachevsky and Ackley case studies.The model hyperparameters for this test include a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant correlation (critical P value 0.05).

Model Type	Bohachevsky Case Study						Ackley Case Study					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value
$PGNN_1$	−0.85	0.00	−0.34	0.00	−0.82	0.00	−0.92	0.00	−0.79	0.00	−0.91	0.00
$PGNN_2$	−0.84	0.00	−0.34	0.00	−0.80	0.00	−0.92	0.00	−0.72	0.00	−0.93	0.00

Table F.24: Spearman's rho correlation and P value between the empirical RMSE and the prior-consistency RMSE for the three prior scenarios of the Bohachevsky and Ackley case studies.The model hyperparameters for this test include a regularization weight of $1e-5$ and a prior weight of $1e-6$. P-values are shaded to indicate a statistically significant correlation (critical P value 0.05).

Model Type	Bohachevsky Case Study						Ackley Case Study					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value	ρ	P value
NN	0.94	0.00	0.34	0.00	0.98	0.00	0.95	0.00	0.74	0.00	0.90	0.00
PGNN-30	0.95	0.00	0.11	0.24	0.89	0.00	0.92	0.00	0.85	0.00	0.85	0.00
PGNN-65	0.93	0.00	0.26	0.00	0.95	0.00	0.93	0.00	0.77	0.00	0.88	0.00
PGNN-100	0.96	0.00	0.36	0.00	0.98	0.00	0.94	0.00	0.75	0.00	0.89	0.00
$PGNN_1$	0.94	0.00	0.43	0.00	0.98	0.00	0.94	0.00	0.85	0.00	0.90	0.00
$PGNN_2$	0.95	0.00	0.37	0.00	0.97	0.00	0.93	0.00	0.76	0.00	0.88	0.00

Appendix E. Data set size and RMSE performance correlation study

Tables E.22 and E.23 present the strength and direction of correlation between the data set size and the empirical and prior-consistency errors using Superman's rank correlation and the results from the data set size sensitivity study results in Subsection 4.2.

Appendix F. Empirical and prior-consistency performance correlation study

Table F.24 presents the strength and direction of correlation between the empirical and prior-consistency errors using Superman's rank correlation and the results from the data set size sensitivity study results in Subsection 4.2.

Table G.25: Empirical and prior-consistency RMSE sensitivity study for the Bohachevsky case study using a 1000 sample data set a regularization weight of $1e-3$.

Model Type	Prior	Scenario 1		Scenario 2		Scenario 3	
	Weight	Empirical Error	Prior_1	Empirical	Prior_2	Empirical	Prior_3
NN−100	-	21.11 ± 3.67	40.42 ± 8.17	21.11 ± 3.67	23.78 ± 16.67	21.11 ± 3.67	27.86 ± 5.55
PGNN−30	$1e-4$	61.36 ± 15.32	111.34 ± 29.26	61.41 ± 15.41	69.33 ± 40.54	61.41 ± 15.41	61.82 ± 24.14
	$1e-6$	61.44 ± 15.49	111.52 ± 29.67	61.41 ± 15.41	69.33 ± 40.54	61.41 ± 15.41	61.82 ± 24.14
	$1e-8$	61.41 ± 15.41	111.44 ± 29.43	61.41 ± 15.41	69.33 ± 40.54	61.41 ± 15.41	61.82 ± 24.14
PGNN−65	$1e-4$	27.19 ± 3.77	50.62 ± 9.72	27.17 ± 3.78	29.95 ± 16.87	27.17 ± 3.78	33.15 ± 5.88
	$1e-6$	27.17 ± 3.78	50.42 ± 9.95	27.17 ± 3.78	29.95 ± 16.87	27.17 ± 3.78	33.15 ± 5.88
	$1e-8$	27.17 ± 3.78	50.43 ± 9.95	27.17 ± 3.78	29.95 ± 16.87	27.17 ± 3.78	33.15 ± 5.88
PGNN−100	$1e-4$	21.12 ± 3.68	40.23 ± 8.33	21.11 ± 3.67	23.78 ± 16.76	21.11 ± 3.67	27.86 ± 5.55
	$1e-6$	21.11 ± 3.67	40.42 ± 8.17	21.11 ± 3.67	23.78 ± 16.76	21.11 ± 3.67	27.86 ± 5.55
	$1e-8$	21.11 ± 3.67	40.42 ± 8.17	21.11 ± 3.67	23.78 ± 16.76	21.11 ± 3.67	27.86 ± 5.55
PGNN_1	$1e-4$	19.34 ± 2.81	35.09 ± 5.49	19.38 ± 2.98	26.06 ± 17.04	19.38 ± 2.98	25.10 ± 4.71
	$1e-6$	19.38 ± 2.98	35.08 ± 6.05	19.38 ± 2.98	26.06 ± 17.04	19.38 ± 2.98	25.10 ± 4.71
	$1e-8$	19.38 ± 2.98	35.08 ± 6.05	19.38 ± 2.98	26.06 ± 17.04	19.38 ± 2.98	25.10 ± 4.71
PGNN_2	$1e-4$	19.66 ± 2.26	35.71 ± 5.53	20.97 ± 3.67	24.07 ± 13.09	19.65 ± 2.94	25.24 ± 4.48
	$1e-6$	19.85 ± 2.50	35.66 ± 5.71	20.97 ± 3.67	24.07 ± 13.09	19.65 ± 2.94	25.24 ± 4.48
	$1e-8$	19.85 ± 2.50	35.66 ± 5.71	20.97 ± 3.67	24.07 ± 13.09	19.65 ± 2.94	25.24 ± 4.48

Table G.26: T-test of the Empirical RMSE study for the Bohachevsky case study using a 1000 sample data set a regularization weight of $1e-3$.

Model Type	Prior weights	Scenario 1				Scenario 2				Scenario 3			
	compared	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
PGNN_1	$[1e-4, 1e-6]$	0.07	0.98	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
	$[1e-4, 1e-8]$	0.07	0.98	0.01	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
	$[1e-6, 1e-8]$	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
PGNN_2	$[1e-4, 1e-6]$	0.08	0.97	0.16	0.92	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
	$[1e-4, 1e-8]$	0.08	0.97	0.16	0.92	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
	$[1e-6, 1e-8]$	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00

Appendix G. Hyperparameter sensitivity investigation

Tables G.25 and G.27 present the results of training the PGNN−30, PGNN−65, PGNN−100, PGNN_1, and PGNN_2 models with the following $[\rho_r, \rho_p]$ loss weights: $[1e-3, 1e-4]$, $[1e-3, 1e-6]$, $[1e-3, 1e-6]$, $[1e-5, 1e-4]$, and $[1e-7, 1e-4]$. The NN−100 reference model was trained with a regularization weight $\rho_r = 1e-3$. All the models were trained 24 times with randomly initialized weights for the Bohachevsky function using a training data set size of 1000.

Tables G.26 and G.28 present the Box's M and two-sample Hotelling's T^2 tests results which assess if changing the prior weight or the regularization weight have an effect on the multivariate empirical and prior-consistency RMSEs of the PGNN_1 and PGNN_2 models.

Table G.29 presents the Box's M and two-sample Hotelling's T^2 tests results comparing the proposed models to the reference models with the prior weight $1e-4$ and the three regularization weights (72 samples per model variant, 24 random weight initialization by 3 regularization weights).

Table G.30 presents the Box's M and two-sample Hotelling's T^2 tests results comparing the proposed models to the reference models with the three prior weights and the three regularization weights for each model variant (144 samples per model variant, 24 random weight initialization by 3 regularization weights and 24 random weight initialization by 3 prior weights).

Table G.27: Empirical and prior-consistency RMSE sensitivity study for the Bohachevsky case study using a 1000 sample data set a prior weight of $1e-4$.

Model Type	$\mathcal{L}_2$	Scenario 1		Scenario 2		Scenario 3	
	Weight	Empirical Error	Prior_1	Empirical	Prior_2	Empirical	Prior_3
NN−100	-	21.11 ± 3.67	40.42 ± 8.17	21.11 ± 3.67	23.78 ± 16.67	21.11 ± 3.67	27.86 ± 5.55
	$1e-3$	61.36 ± 15.32	111.34 ± 29.26	61.41 ± 15.41	69.33 ± 40.54	61.41 ± 15.41	61.82 ± 24.14
PGNN−30	$1e-5$	61.78 ± 15.67	111.04 ± 29.82	61.79 ± 15.67	63.19 ± 44.10	61.79 ± 15.67	63.71 ± 24.56
	$1e-7$	61.47 ± 16.25	111.42 ± 27.77	61.33 ± 16.30	57.06 ± 44.01	61.33 ± 16.29	62.93 ± 24.72
	$1e-3$	27.19 ± 3.77	50.62 ± 9.72	27.17 ± 3.78	29.95 ± 16.87	27.17 ± 3.78	33.15 ± 5.88
PGNN−65	$1e-5$	27.84 ± 4.40	50.99 ± 10.20	27.83 ± 4.40	28.12 ± 19.00	27.83 ± 4.40	34.48 ± 6.80
	$1e-7$	28.03 ± 4.02	51.99 ± 10.54	28.02 ± 4.01	28.36 ± 17.94	28.02 ± 4.01	34.45 ± 6.07
	$1e-3$	21.12 ± 3.68	20.23 ± 8.33	21.11 ± 3.67	23.78 ± 16.76	21.11 ± 3.67	27.86 ± 5.55
PGNN−100	$1e-5$	21.71 ± 4.10	40.56 ± 7.34	21.74 ± 4.08	23.56 ± 17.01	21.74 ± 4.08	28.35 ± 6.19
	$1e-7$	21.85 ± 3.98	41.95 ± 7.45	21.83 ± 4.00	23.82 ± 12.32	21.83 ± 4.00	28.53 ± 5.82
	$1e-3$	19.34 ± 2.81	35.09 ± 5.49	19.38 ± 2.98	26.06 ± 17.04	19.38 ± 2.98	25.10 ± 4.71
PGNN_1	$1e-5$	19.19 ± 2.96	35.10 ± 7.49	19.06 ± 2.96	26.88 ± 17.88	19.06 ± 2.96	24.52 ± 4.42
	$1e-7$	18.59 ± 2.34	33.60 ± 5.62	18.77 ± 2.58	21.16 ± 11.81	18.50 ± 2.34	23.86 ± 4.11
	$1e-3$	19.66 ± 2.26	35.71 ± 5.53	20.97 ± 3.67	24.07 ± 13.09	19.65 ± 2.94	25.24 ± 4.48
PGNN_2	$1e-5$	19.21 ± 2.65	32.88 ± 6.79	20.95 ± 3.89	27.32 ± 15.57	19.44 ± 3.00	24.58 ± 4.43
	$1e-7$	19.79 ± 3.27	36.37 ± 7.13	21.22 ± 2.89	19.09 ± 13.38	18.70 ± 2.35	23.52 ± 3.95

Table G.28: T-test of the Empirical RMSE sensitivity study for the Bohachevsky case study using a 1000 sample data set a prior weight of $1e-4$.

Model Type	Reg. weights	Scenario 1				Scenario 2				Scenario 3			
	compared	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
	$[1e-3, 1e-5]$	0.92	0.43	0.07	0.97	0.03	0.99	0.19	0.91	0.46	0.71	0.22	0.90
PGNN_1	$[1e-3, 1e-7]$	0.43	0.73	1.10	0.59	1.16	0.32	1.71	0.44	0.82	0.49	1.40	0.51
	$[1e-5, 1e-7]$	0.94	0.42	0.71	0.71	1.38	0.25	1.77	0.43	1.21	0.30	0.69	0.71
	$[1e-3, 1e-5]$	0.44	0.72	2.84	0.26	0.40	0.75	0.62	0.74	3.88	0.01	0.67	0.72
PGNN_2	$[1e-3, 1e-7]$	1.01	0.39	0.16	0.92	0.42	0.74	1.75	0.43	2.99	0.03	2.05	0.39
	$[1e-5, 1e-7]$	0.59	0.62	3.97	0.16	0.85	0.46	4.09	0.15	0.71	0.55	0.90	0.65

Table G.29: T-test for the Bohachevsky case study using a 1000 sample data set with a prior weight of $1e-4$ and the results from the three regularization weights for each model variant (72 samples per model variant (24 random weight initialization by 3 regularization weights).

Models	Scenario 1				Scenario 2				Scenario 3			
compared	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
[PGNN−30,PGNN_1]	66.58	0.00	531.41	0.00	68.96	0.00	537.92	0.00	111.82	0.00	875.43	0.00
[PGNN−30,PGNN_2]	64.11	0.00	520.08	0.00	63.45	0.00	488.84	0.00	101.06	0.00	847.02	0.00
[PGNN−65,PGNN_1]	6.22	0.00	228.95	0.00	3.65	0.01	221.12	0.00	20.65	0.00	262.65	0.00
[PGNN−65,PGNN_2]	5.02	0.00	200.96	0.00	1.68	0.17	115.86	0.00	12.84	0.00	229.74	0.00
[PGNN−100,PGNN_1]	3.03	0.03	29.65	0.00	2.37	0.07	20.68	0.00	2.99	0.03	21.17	0.00
[PGNN−100,PGNN_2]	3.06	0.03	25.13	0.00	0.59	0.62	0.70	0.70	3.37	0.02	20.37	0.00
[PGNN_1,PGNN_2]	0.12	0.95	1.73	0.42	1.37	0.25	14.80	0.00	1.62	0.18	3.79	0.15

* This value is less than 0.05, but has been rounded to two decimal places for consistency.

Table G.30: T-test for the Bohachevsky case study using a 1000 sample data set for the results form three prior weights and the results from the three regularization weights for each model variant (144 samples per model variant, 24 random weight initialization by 3 regularization weights and 24 random weight initialization by 3 prior weights).

Models	Scenario 1				Scenario 2				Scenario 3			
compared	F	P	T^2	P	F	P	T^2	P	F	P	T^2	P
[PGNN−30,$PGNN_1$]	131.76	0.00	1069.33	0.00	132.29	0.00	108.37	0.00	225.18	0.00	1898.06	0.00
[PGNN−30,$PGNN_2$]	134.94	0.00	1052.08	0.00	126.66	0.00	995.91	0.00	216.50	0.00	1850.09	0.00
[PGNN−65,$PGNN_1$]	11.68	0.00	428.14	0.00	5.35	0.00	416.01	0.00	48.90	0.00	483.94	0.00
[PGNN−65,$PGNN_2$]	11.70	0.00	402.15	0.00	40.95	0.00	433.49	0.00	40.95	0.00	433.49	0.00
[PGNN−100,$PGNN_1$]	4.63	0.00	49.71	0.00	3.32	0.02	32.57	0.00	5.52	0.00	30.53	0.00
[PGNN−100,$PGNN_2$]	6.82	0.00	42.27	0.00	2.20	0.09	0.59	0.74	3.89	0.01	31.37	0.00
[$PGNN_1$,$PGNN_2$]	0.63	0.60	2.49	0.29	4.34	0.00	23.90	0.00	0.90	0.44	6.48	0.04

Chapter 3

Paper Two (methods): Prior-knowledge-guided Neural Networks: Analysis of competing prior objectives and new correlation-based weighting

3.1 Paper Two Summary

Paper two investigates the effects of competing prior objectives in prior-knowledge-guided neural networks and proposes new correlation-based prior weighting metrics. The methods presented in the paper are tested on a regression benchmark problem using four prior objectives separately and combined. The proposed methods are assessed through empirical root-mean-square-error (RMSE) accuracy and prior-knowledge RMSE consistency. Below is a summary of the paper's contributions.

- **For the first time in the literature, an analysis and comparison of the effects of the prior loss weights on the PGNN performance in the case of single versus multi-prior modeling problems.**
  - We conclude that for modeling problems with single or multiple priors, if the relationship between the prior errors and the empirical error is nonlinear or unknown, all the priors should be equally weighted in the validation performance index. Equally weighting priors that have an unknown relationship to the empirical error, results in selecting the most generalizable model from the candidate models in terms of prior-consistency and empirical performance.

  - For modeling problems with multiple priors, the empirical/prior-consistency performance is affected more significantly by the prior loss weights than the prior error weights in the validation performance index (PI). We hypothesize and empirically validate that the prior loss weights significantly impact the model performance relative to the validation error weights because the loss search space is larger than the validation PI search space.

- **Correlation-based prior loss adaptive weighting:** A new prior-guided neural network computational framework that includes for the first time a scalable multi-prior correlation-based adaptive weighting algorithm to balance the interplay between the competing prior and empirical objectives in the loss function.

- **Weighted correlation-based model validation:** For the first time in literature, we propose to adaptively weigh the prior errors in the validation performance index of prior-knowledge-guided neural networks. We propose a scalable multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step to balance the interplay between the different error terms in the cross-validation error function.

  - In most applications, without further studies, the type of relation the empirical and prior-consistency loss/validation error variables have is often unknown. The relation between the empirical and prior-consistency errors is often unknown due to the lack of knowledge on the relevance of a given prior towards the empirical error. The weighting metrics proposed in this work utilize Pearson's correlation coefficient to assess the strength and direction of the linear relationships between the empirical and prior-consistency error variables.

  - The proposed metrics aim to balance the interplay between the competing empirical and prior errors in the loss function and the validation performance index function. The proposed weighting metrics are not limited to PGNN discussed in this work and apply to data-driven modeling frameworks with competing loss and validation errors.

- For modeling problems with multiple priors, the proposed correlation-based adaptive prior loss weighting metric, along with the equally weighted sum of the empirical and prior errors as the cross-validation performance index (CoReg-PGNN$_3$), achieve the best empirical and prior-consistency RMSE improvements.
- For modeling problems with a single prior and a lack of knowledge of the value of said prior, the proposed correlation-based prior loss adaptive weighting and weighted correlation-based model validation (CoReg-PGNN$_4$), on average, achieve the best empirical and prior-consistency RMSE improvements.
- Any prior knowledge of a given problem will add value to a data-driven framework such as the PGNN when the data is not fully representative. However, as evident in this work, the methods used to incorporate the priors into a NN training/validation framework might be detrimental to the PGNN empirical performance. Given the proposed methods and results, we conclude that for modeling problems with single or multiple priors, if the relationship between any of the prior errors and the empirical error is nonlinear or unknown, all the priors should be equally weighted in the validation performance index to select the most generalizable model.

The significance of this work in the field of prior-guided neural networks includes the following findings:

- For the first time in the literature, our proposed correlation-based adaptive weighting method for the prior losses provides a multi-prior, application-independent, scalable, and more generalizable approach. Unlike the work in the literature, in which custom designs prior-dependent adaptive weighting functions, our proposed adaptive weighting method leverages the correlation between the prior and empirical losses. Leveraging the correlation coefficient between the empirical and prior losses to weight prior losses adaptively is independent of the prior type, prior value, training data landscape, and the training data size.
- Our proposed scalable weighted correlation-based model validation is the first attempt in the literature to weight the prior error in the cross-validation performance index. The proposed weighted correlation-based model validation generates fresh insight into the use and value of priors toward more generalizable prior-guided neural networks.

> For example, we have shown that, on average, for modeling problems with multiple priors, using the equally weighted sum of the empirical and prior errors as the cross-validation performance index outperforms the adaptive weighting method proposed. However, for modeling problems with a single prior, the proposed weighted correlation-based model validation, on average, achieves the best empirical and prior-consistency RMSE improvements.

The weighting metrics proposed in this work utilize Pearson's correlation coefficient to assess the strength and direction of the linear relationships between the empirical and prior-inconsistency error variables. A limitation of this study is that Pearson's correlation coefficient would not be a suitable choice for applications where the variables are already known to have a nonlinear relation [40]. However, linearity is not a strict assumption of Pearson's correlation since the metric determines the linear association between the variables. Possible solutions include transforming the variables to have a normal distribution and linearizing the relationship between the variables. Another approach is to utilize other correlation metrics, such as Spearman's rank correlation or distance correlation [41, 42]. For example, Spearman's rank correlation can examine nonlinear monotonic relationships and does not require normally distributed variables.

Prior-knowledge-guided Neural Networks: Analysis of competing prior objectives and new correlation-based weighting

Mohamed Atwya and George Panoutsos

Abstract

Prior-knowledge-guided Neural Networks (PGNNs) are an emerging class of machine learning models that utilize prior knowledge in the form of constraint losses in the loss function. PGNNs use prior constraints (e.g., physics laws and empirical relations) losses in the learning stage to guide the optimizer towards more prior-consistent and generalizable neural networks. An interesting challenge posed by PGNNs is weighing the value of a prior constraint loss relative to the empirical and regularization losses. Existing work in PGNNs has demonstrated remarkable empirical success in utilizing up to two prior losses in a neural network loss function and using a grid search or an adaptive metric to weight the losses and ensure better generalizability. However, current adaptive weighting methods introduce further complexity, additional hyperparameters and are tailor-made for specific prior types and with particular applications in mind. The additional complexity and hyperparameters pose a further challenge in the presence of multiple competing prior loss functions. Furthermore, there are no attempts in the literature to adaptively weigh the prior errors in the validation performance index of PGNNs. In this work, we present multi-prior-scalable correlation-based adaptive weighting algorithms to balance the interplay between prior and empirical errors in the loss function and the validation performance index function. We show significant empirical and prior-consistency improvements for regression modeling problems with a boundary prior (up to 21.493% and 16.991% improvement, respectively) and multi-prior modeling problems (up to 15.191% and 8.096% improvement, respectively). Our results show (1) the importance of adaptively tuning weights instead of setting them as constants, (2) some priors are more valuable than others, (3) the relationship between the prior-consistency errors and the empirical error is not necessarily linear in the loss and validation performance index functions, and (4) multi-prior PGNN modeling empirical performance is affected more significantly by the prior loss weights than the prior validation performance index error weights. This work can set the agenda for further studies in weighting methods, such as examining more robust correlation measures and analyzing the effects of the PGNN model structure.

Index Terms

Physics-informed neural network, Constrained optimization, Adaptive weighting, Pearson's correlation.

I. Introduction

SINCE the formulation of the theory-guided data-driven modeling (TGDM) paradigm in [1], several papers were published on theory-guided machine learning methods in applications including material systems, satellite data fusion, and attention-deficit/hyperactivity disorder detection [2], [3], [4]. Within TGDM for machine learning, rising research interest is targeted towards prior-knowledge-guided neural networks [5], [6], [7], [8]. Prior-knowledge-guided neural networks softly impose theory-based constraints by adding prior loss functions that measure the prior violations of a NN. By guiding the optimization of NN loss functions via prior constraints, prior-knowledge-guided neural network loss functions have been successfully shown to reach more generalizable solutions across several scientific applications. Note that we use the term prior-knowledge-guided neural networks (PGNNs) as a reference to applying any domain knowledge as a constraint loss in the loss function of a NN such as in [5], [9], [10].

The introduction of prior constraints to loss functions in PGNNs has introduced new research challenges in machine learning. An interesting challenge posed by PGNNs is weighing the value of a prior constraint loss relative to the empirical and regularization losses. It is essential to strike a trade-off between the importance of the empirical, regularization, and prior losses to successfully benefit from the prior constraint loss and reach a more generalizable solution. Striking a trade-off between losses also poses the challenge of adaptively balancing the weights of each loss at each learning epoch to converge towards a generalizable solution [10]. It is also essential to take scalability into account and ensure that novel adaptive weighting algorithms can handle multiple prior constraint losses.

In [5] the authors construct elastic physics-guided NNs with a loss function including an empirical loss, two regularization losses, and a physics-based loss. The regularization loss weights were set to 1 ($\lambda_{\mathrm{R},1} = 1$ and $\lambda_{\mathrm{R},2} = 1$) and the physics-based loss weight was empirically set to a constant which scales the physics-based loss to a similar scale of the empirical loss. The authors in [11] add equality and inequity physics-based constraints to the NN loss function and use L_2 regularization. They set the L_2 regularization weight value to a constant $\lambda_R = 1$ and empirically optimize the physics-based constraint loss weight λ_{p}. Similarly, Jia et al. [9] use recurrent NNs with a loss function containing empirical, energy conservation equality

This work was supported by the University of Sheffield.

M. Atwya and G. Panoutsos are with the University of Sheffield, S1 3JD, UK (e-mail: matwya1@sheffield.ac.uk, g.panoutsos@sheffield.ac.uk).

constraint, and physics-based inequality constraint losses. They perform sensitivity studies on the impact of the constraint loss weights (E_{EC} and E_{DC}) by keeping one weight constant and varying the other. Raissi et al. propose novel physics-informed neural networks which compute physics-constrained neural network solutions to PDEs [6]. They demonstrate this method on the Burger's equation with Dirichlet boundary conditions and utilize an equally weighted loss function with an empirical loss (E_u) and a physics-based loss imposed by the Dirichlet boundary conditions (E_f).

The work in [5], [6], [7], [9], [11], [12] among several other works either sets the loss weights to unity or utilizes sensitivity studies such as grid search to assess the impact of the prior and model regularization loss weights on the models' empirical performance. However, manually tuning the loss weights is a trial-and-error procedure and assumes that the prior losses for a given problem have a constant weight through the learning epochs towards convergence. Within the context of PGNNs, adaptive loss weighting methods have been shown to improves the loss optimization robustness, and the empirical accuracy of the model predictions [8], [10], [13], [14], [15].

Chen et al. [16] propose GradNorm and make the loss weights trainable at each learning epoch via a separate optimizer such that the loss terms improve at the same relative rate with respect to their initial loss. They propose computing the L^2 norm of the gradient of each weighted loss with respect to the model weights and averaging the gradient norms across all the loss terms at each learning epoch. The averaged gradient norm is used as a baseline to place the gradient norms of the weight/loss pairs on a common scale by which they can determine the relative gradient sizes and dynamically adjust gradient norms so different tasks train at similar rates. The method has a single scalar hyperparameter that sets the strength of the restoring force, which pulls tasks back to a common training rate. The authors show that the proposed metric improves the empirical performance for regression and classification multi-prior problems relative to other adaptive metrics. In the case of several priors, this approach may suffer scalability issues due to the computationally expensive separate backward-pass required for each prior during each training epoch.

Similar to GradNorm, SoftAdapt [17] leverages the assumption of relative progress to balance the loss terms. However, the metric only considers the previous time-step and utilizes the loss gradient optimization components, eliminating the need to perform separate optimization for each loss. SoftAdapt takes advantage of each loss's gradient and step information from the previous time-step to dictate the loss weights relative to the one-step rate of change of each loss. The metric uses a softmax function and a scalar hyperparameter which dictates whether the largest or smallest one-step loss rate of change performance should have the largest or smallest weight. The modified variant of the metric additionally utilizes the current-step values of the losses to assign smaller weights to loss functions closer to their minima.

In [13] the authors develop a multi-fidelity physics-constrained NN and propose an adaptive weighting scheme that eliminates the individual loss weights, and instead, the weights are claimed to become proportional to the individual losses.

Wang et al. [8] draw motivation from the Adam optimizer [18] and propose Learning Rate Annealing, an adaptive loss weighting algorithm that utilizes the normalized back-propagated loss gradient statistic of each term in the loss function. They demonstrate that the proposed adaptive weighting algorithm can balance the interplay between the prior and the empirical loss and improves the empirical error by more than one order of magnitude. The authors in [14] implement the adaptive weighting metric proposed in [8] along with other contributions and report significantly better convergence and higher prediction accuracy when the adaptive loss weight technique is utilized instead of unity weights.

In [10] the authors propose two adaptive prior loss weighting metrics for two types of priors. They present an annealing weighting metric for the spectrum loss and a cold starting weighting metric for the characteristic loss. The annealing weighting metric takes on a large value in the initial learning epochs and then decays to null at a rate dictated by three hyperparameters. The cold starting weighting metric is null in the initial learning epochs and is then raised to a constant value using a sigmoid function. The cold starting metric has three hyperparameters which dictate the number of null weight epochs, the constant weight, and the growth rate of the sigmoid. The authors perform a search for the best model architecture and conduct a preliminary hyperparameter search using a dataset with 2000 samples to set the six hyperparameters in the adaptive weighting metrics. They demonstrate that adaptive weighting metrics result in learning more generalizable solutions. The proposed annealing weighting metric for the spectrum loss and the cold starting weighting metric for the characteristic loss are tailored to the loss types in the study. The authors' approach results in an application-tailored framework that requires a data set with enough samples to perform the model structure search, hyperparameter search, model training, model validation, and model testing.

Xiang et al. propose the use of Gaussian probabilistic models to define the loss terms and utilizing the noise collection to set the weight of each loss term [15]. The generalizable self-adaptive loss function method utilizes the maximum likelihood estimation method to automatically update the weights by updating the noise parameters in each learning epoch. The results show that adaptively updating the loss weights results in improved empirical prediction accuracy. However, this method suffers from scalability and sensitivity issues due to the hyperparameters (initial values for the noise collection), which affect the predictive performance by orders of magnitude.

In [19] the authors propose ReLoBRaLo a new adaptive weighting metric that combines methods from the Learning Rate Annealing [8] and SoftAdapt [17] weighting metrics. ReLoBRaLo utilizes the rate of change between consecutive training steps and a softmax normalization function like in SoftAdapt. However, they update the weights via exponential decay to utilize the loss statistics from all the previous steps like in Learning Rate Annealing. The novelty in ReLoBRaLo is the introduction

of a random look-back to the exponential decay, which decides whether to use the last epoch's loss statistics or use the loss statistics from all the previous training epochs. ReLoBRaLo uses two hyperparameters that dictate the exponential decay rate and the magnitude of the weights. The results show that ReLoBRaLo outperforms GradNorm, Learning Rate Annealing, SoftAdapt, and manual scaling in empirical accuracy and computational efficiency for the Helmholtz, Burgers, and Kirchhoff partial differential equations (PDEs). The authors also discover that the Helmholtz PDE benefits more from frequent look-backs than the Burgers and Kirchhoff PDEs. The study would have been more useful if it had included an analysis of the results in relation to the structure of the algorithms to explain why the algorithm works that way.

Although several studies have examined and verified the empirical performance effects of prior-consistency loss functions on PGNNS, researchers have not treated the competitive relationship between prior-consistency loss functions in much detail. Research on the subject has been mostly restricted to limited comparisons of vanilla NNs and PGNN variants (e.g., with/without labeled data, different model architectures, and with/without the empirical loss) with a fixed number of priors. Previous studies of PGNNs have not examined the impact of weighting competitive prior loss functions using single priors versus using $3+$ combinations of priors in PGNNs. Similar to the prior loss analysis gap, there is no analysis on using single priors in the literature versus using $3+$ combinations of priors in the validation performance index (PI) of PGNNs. There is also no attempt in the literature to adaptively or empirically weigh the prior errors in the validation performance index of PGNNs. Another gap in the PGNN literature is the lack of generalizability of the proposed methods. Most PGNN techniques presented in the literature are tailored to specific applications, use several sensitivity studies to set hyperparameters, and suffer from scalability limits.

To summarize, the prior-consistency adaptive weighting gaps in PGNNs include an analysis of using single versus multiple priors, methods to weigh prior errors in the validation performance index, application independence, and scalability. In this work, we aim to investigate and address these gaps via the following contributions towards prior-knowledge guided feedforward neural networks (PGNNs):

1) An analysis and comparison of the effects of the prior loss weights on the PGNN performance in the case of single versus multi-prior modeling problems.
2) Correlation-based prior loss adaptive weighting: A scalable multi-prior correlation-based adaptive weighting metric for choosing the prior-error weights during the learning step to balance the interplay between the competing prior loss terms in the cost function.
3) Weighted correlation-based model validation: A scalable multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step to balance the interplay between the different error terms in the cross-validation error function.

The remainder of the paper is structured as follows. In Section II we present the preliminaries and problem statement (Subsection II-A), the proposed adaptive prior loss error weighting metric (Subsection II-B), and the proposed validation error weighting metrics (Subsection II-C). In Section III we present a benchmark modeling problem, the type of prior losses utilized, benchmark model variants, and model variants using our proposed weighting metrics. In Section IV we present the results and analysis across five case studies which examine the impact of different prior losses. Finally, in Section VI we conclude our findings and provide potential research future directions.

II. Methodology

A. Preliminaries and Problem Statement

A single neuron output is given by Eq. 1 and 2. Similarly, the model predicted prior outputs $\hat{Z}_j^{(m)}$ are given by Eq. 3 and 4.

$$\underline{x}_j = \sum_{k \in K^{(j)}} \underline{W}_{kj} \underline{y}_k, j \in \{0, 1, \cdots, J\}, k \in \{0, 1, \cdots, K\}, \tag{1}$$

where $J \in \mathbb{N}$ is the number of nodes in layer j, $K \in \mathbb{N}$ is the number of nodes in layer k, $W_{kj} \in \mathbb{R}^{kj}$ are the weights between layers k and j, $y_k \in \mathbb{R}^k$ are the regression activation outputs of layer k, $x_j \in \mathbb{R}^j$ are the regression weighted sum values in layer j, and $K^{(j)} \in \mathbb{N}$ are the set of nodes from the k^{th} layer which feed layer j.

$$\underline{\hat{y}}_j = f_j(\underline{x}_j), \tag{2}$$

where $f_j(.)$ is the activation function for layer j and $y_j \in \mathbb{R}^j$ are the neuron outputs.

$$\underline{a}_j^{(m)} = \sum_{k \in K^{(j)}} \underline{W}_{kj} \underline{y}_k^{(m)}, m \in \{0, 1, \cdots, M\}, \tag{3}$$

Where $M \in \mathbb{N}$ is the number of priors available, $y_k^{(m)} \in \mathbb{R}^k$ are the prior activation outputs of layer k, and $a_j^{(m)} \in \mathbb{R}^j$ are the prior weighted sum values in layer j.

$$\hat{\underline{z}}_j^{(m)} = f_j\left(\underline{a}_j^{(m)}\right). \tag{4}$$

Assuming j is the output layer, the loss function for a prior-guided NN with an empirical loss, a regularization loss, and M prior losses is defined in Eq. 5.

$$\mathcal{L}_{\mathrm{PN}} = \mathcal{L}_{\mathrm{e}}(\hat{\underline{y}}_j, \underline{y}_j) + \gamma \mathcal{L}_{\mathrm{r}}(\underline{W}_{kj}) + \sum_{i=1}^{m} \lambda^{(m)} \mathcal{L}_{\mathrm{p}}^{(m)}\left(\hat{\underline{z}}_j^{(m)}, \underline{z}_j^{(m)}\right), \tag{5}$$

where $\mathcal{L}_{\mathrm{e}}$ is the empirical loss, $\mathcal{L}_{\mathrm{r}}$ is the regularization loss, and $\gamma \in \mathbb{R} : \gamma \geq 0$ is the regularization loss weight, $\mathcal{L}_{\mathrm{p}}^{(m)}$ are the prior losses, and $\lambda^{(m)} \in \mathbb{R} : \lambda^{(m)} \geq 0$ are the prior loss weights.

Typically, the model weights $\underline{W}_{kj}$ are randomly initialized and an iterative optimizer is used to find the weights which minimize the loss function for a set X of training data (Eq. 6). Assuming l is the network output layer, the solution $\hat{\underline{y}}_j$ for a given a set x_l of independent input features is approximated using the optimal weights via Eq. 7. Similarly, the predicted prior output $\hat{\underline{z}}_j^{(m)}$ for the prior input set $\breve{x}_l^{(m)}$ is given in Eq. 8. After the model is trained its empirical and prior-consistency performance is assessed using a validation and a testing data set. Several performance metrics exist in literature, in Eq. 9 and 10 we measure the empirical and prior consistency error via the root mean square error (RMSE).

$$\underline{W}_{kj}^{*} = \underset{\underline{W}_{kj} \in \mathbb{R}^D}{\arg\min} \quad \mathcal{L}_{PN}\left(X, \underline{W}_{kj}, \gamma, \lambda^{(m)}\right), \tag{6}$$

where $\underline{W}_{kj} \in \mathbb{R}^D$ is the set of all network weights and D is the number of network weights.

$$\hat{\underline{y}}_j = \mathcal{N}\left(\underline{x}_l; \underline{W}_{kj}^{*}\right), l \in \{0, 1, \cdots, L\}, \tag{7}$$

where $\mathcal{N}$ is the neural network, x_l are the empirical input features, and $L \in \mathbb{N}$ is the number of nodes in layer l.

$$\hat{\underline{z}}_j^{(m)} = \mathcal{N}\left(\underline{\breve{x}}_l^{(m)}; \underline{W}_{kj}^{*}\right), l \in \{0, 1, \cdots, L\}, \tag{8}$$

where $\breve{x}_l^{(m)}$ are the prior input features.

$$\mathcal{E}_{\mathrm{e}} = \sqrt{\frac{\sum_{i=1}^{n}\left(\hat{\underline{y}}_j^{(i)} - \underline{y}_j^{(i)}\right)^2}{n}}, \tag{9}$$

where n is the number of samples in $\underline{y}_j$.

$$\mathcal{E}_{\mathrm{p}}^{(m)} = \sqrt{\frac{\sum_{i=1}^{n}\left(\hat{\underline{z}}_j^{(m,i)} - \underline{z}_j^{(m,i)}\right)^2}{n^{(m)}}}, \tag{10}$$

where $n^{(m)}$ is the number of samples in $\underline{z}_j^{(m)}$.

Typically a validation dataset is used for an unbiased evaluation of a model fit on the training data set while tuning the model's hyperparameters. For example, we can train the model with different regularization loss and prior loss weights and pick the model with the smallest validation empirical error. For a working example, we define vectors of possible regularization loss weights and prior loss weights in Eq. 11 and 12.

$$\underline{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_c]^{\intercal}, c \in \{1, 2, \cdots, C\}, \tag{11}$$

where $C \in \mathbb{N}$ is the number of empirical loss weights to evaluate.

$$\underline{\lambda}^{(m)} = \left[\lambda_1^{(m)}, \lambda_2^{(m)}, \cdots, \lambda_g^{(m)}\right]^{\intercal}, g \in \{1, 2, \cdots, G\}, \tag{12}$$

where $G \in \mathbb{N}$ is the number of prior loss weights to evaluate.

The model weights corresponding to minimizing Eq. 6 with the Cartesian product of the loss weights in Eq. 11 and 12 is given in Eq. 13. The output predictions of each model candidate are then computed via Eq. 7 and 8.

$$\boldsymbol{W} = \begin{bmatrix} \underline{W}_{kj}^{*\,(1,1)} & \underline{W}_{kj}^{*\,(1,2)} & \cdots & \underline{W}_{kj}^{*\,(1,g)} \\ \underline{W}_{kj}^{*\,(2,1)} & \underline{W}_{kj}^{*\,(2,2)} & \cdots & \underline{W}_{kj}^{*\,(2,g)} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{W}_{kj}^{*\,(c,1)} & \underline{W}_{kj}^{*\,(c,2)} & \cdots & \underline{W}_{kj}^{*\,(c,g)} \end{bmatrix}. \tag{13}$$

$$\boldsymbol{\mathcal{E}}_{\mathbf{e}} = \begin{bmatrix} \mathcal{E}_{\mathrm{e}}^{(1,1)} & \cdots & \mathcal{E}_{\mathrm{e}}^{(1,g)} \\ \vdots & \ddots & \vdots \\ \mathcal{E}_{\mathrm{e}}^{(c,1)} & \cdots & \mathcal{E}_{\mathrm{e}}^{(c,g)} \end{bmatrix}. \tag{14}$$

$$\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)} = \begin{bmatrix} \mathcal{E}_{\mathrm{p}}^{(m,1,1)} & \cdots & \mathcal{E}_{\mathrm{p}}^{(m,1,g)} \\ \vdots & \ddots & \vdots \\ \mathcal{E}_{\mathrm{p}}^{(m,c,1)} & \cdots & \mathcal{E}_{\mathrm{p}}^{(m,c,g)} \end{bmatrix}. \tag{15}$$

Before the formulation of PGNNs, the empirical error was used as the performance index to validate the model fit. The weight vector $\underline{W}_{kj}^{*}{}^{(c,g)}$ with the smallest empirical error $\mathcal{E}_{\mathrm{e}}^{(c,g)}$ is chosen. However, with PGNNs, the prior errors are available to validate the model fit. Where the prior knowledge is utilized in the validation step, the validation error function is given by Eq. 16. Similar to the loss weights, the performance index used to select the best model candidate has the empirical error weight φ and prior-based error weights $\rho^{(m)}$ (Eq. 16). Note that he empirical and prior errors in Eq. 16 are re-scaled via min-max normalization such that each error contributes equally (Eq. 18 and 19).

The hyperparameters φ and $\rho^{(m)}$ dictate the importance of the empirical performance as well as the prior-consistency when selecting a model during the validation step (Eq. 16). Once the hyperparameters are chosen, the matrix $\boldsymbol{\mathcal{E}}_{\mathbf{PN}}$ is minimized in Eq. 17. The index (b,t) of the validation error minima $\mathcal{E}_{\mathrm{PN}}^{*}{}^{(b,t)}$ are then used to find the optimal set of weights $\boldsymbol{W}^{(b,t)}$.

$$\boldsymbol{\mathcal{E}}_{\mathbf{PN}} = \varphi \boldsymbol{\mathcal{E}}'_{\mathbf{e}} + \sum_{i=1}^{m} \rho^{(m)} \boldsymbol{\mathcal{E}}'_{\mathbf{p}}{}^{(m)}, \tag{16}$$

where $\mathcal{E}'_{\mathrm{e}}$ is the normalized validation empirical error, $\varphi \in \mathbb{R} : \varphi \geq 0$ is the empirical error weight, $\mathcal{E}_{\mathrm{p}}^{(m)'}$ are the normalized validation prior errors, and $\rho^{(m)} \in \mathbb{R} : \rho^{(m)} \geq 0$ are the prior error weights.

$$\mathcal{E}_{\mathrm{PN}}^{*}{}^{(b,t)} = \underset{b \in c, t \in g}{\arg\min} \quad \boldsymbol{\mathcal{E}}_{\mathbf{PN}}{}^{(b,t)}. \tag{17}$$

$$\boldsymbol{\mathcal{E}}'_{\mathbf{e}} = \frac{\boldsymbol{\mathcal{E}}_{\mathbf{e}}\left(\underline{\hat{y}}_j, \underline{y}_j\right) - min\left(\boldsymbol{\mathcal{E}}_{\mathbf{e}}\right)}{max\left(\boldsymbol{\mathcal{E}}_{\mathbf{e}}\right) - min\left(\boldsymbol{\mathcal{E}}_{\mathbf{e}}\right)}. \tag{18}$$

$$\boldsymbol{\mathcal{E}}'_{\mathbf{p}}{}^{(m)} = \frac{\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)}\left(\underline{\hat{z}}_j^{(m)}, \underline{z}_j^{(m)}\right) - min(\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)})}{max(\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)}) - min(\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)})}. \tag{19}$$

The problem formulation leads to four hyperparameters, the regularization loss weight, the prior loss weights, the empirical validation error weight, and the prior validation error weights (Eq. 5 and 16). In Subsections II-B and II-C we propose correlation-based metrics to balance the interplay between the empirical and prior value in training and validation. Note that the regularization loss weight is outside the scope of this work.

B. Correlation-based prior loss adaptive weighting

The prior-based loss functions have the hyperparameter weights $\lambda_{\mathrm{p}} \geq 0$ which dictate the importance of prior-consistency (Eq. 5). We make the following remarks and propose a multi-prior correlation-based adaptive weighting metric for choosing the prior-loss weights during the learning step to balance the interplay between the different loss terms in the cost function.

a) Data set size versus prior-knowledge value: MLPs are universal approximator models, and accordingly, a two-layer feed-forward MLP with enough hidden units can model any continuous function on a finite interval, given there is enough data to estimate the network weights (Stone-Weierstrass theorem) [20]. Accordingly, there is no benefit in incorporating prior knowledge with MLPs where a fully representative dataset is available (enough samples with sufficient sample density). Accordingly, for such a representative dataset, the correlation coefficient between the empirical and prior-inconsistency performances would be high (i.e., the prior knowledge has a low value and should have a smaller weight in the learning step). However, for small data sets, the correlation between the empirical and the prior-inconsistency errors is likely to be low (i.e., the prior knowledge is more valuable and should have a more significant weight in the learning step).

b) Early learning stage versus prior-knowledge value: When local gradient methods optimize highly non-convex loss functions of NNs, the optimizer tends to converge towards a local minimum [21]. Once close to a local minimum, the optimizer refines the solution until convergence (i.e., satisfying a set of stopping criteria). The desirable outcome is to avoid the selection of erroneous local minima and instead converge towards a more generalizable local minimum. Accordingly, during the initial learning stage (first few learning iterations), using large prior-based loss function weights is helpful to guide the optimizer towards a more generalizable solution.

c) Late learning stage versus prior-knowledge value: The optimizer will start converging towards a generalizable local minimum if the prior knowledge has been utilized with a large weight in the initial learning stage. Once the optimizer is close to convergence or oscillates around the generalizable local minima, the prior knowledge is no longer valuable.

d) Loss function multi-modality versus prior-knowledge value: While looking at the early/late learning stages versus the prior knowledge value, it is essential to account for the non-linearity of the loss function. The loss landscape is high-dimensional and highly non-convex. Therefore, if the optimizer is close to convergence and a prior weight decays to 0, the optimizer can get drawn to a different less-generalizable minimum. Accordingly, after a prior weight has decayed to 0, where necessary, an adaptive prior weighting metric should be capable of "cold booting" the prior weight.

e) Training data versus prior-knowledge value: In some modeling problems, it might be the case that even though the prior knowledge is not explicitly represented in the training data, an NN model can still learn this prior knowledge without being provided the prior knowledge. For example, a typical prior is an initial condition. However, the initial condition is usually within the interpolation range of the data set used to train the neural network models. Vanilla NNs are good interpolators, and therefore if the prior is within the interpolation range, then it has a low value.

Based on these remarks, we define the following prior adaptive weighting requirements:

- The prior weights should initially have a large value, $\lambda^{(m)} \approxeq 1$ since data sets are rarely representative and the prior knowledge is valuable in the early learning stage (remark 1).
- The prior weights should initially have a large value, $\lambda^{(m)} \approxeq 1$, to direct the optimizer towards a more generalizable local minimum (remark 2.1).
- The prior weights should decay to $\lambda^{(m)} \approxeq 0$ once the optimizer is close to convergence (remark 2.2). However, if the optimizer is drawn to a different less-generalizable minimum after the weight decays to $\lambda^{(m)} \approxeq 0$, the weight should be increased again (remark 2.3).
- If a prior is within the subspace spanned by the training samples (i.e., interpolation range), the prior's weight should rapidly decay to $\lambda^{(m)} \approxeq 0$, relative to more valuable priors (remark 3).

According to these remarks and requirements, we utilize Pearson's correlation coefficient to develop a multi-prior adaptive weighting metric. Starting with a random set of model weights, in the initial learning stage, the correlation between the empirical loss and the prior losses tends to be weak as the optimizer is far from a generalizable minimum. However, as the optimizer gets closer to a generalizable minimum, the correlation between the empirical and prior losses increases. For an iterative optimizer with $N \in \mathbb{N}$ being the maximum number of iterations, we define the empirical and prior loss vectors as $\underline{\mathcal{L}}_{\mathrm{e}}$ and $\underline{\mathcal{L}}_{\mathrm{p}}^{(m)}$ in Eq. 20 and 21.

To achieve the desired weighting behavior, at each optimization iteration, we take the complement of the value 1 to the correlation coefficients between the empirical and prior loss vectors as in Eq. 22. The weights are updated in Eq. 23 using a moving average filter from the work in [8]. The correlation coefficients are set to 0, and the prior weights are set to 1 for the first three iterations (Eq. 22 and 23). Note that if the optimizer is drawn to a different less-generalizable minimum after the weight decays to 0, the correlation will weaken, and the weight increases in line with remark 2.3.

$$\underline{\mathcal{L}}_{\mathrm{e}} = \left[\mathcal{L}_{\mathrm{e}}^{(1)}, \mathcal{L}_{\mathrm{e}}^{(2)}, \cdots, \mathcal{L}_{\mathrm{e}}^{(n)}\right]^{\intercal}, n \in \{1, 2, \cdots, N\}. \tag{20}$$

$$\underline{\mathcal{L}}_{\mathrm{p}}^{(m)} = \left[\mathcal{L}_{\mathrm{p}}^{(m,1)}, \mathcal{L}_{\mathrm{p}}^{(m,2)}, \cdots, \mathcal{L}_{\mathrm{p}}^{(m,n)}\right]^{\intercal}, n \in \{1, 2, \cdots, N\}. \tag{21}$$

$$\acute{\lambda}_n^{(m)} = \begin{cases} 0 & \text{for} \quad n \leq 3 \\ 1 - cov\left(\underline{\mathcal{L}}_{\mathrm{e}}, \underline{\mathcal{L}}_{\mathrm{p}}^{(m)}\right) / \left(\sigma_{\underline{\mathcal{L}}_{\mathrm{e}}} \sigma_{\underline{\mathcal{L}}_{\mathrm{p}}^{(m)}}\right) & \text{for} \quad n > 3, \end{cases} \tag{22}$$

where $cov\left(\underline{\mathcal{L}}_{\mathrm{e}}, \underline{\mathcal{L}}_{\mathrm{p}}^{(m)}\right)$ is the covariance of $\underline{\mathcal{L}}_{\mathrm{e}}$ and $\underline{\mathcal{L}}_{\mathrm{p}}^{(m)}$, $\sigma_{\underline{\mathcal{L}}_{\mathrm{e}}}$ is the standard deviation of $\underline{\mathcal{L}}_{\mathrm{e}}$, and $\sigma_{\underline{\mathcal{L}}_{\mathrm{p}}^{(m)}}$ is the standard deviation of $\underline{\mathcal{L}}_{\mathrm{p}}^{(m)}$.

$$\lambda_n^{(m)} = \begin{cases} 1 & \text{for} \quad n \leq 3 \\ (1 - \beta^{(m)})\lambda_{n-1}^{(m)} + \beta^{(m)} \acute{\lambda}_n^{(m)} & \text{for} \quad n > 3, \end{cases} \tag{23}$$

where $n \in \mathbb{Z}^+$ is the training iteration and $\beta^{(m)} \in \mathbb{R} : \beta^{(m)} > 0$ is the moving average weight hyperparameter for each prior.

In this work, we utilize Pearson's correlation coefficient and assumed linear relationships between the empirical and prior-inconsistency loss errors. The relevant assumptions made to use Pearson's correlation coefficient:

1) The two variables ($\mathcal{L}_{\mathrm{e}}$ and each $\mathcal{L}_{\mathrm{p}}^{(m)}$) are on a continuous scale.
2) Pairs of observations are independent. Meaning, each pair of $\mathcal{L}_{\mathrm{e}}$ and $\mathcal{L}_{\mathrm{p}}^{(m)}$ is independent of other pairs across the training iterations.
3) Linearity, the two continuous variables should have a linear relationship.
4) There are no univariate or multivariate outliers in the observations.

5) Both continuous variables should follow a bivariate normal distribution.
6) Both continuous variables should be homoscedastic. Meaning, the variances along the line of best fit of the two variables remain similar along the line.

In practice, when working with real-world data, it is common to violate one or more of the stated assumptions. However, it is often possible to make adjustments to the data to satisfy certain assumptions. The effects of these assumptions and any violations are discussed with the results in Section IV.

As the number of training iterations increases and the optimizer starts converging to a generalizable solution, the correlations between the empirical and prior losses tend to decay exponentially and never reach zero. The exponential decay of the correlation can result in the optimizer failing to recognize when it converges and continues futile iterations. To avoid futile iterations and the problem of floating-point underflow, we set a step tolerance and an absolute tolerance as stopping criteria. The step tolerance applies a lower bound on the absolute change in the weight during a step $\left(\left|\acute{\lambda}_n^{(m)} - \acute{\lambda}_{n-1}^{(m)}\right| > 10^{-4} : n > 3\right)$. If the absolute weight change step is smaller than the step tolerance, the weight is set to 0. The absolute tolerance applies a lower bound on the the absolute value of the weight $\left(\left|\acute{\lambda}_{n+1}^{(m)}\right| > 10^{-3} : n > 3\right)$. If the absolute weight is smaller than the absolute tolerance, the weight is set to 0.

C. *Weighted correlation-based model validation*

The hyperparameters φ and $\rho^{(m)}$ dictate the importance of the empirical performance as well as the prior consistency when selecting a model during the validation step (Eq. 16). We make the following remark and propose a multi-prior correlation-based weighting metric for choosing the empirical and prior-error weights during the validation step to balance the interplay between the different error terms in the validation error function.

a) Empirical performance versus prior-consistency significance: Intuitively, the candidate model with the best empirical performance on the validation data set would lead to the best result on the testing data set. However, from the list of candidate models, the model with the best empirical performance is not necessarily the most prior-consistent. If there is a strong correlation between the empirical and prior errors for the candidate models, the prior errors are less valuable when selecting a model. However, when the correlation is weak, utilizing a performance index that balances the empirical error and prior consistency errors is likely to lead to a more generalizable model selection.

Based on the empirical performance versus prior-consistency significance remark, we define the following prior adaptive weighting requirement:

- A weaker correlation between the empirical and the prior-consistency errors indicates that the candidate models are less generalizable and vice versa. Accordingly, when the empirical and prior-consistency error correlation is weak, the prior error weight should have a larger value, and the empirical error weight should have a smaller value. Contrarily, if the empirical and prior-consistency error correlation is strong, the prior error weight should have a smaller value, and the empirical error weight should have a larger value.

According to these remarks and requirements, we utilize Pearson's correlation coefficient characteristics to develop a multi-prior error weighting metric (Eq. 24 and 25). To achieve the desired weighting behavior, the empirical error weight φ is computed as the averaged correlation between the empirical error vector and each prior error vector. Accordingly, if the correlation between the empirical and prior errors is strong, the empirical error weight would be large, and vice versa. The prior error weights $\rho^{(m)}$ are computed by taking the complement of the value 1 to the correlation coefficients between the empirical and prior error vectors as in Eq. 25. The error weight metrics are divided by the number of priors (m) such that the sum of the weights is unity. The assumptions made to use Pearson's correlation coefficient in the validation error weighting metrics are the same as the loss weighting metric assumptions in Subsection II-B.

$$\varphi = \frac{\sum_{i=1}^{m} \left(cov\left(\underline{\mathcal{E}_{\mathrm{e}}}, \underline{\mathcal{E}_{\mathrm{p}}}^{(m)}\right) / \left(\sigma_{\underline{\mathcal{E}_{\mathrm{e}}}} \sigma_{\underline{\mathcal{E}_{\mathrm{p}}}^{(m)}}\right)\right)}{m}, \tag{24}$$

where $\underline{\mathcal{E}_{\mathrm{e}}}$ is the vectorization of $\boldsymbol{\mathcal{E}}_{\mathbf{e}}$ and $\underline{\mathcal{E}_{\mathrm{p}}}^{(m)}$ is the vectorization of $\boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m)}$.

$$\rho^{(m)} = \frac{1 - cov\left(\underline{\mathcal{E}_{\mathrm{e}}}, \underline{\mathcal{E}_{\mathrm{p}}}^{(m)}\right) / \left(\sigma_{\underline{\mathcal{E}_{\mathrm{e}}}} \sigma_{\underline{\mathcal{E}_{\mathrm{p}}}^{(m)}}\right)}{m}. \tag{25}$$

III. Evaluation

We test the proposed methods capability of modeling the Bohachevsky function with a data set of 1000 uniformly distributed samples (Eq. 26). We set $\boldsymbol{x} \coloneqq [\underline{x}_1, \underline{x}_2]$ as the set of independent input features and $\underline{y}$ as the set of dependent output variables. The features $\underline{x}_1$ and $\underline{x}_2$ were transformed to have a zero mean and a unit standard deviation.

$$\underline{y} = \underline{x}_1^2 + 2 \cdot \underline{x}_2^2 - 0.3 \cdot \cos\left(3 \cdot \pi \underline{x}_1\right) - 0.4 \cdot \cos\left(4 \cdot \pi \underline{x}_2\right) + 0.7, \tag{26}$$

where x_1 and $x_2 \in [-100.00, 100.00]$.

We use the boundary, initial, and symmetry constraints as prior loss functions to demonstrate the proposed methods. The prior predicted and expected values are given by Eq. 27. The boundary constraint input features are given in Eq. 28-30. For each boundary we generate $D = 250$ uniformly distributed samples. The uniform distribution is given in Eq. 28 and the two input features are given in Eq. 29 and 30. The initial constraint input features are set as $\breve{\boldsymbol{x}}^{(2)} = [0, 0]$. The symmetry constraints include vertical and horizontal bilateral symmetry conditions and the input features are set as $\breve{\boldsymbol{x}}^{(3)} = [-\underline{x}_1, \underline{x}_2]$ and $\breve{\boldsymbol{x}}^{(4)} = [\underline{x}_1, -\underline{x}_2]$, respectively. Note that no new data sets are created for the symmetry conditions, instead we utilize the training/testing data and flip the input feature signs.

$$\left[\underline{\hat{z}}_j^{(m)}, \underline{z}_j^{(m)}\right] = \left[\underline{\hat{y}}\left(\breve{\boldsymbol{x}}^{(m)}\right), \underline{y}\left(\breve{\boldsymbol{x}}^{(m)}\right)\right]. \tag{27}$$

$$[g_1, g_2, \cdots, g_d] \overset{\text{i.i.d.}}{\sim} U(-100, 100), d \in \{1, 2, \cdots, D\}, \tag{28}$$

where $U(-100, 100)$ is a uniform distribution with -100 and 100 as the smallest and largest values of g_d, respectively.

$$\underline{\breve{x}}_1^{(1)} = \left[-100 \cdot \underline{1}, 100 \cdot \underline{1}, \underline{g}, \underline{g}\right]^\intercal, \tag{29}$$

where $\underline{1}$ is an all-ones vector with D elements.

$$\underline{\breve{x}}_2^{(1)} = [\underline{g}, \underline{g}, -100 \cdot \underline{1}, 100 \cdot \underline{1}]^\intercal, \tag{30}$$

where $\underline{1}$ is an all-ones vector with D elements.

A. Model training

We use a multi-layer perceptron with two input neurons, one hidden layer with 30 hidden units, and an output layer with one neuron. We use the rectified linear unit activation function for the hidden layer and a linear activation function for the output layer. The empirical loss $\mathcal{L}_{\text{e}}(\underline{\hat{y}}_j, \underline{y}_j)$ was computed via the half sum squared error (HSSE) and L2 regularization was used for the regularization loss $\mathcal{L}_{\text{r}}(\underline{W}_{kj})$. To quantify the constraints' violations $\mathcal{L}_{\text{p}}^{(m)}\left(\underline{\hat{z}}_j^{(m)}, \underline{z}_j^{(m)}\right)$, we utilize the HSSE metric. To minimize the loss functions (Eq. 5 and 31) and find the optimal weights we use the scaled conjugate gradient optimization algorithm. The algorithm was allowed a maximum of $N = 5000$ iterations to find the solution.

B. Benchmark models

We test three benchmark models (NN, PGNN$_1$, and PGNN$_2$) to examine our proposed methods' empirical and prior consistency improvements. The benchmark models utilize the grid search method to tune the loss weight hyperparameters γ and $\lambda^{(m)}$.

- NN: A black-box vanilla Neural Network. The loss function consists of an empirical loss and a non-adaptively weighted (line search) L2 regularization loss (Eq. 31). The CV PI for the NN model is the empirical error ($\mathcal{E}_{\text{e}}$). The vector $\underline{\mathcal{E}_{\text{e}}}$ corresponding to a vector of different L2 regularization loss weights ($\underline{\gamma}$) is minimized in Eq. 32. The index (b) of the validation error minima $\mathcal{E}_{\text{e}}^{*(b)}$ are then used to find the optimal set of weights $\underline{W}_{kj}^{*\,(b)}$.

$$\mathcal{L}_{\text{NN}} = \mathcal{L}_{\text{e}} + \gamma \mathcal{L}_{\text{r}}. \tag{31}$$

$$\begin{aligned} & \left[\mathcal{E}_{\text{e}}^{*(b)}, \underline{W}_{kj}^{*\,(b)}\right] = \\ \underset{b \in c}{\arg\min} \quad & \underline{\mathcal{E}}_{\text{e}}\left(\underline{\hat{y}}_j^{(b)}\left(\boldsymbol{W}^{(b)}(\underline{\gamma}_b), \boldsymbol{x}\right), \underline{y}_j\right) \\ \text{subject to} \quad & \underline{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_{\text{c}}]^\intercal, \\ & c \in \{1, 2, \cdots, C\}, \end{aligned} \tag{32}$$

  where $\underline{\gamma} = [1.000 \cdot 10^{-7},\ 5.995 \cdot 10^{-7},\ 3.594 \cdot 10^{-6},\ 2.154 \cdot 10^{-5},\ 1.292 \cdot 10^{-4},\ 7.743 \cdot 10^{-4},\ 4.642 \cdot 10^{-3},\ 2.783 \cdot 10^{-2},\ 1.668 \cdot 10^{-1},\ 1.000]^\intercal$ and $C = 10$.
- PGNN$_1$: This refers to a benchmark Prior-Guided Neural Network with a loss function consisting of an empirical loss, L2 regularization, and prior losses (Eq. 5). The L2 regularization and prior loss weights and their corresponding model weights ($\underline{W}_{kj}^{*\,(b,t)}$) are non-adaptively optimized via a grid search (Eq. 33). Note that if the modeling case study has multiple priors, all the prior losses ($\underline{\lambda}^{(m)}$) share the same weight during the gird search. The CV PI for PGNN$_1$ only utilizes the empirical error, hence the error weights in Eq. 33 are set to $\varphi = 1$ and $\rho^{(m)} = 0$.

- PGNN$_2$: This PGNN benchmark variant has the same non-adaptively weighted loss function as PGNN$_1$ (Eq. 5). Similar to PGNN$_1$, for PGNN$_2$ the L2 regularization and prior loss weights and their corresponding model weights ($\underline{W}^{*}_{kj}{}^{(b,t)}$) are non-adaptively optimized via a grid search (Eq. 33). However, the CV PI for PGNN$_2$ consists of the equally weighted sum of the empirical and prior errors. Therefore, the error weights in Eq. 33 are set to $\varphi = 1$ and $\rho^{(m)} = 1$.

C. Proposed model variants

Our proposed loss weighting and validation error weighting methods are evaluated via the following Correlation Regulated (CoReg) PGNN model variants. The CoReg-PGNN models have a loss function consisting of an empirical loss, L2 regularization, and prior losses (Eq. 5).

- CoReg-PGNN$_1$: The L2 regularization and prior loss weights and their corresponding model weights ($\underline{W}^{*}_{kj}{}^{(b,t)}$) are non-adaptively optimized via a grid search (Eq. 33). Note that if the modeling case study has multiple priors, all the prior losses ($\underline{\lambda}^{(m)}$) share the same weight during the gird search. The CV PI for CoReg-PGNN$_1$ consists of the empirical and prior errors and utilizes the correlation-based weighting metric to dictate the empirical φ (Eq. 24) and prior-error weights $\rho^{(m)}$ (Eq. 25).

$$
\begin{aligned}
& \left[\mathcal{E}^{*}_{\mathrm{PN}}{}^{(b,t)}, \underline{W}^{*}_{kj}{}^{(b,t)}\right] = \\
\underset{b\in c, t\in g}{\arg\min} \quad & \boldsymbol{\mathcal{E}}_{\mathbf{PN}}\left(\boldsymbol{\mathcal{E}}_{\mathbf{e}}{}^{(b,t)}\left(\underline{\gamma}_b\right), \boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m,b,t)}\left(\underline{\lambda}^{(m)}_t\right), \varphi, \rho^{(m)}\right) \\
\text{subject to} \quad & \underline{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_c]^{\intercal}, \\
& \underline{\lambda}^{(m)} = [\lambda_1, \lambda_2, \cdots, \lambda_g]^{\intercal}, \\
& c \in \{1, 2, \cdots, C\},\ g \in \{1, 2, \cdots, G\},
\end{aligned}
\tag{33}
$$

  where $C = 10$, $G = 10$, $\underline{\gamma} := \underline{\lambda}^{(m)}$, $\underline{\lambda}^{(m)} = [1.000 \cdot 10^{-7},\ 5.995 \cdot 10^{-7},\ 3.594 \cdot 10^{-6},\ 2.154 \cdot 10^{-5},\ 1.292 \cdot 10^{-4},\ 7.743 \cdot 10^{-4},\ 4.642 \cdot 10^{-3},\ 2.783 \cdot 10^{-2},\ 1.668 \cdot 10^{-1},\ 1.000]^{\intercal}$.
- CoReg-PGNN$_2$: The second Correlation Regulated PGNN variant only utilizes the empirical error as the CV PI, hence the empirical and prior error weights in Eq. 34 are set to $\varphi = 1$ and $\rho^{(m)} = 0$, respectively. The prior loss weights are computed via the proposed correlation-based adaptive metric in Eq. 22 and 23. The proposed correlation-based adaptive loss weighting metric introduces the moving average weight hpyerparameter $\beta^{(m)} \in \mathbb{R} : \beta^{(m)} > 0$. The L2 regularization and moving average hpyerparameter weights and their corresponding model weights $\left(\underline{W}^{*}_{kj}{}^{(b,t)}\right)$ are non-adaptively optimized via a grid search (Eq. 34).
- CoReg-PGNN$_3$: The CV PI for CoReg-PGNN variant consists of the equally weighted sum of the empirical and prior errors. Therefore, the empirical and prior error weights in Eq. 34 are set to $\varphi = 1$ and $\rho^{(m)} = 1$, respectively. The prior loss weights are computed via the proposed correlation-based adaptive metric in Eq. 22 and 23. The L2 regularization and moving average hpyerparameter weights and their corresponding model weights ($\underline{W}^{*}_{kj}{}^{(b,t)}$) are non-adaptively optimized via a grid search (Eq. 34).
- CoReg-PGNN$_4$: The CV PI consists of the empirical and prior errors and utilizes the correlation-based weighting metric to dictate the empirical φ (Eq. 24) and prior-error weights $\rho^{(m)}$ (Eq. 25). The L2 regularization and moving average hpyerparameter weights and their corresponding model weights ($\underline{W}^{*}_{kj}{}^{(b,t)}$) are non-adaptively optimized via a grid search (Eq. 34). This CoReg-PGNN variant combines the two weighting metrics proposed in this work; (1) correlation-based prior loss adaptive weighting (Eq. 22 and 23) and (2) weighted correlation-based model validation (Eq. 24 and 25).

$$
\begin{aligned}
& \left[\mathcal{E}^{*}_{\mathrm{PN}}{}^{(b,t)}, \underline{W}^{*}_{kj}{}^{(b,t)}\right] = \\
\underset{b\in c, t\in g}{\arg\min} \quad & \boldsymbol{\mathcal{E}}_{\mathbf{PN}}\left(\boldsymbol{\mathcal{E}}_{\mathbf{e}}{}^{(b,t)}\left(\underline{\gamma}_b\right), \boldsymbol{\mathcal{E}}_{\mathbf{p}}{}^{(m,b,t)}\left(\underline{\beta}^{(m)}_t\right), \varphi, \rho^{(m)}\right) \\
\text{subject to} \quad & \underline{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_c]^{\intercal},\ c \in \{1, 2, \cdots, C\}, \\
& \underline{\beta}^{(m)} = [\beta_1, \beta_2, \cdots, \beta_g]^{\intercal},\ g \in \{1, 2, \cdots, G\},
\end{aligned}
\tag{34}
$$

  where $C = 10$, $G = 10$, $\underline{\gamma} = [1.000 \cdot 10^{-7},\ 5.995 \cdot 10^{-7},\ 3.594 \cdot 10^{-6},\ 2.154 \cdot 10^{-5},\ 1.292 \cdot 10^{-4},\ 7.743 \cdot 10^{-4},\ 4.642 \cdot 10^{-3},\ 2.783 \cdot 10^{-2},\ 1.668 \cdot 10^{-1},\ 1.000]$, and
  $\underline{\beta}^{(m)} = [0.010,\ 0.120,\ 0.230,\ 3.400,\ 0.450,\ 0.560,\ 0.670,$
  $0.780,\ 0.890,\ 1.000]^{\intercal}$.

Note that for all the model variants tested, we perform a second fine search with a finer vector of 10 linearly spaced points around the best-estimated values $\gamma^{*} \pm 1.000$, ${\lambda^{(m)}}^{*} \pm 1.000$, and ${\beta^{(m)}}^{*} \pm 0.100$.

D. Model Validation and Testing

We apply 3-times-repeated 5-fold cross-validation and use the root mean square error (RMSE) metric for the empirical and prior-based CV error functions (Eq. 9, 10, and 27). Additionally, the repeated CV was performed 12 times using randomly initialized model weights for each NN and PGNN model variant.

We use an out-of-sample (OOS) data set with $n = 1000$ samples to test the empirical and initial/symmetry prior-consistency RMSE performance (Eq. 9, 10, and 27). Similarly, the boundary prior was tested via an OOS boundary data set with 1000 samples (Eq. 10 and 28-30). Note that the OOS data sets were not used in training/validation and are only used to test the model variants.

IV. Results and Analysis

We test the proposed methods via five case studies. In the first four case studies, we test the effectiveness of the proposed method using only one of the four priors per case study (boundary, initial, horizontal symmetry, and vertical symmetry). Case study five utilizes the four priors to test the effectiveness of the proposed methods for problems with multiple priors. Tables I-V reports the empirical and prior-consistency RMSEs of each case study. Note that the results reported in Tables I-V are the average and standard deviation RMSEs of the methods repeated 12 times using randomly initialized model weights.

A. Case study one

In case study one, we apply the methods described using only the boundary prior and show the results in Table I. PGNN_1 and PGNN_2 have a similar empirical and prior-consistency RMSE to the vanilla NN, but the vanilla NN results in the poorest empirical RMSE performance. CoReg-PGNN_1 improves the empirical and prior-consistency performance in comparison to NN, PGNN_1, and PGNN_2. Specifically, we can see an improvement in CoReg-PGNN_1 in comparison to PGNN_2 which shows the benefit of using the proposed weighted validation PI (Eq. 24, 25, and 33) instead of an equally weighted validation PI.

When we use only the empirical error as the CV PI, coReg-PGNN_2 outperforms PGNN_1 showing the benefit of using the prior loss adaptive weighting metric instead of using a grid search to tune the loss weights. Using an equally weighted CV PI, coReg-PGNN_3 in comparison to PGNN_2 improves the empirical and prior-consistency performance demonstrating the benefit of using the prior loss adaptive weighting metric opposed to using a grid search to tune the loss weights.

Using the two proposed weighting metrics in coReg-PGNN_4 results in the best empirical and prior-consistency RMSE performance. Specifically, relative to the vanilla NN, coReg-PGNN_4 results in a 21.493% improvement in the empirical RMSE and a 16.991% prior-consistency RMSE improvement.

B. Case study two

In case study two, we use the initial prior. Case study two results are provided in Table II. With the initial prior, we can see that using a grid search to optimize the loss hyperparameters (PGNN_1) degrades the empirical and prior-consistency performance relative to the vanilla NN. However, when the prior consistency is used in the equally weighted validation PI (PGNN_2), the empirical and prior-consistency RMSEs are improved relative to the vanilla NN.

When we apply the proposed weighted correlation-based model validation in coReg-PGNN_1, the empirical performance is degraded relative to PGNN_2. However, coReg-PGNN_1 results in the best prior-consistency RMSE. coReg-PGNN_1 performance indicates that the relationship between the empirical and initial prior-consistency RMSEs is not linear in the validation step. The non-linear relationship is intuitive since the initial prior is a single sample compared to a 1000 testing samples distributed across the input features x_1 and $x_2 \in [-100.00, 100.00]$.

When we use only the empirical error as the CV PI, coReg-PGNN_2 outperforms NN/PGNN_1 and performs similarly to PGNN_2 empirical-performance-wise. The coReg-PGNN_2 model performance demonstrates the benefit of using the prior loss adaptive weighting metric and indicates that it is acceptable to assume a linear relationship between the empirical and initial prior-consistency losses in the training step. Subsequently, the coReg-PGNN_2 model performance also implies that the validation PI function is more sensitive than the loss function in terms of weighting the prior consistency.

Using an equally weighted CV PI, coReg-PGNN_3, albeit comparable to PGNN_2 and coReg-PGNN_2, results in the best empirical RMSE (1000 testing samples). However, CoReg-PGNN_3 in comparison to PGNN_2 minimally improves the empirical RMSE, increases the empirical RMSE standard deviation, and degrades the prior consistency RMSE (one sample). However, when we combine the proposed validation and loss weighting metrics, coReg-PGNN_4 performance degrades relative to coReg-PGNN_3, supporting the hypothesis that the relationship between the empirical and initial prior-consistency RMSEs is not linear in the validation step.

C. Case study three

Case study three utilizes the horizontal symmetry prior, and the results are shown in Table III. With the horizontal symmetry prior, using a grid search to optimize the loss hyperparameters (PGNN_1) improves the empirical and prior-consistency performance relative to the vanilla NN. Similarly, when the prior consistency is used in the equally weighted validation

TABLE I
CASE STUDY ONE: THE EMPIRICAL AND PRIOR-CONSISTENCY RMSES WHEN ONLY THE BOUNDARY PRIOR IS UTILIZED.

Model Type	Empirical	Boundary
NN	89.542 ± 18.335	179.882 ± 42.815
$PGNN_1$	87.792 ± 18.343	182.169 ± 40.237
$PGNN_2$	87.211 ± 26.567	176.318 ± 53.382
CoReg-$PGNN_1$	85.977 ± 18.745	171.095 ± 45.687
CoReg-$PGNN_2$	79.722 ± 23.746	160.583 ± 51.509
CoReg-$PGNN_3$	77.513 ± 22.972	161.582 ± 44.772
CoReg-$PGNN_4$	70.297 ± 18.561	149.319 ± 45.337

TABLE II
CASE STUDY TWO: THE EMPIRICAL AND PRIOR-CONSISTENCY RMSES WHEN ONLY THE INITIAL PRIOR IS UTILIZED.

Model Type	Empirical	Initial
NN	89.542 ± 18.335	119.310 ± 37.949
$PGNN_1$	95.601 ± 20.144	134.372 ± 61.739
$PGNN_2$	87.468 ± 18.194	111.923 ± 31.855
CoReg-$PGNN_1$	90.020 ± 18.075	109.105 ± 36.697
CoReg-$PGNN_2$	87.465 ± 17.307	121.636 ± 38.898
CoReg-$PGNN_3$	87.273 ± 21.073	126.276 ± 39.940
CoReg-$PGNN_4$	88.548 ± 18.036	114.681 ± 48.498

TABLE III
CASE STUDY THREE: THE EMPIRICAL AND PRIOR-CONSISTENCY RMSES WHEN ONLY THE HORIZONTAL SYMMETRY PRIOR IS UTILIZED.

Model Type	Empirical	Horizontal Symmetry
NN	89.542 ± 18.335	198.045 ± 15.521
$PGNN_1$	85.822 ± 18.940	194.362 ± 16.464
$PGNN_2$	84.930 ± 24.236	195.337 ± 15.357
CoReg-$PGNN_1$	85.057 ± 18.251	193.720 ± 15.662
CoReg-$PGNN_2$	85.479 ± 22.160	196.787 ± 17.147
CoReg-$PGNN_3$	85.550 ± 23.918	195.711 ± 21.033
CoReg-$PGNN_4$	84.388 ± 19.544	196.065 ± 17.367

TABLE IV
CASE STUDY FOUR: THE EMPIRICAL AND PRIOR-CONSISTENCY RMSES WHEN ONLY THE VERTICAL SYMMETRY PRIOR IS UTILIZED.

Model Type	Empirical	Vertical Symmetry
NN	89.542 ± 18.335	222.270 ± 14.602
$PGNN_1$	87.059 ± 17.981	221.033 ± 17.271
$PGNN_2$	79.469 ± 21.270	215.503 ± 16.301
CoReg-$PGNN_1$	82.986 ± 21.597	219.423 ± 18.876
CoReg-$PGNN_2$	88.379 ± 26.436	221.462 ± 18.891
CoReg-$PGNN_3$	86.182 ± 25.074	216.779 ± 15.800
CoReg-$PGNN_4$	82.951 ± 28.178	220.287 ± 17.300

PI ($PGNN_2$), the empirical and prior-consistency RMSEs are improved relative to the vanilla NN. $PGNN_2$ results in a larger empirical RMSE standard deviation relative to NN and $PGNN_1$. The larger empirical RMSE standard deviation indicates a larger variance in the performance of $PGNN_2$ relative to the NN and $PGNN_1$.

CoReg-$PGNN_1$-CoReg-$PGNN_4$ perform similarly to $PGNN_1$/$PGNN_2$ and improve the empirical and prior-consistency RMSE performance relative to NN. When we apply the proposed weighted correlation-based model validation in coReg-$PGNN_1$, the prior-consistency RMSE and the empirical RMSE standard deviation are improved relative to $PGNN_2$. Similar to $PGNN_2$, coReg-$PGNN_2$ and coReg-$PGNN_3$ have a large empirical RMSE standard deviation, supporting the benefit of using the proposed weighted validation PI (coReg-$PGNN_1$ and coReg-$PGNN_4$) instead of an equally weighted validation PI. The coReg-$PGNN_4$ model results in the smallest empirical RMSE and a smaller empirical RMSE standard deviation relative to $PGNN_2$.

D. Case study four

Case study four utilizes the vertical symmetry prior, and the results are shown in Table IV. With the vertical symmetry prior, using a grid search to optimize the loss hyperparameters ($PGNN_1$) minimally improves the empirical and prior-consistency performance relative to the vanilla NN. However, when the prior consistency is used in the equally weighted validation PI ($PGNN_2$), the empirical and prior-consistency RMSEs are significantly improved relative to the vanilla NN and $PGNN_1$.

CoReg-$PGNN_1$ and CoReg-$PGNN_4$ perform similarly to $PGNN_2$ and improve the empirical and prior-consistency RMSE performance relative to NN. However, $PGNN_2$ outperforms CoReg-$PGNN_1$ and CoReg-$PGNN_4$ with smaller empirical and prior consistency RMSEs and standard deviations. The proposed adaptive loss weighting metric in CoReg-$PGNN_2$ and CoReg-$PGNN_3$ perform similarly to the vanilla NN and $PGNN_1$, but worse than $PGNN_2$. Since $PGNN_2$ and CoReg-$PGNN_3$ have the same validation PI and only differ in the loss weighting method, we note that the proposed adaptive loss weighting metric degrades the empirical and prior-consistency RMSE performance. We discuss CoReg-$PGNN_1$-CoReg-$PGNN_4$ performance with the vertical symmetry in Subsection V.

E. Case study five

In case study five, we utilize the four priors from case studies 1-4 to test the performance of the proposed methods when multiple priors are competing in the loss and validation functions. Case study five results are provided in Table V. Using a grid search to optimize the loss hyperparameters ($PGNN_1$) and using the prior consistency in an equally weighted validation PI ($PGNN_2$), the empirical and prior consistency RMSEs and the standard deviations degrade or remain the same.

When we apply the proposed weighted correlation-based model validation in coReg-$PGNN_1$, the empirical and prior-consistency performance is improved relative to the NN and $PGNN_1$/$PGNN_2$. coReg-$PGNN_1$ performance relative to $PGNN_2$ shows the importance of taking into account the weight of each prior in the validation PI to select a more generalizable model.

When we use the proposed adaptive prior loss weighting metric and only the empirical error as the CV PI, coReg-$PGNN_2$ outperforms NN, $PGNN_1$, $PGNN_2$, and coReg-$PGNN_1$ empirical and prior-consistency performance-wise. coReg-$PGNN_2$ performance demonstrates the performance benefit of using the prior loss adaptive weighting metric for multi-prior problems instead of a grid search. coReg-$PGNN_2$ performance also shows that the prior loss weights significantly impact the empirical/prior-consistency performance compared to the prior error weights in the validation PI (coReg-$PGNN_1$).

Using an equally weighted CV PI, coReg-$PGNN_3$ results in the best empirical RMSE demonstrating the benefit of using the prior loss adaptive weighting metric. However, when we combine the proposed validation and loss weighting metrics, coReg-$PGNN_4$ performance degrades relative to coReg-$PGNN_3$. We hypothesize that the performance degradation in coReg-$PGNN_4$ is due to the incorrect assumption that the initial prior and the empirical RMSE have a linear relationship in the validation step.

V. Discussion

The boundary prior is more valuable empirical RMSE performance-wise than the initial and symmetry priors (Table I-IV). The boundary prior is hypothesized to be the most valuable prior since it provides the knowledge of 1000 uniformly distributed samples. The initial prior only provides knowledge of one sample and is, therefore, the least valuable prior empirical RMSE performance-wise. The symmetry priors are more valuable than the initial prior empirical RMSE performance-wise. However, even though each symmetry prior provides knowledge of 1000 new uniformly distributed samples, the boundary prior is more valuable RMSE performance-wise than the symmetry priors. The boundary samples are the complement of the training sample space and therefore provide new knowledge outside the training space. On the other hand, the symmetry prior samples and the training samples intersect and share the same space (i.e., interpolation), making the symmetry prior less valuable.

In our work and other applications, without further studies, the type of relation the empirical and prior-inconsistency loss/validation error variables have is often unknown. The weighting metrics proposed in this work utilize the Pearson's correlation coefficient to assess the strength and direction of the linear relationships between the empirical and prior-inconsistency error variables. Linearity is not a strict assumption of Pearson's correlation since the metric determines the linear association between the variables. However, Pearson's correlation coefficient would not be a suitable choice for applications where the variables are already known to have a nonlinear relation [22]. Possible solutions include transforming the variables to have a normal distribution and linearizing the relationship between the variables. Another approach is to utilize other correlation metrics, such as Spearman's rank correlation or distance correlation. For example, Spearman's rank correlation can examine nonlinear monotonic relationships and does not require normally distributed variables.

To achieve the best empirical and prior-consistency RMSE improvements for modeling problems with multiple priors we recommend utilizing CoReg-$PGNN_3$; the proposed correlation-based adaptive prior loss weighting metric in Eq. 22 and 23 along with the the equally weighted sum of the empirical and prior errors as the cross-validation performance index. For modeling problems with a single prior and a lack of knowledge of the value of said prior, we recommend utilizing CoReg-$PGNN_4$; the proposed correlation-based prior loss adaptive weighting (Eq. 22 and 23) and weighted correlation-based model validation (Eq. 24 and 25).

TABLE V
CASE STUDY FIVE: THE EMPIRICAL AND PRIOR-CONSISTENCY RMSES WHEN THE BOUNDARY, INITIAL, AND SYMMETRY PRIORS ARE ALL UTILIZED.

Model Type	Empirical	Boundary	Initial	Horizontal Symmetry	Vertical Symmetry
NN	89.542 ± 18.335	179.882 ± 42.815	119.310 ± 37.949	198.045 ± 15.521	222.270 ± 14.602
$PGNN_1$	89.927 ± 18.186	178.701 ± 42.539	118.455 ± 34.805	198.924 ± 16.246	223.779 ± 16.718
$PGNN_2$	89.776 ± 24.551	177.321 ± 52.114	139.850 ± 59.693	196.637 ± 17.046	226.110 ± 23.232
CoReg-$PGNN_1$	86.708 ± 19.495	174.100 ± 46.645	120.268 ± 31.440	195.247 ± 16.870	221.802 ± 17.691
CoReg-$PGNN_2$	80.407 ± 26.374	168.947 ± 55.029	94.736 ± 53.162	190.227 ± 15.731	213.354 ± 16.621
CoReg-$PGNN_3$	75.940 ± 19.536	162.956 ± 42.356	96.102 ± 31.963	190.454 ± 15.746	211.743 ± 12.558
CoReg-$PGNN_4$	81.573 ± 23.836	175.826 ± 49.291	86.954 ± 49.815	192.198 ± 15.998	214.107 ± 12.965

To achieve the best empirical and prior-consistency RMSE improvements for modeling problems with a boundary prior, we recommend utilizing CoReg-$PGNN_4$. To achieve the best empirical RMSE improvements with the smallest standard deviation for modeling problems with an initial prior, we recommend utilizing CoReg-$PGNN_2$. To achieve the best empirical RMSE improvements with the smallest standard deviation for modeling problems with a horizontal symmetry prior, we recommend using CoReg-$PGNN_4$. To achieve the best empirical and prior-consistency RMSE improvements for modeling problems with a vertical symmetry prior, we recommend using $PGNN_2$.

For modeling problems with multiple priors, the empirical/prior-consistency performance is affected more significantly by the prior loss weights than the prior error weights in the validation PI (Table V). We hypothesize that the prior loss weights significantly impact the model performance relative to the validation error weights because the loss search space is larger than the validation PI search space. Furthermore, combining multiple priors does not necessarily improve the empirical RMSE performance. The best empirical RMSE achieved in this work is when the proposed loss and validation error weighting methods (CoReg-$PGNN_4$) are utilized with just the boundary prior (Table I).

In case study two, we saw that coReg-$PGNN_1$ and coReg-$PGNN_4$ performed poorly with the initial prior due to the erroneous assumption that the empirical and initial prior validation errors have a linear relationship (Table II). Similarly, in case study five, while the proposed methods outperformed the vanilla NN and the PGNN models, we saw that coReg-$PGNN_4$ performance degrades relative to coReg-$PGNN_3$ and hypothesize that the performance degradation due to the erroneous assumption that the initial prior and the empirical RMSE have a linear relationship in the validation step. Any prior knowledge of a given problem will add value to a data-driven framework such as the PGNN. However, as evident by case studies two and five, the methods used to incorporate the priors into a NN training/validation framework might be detrimental to the PGNN empirical performance. Given the proposed methods and results, we conclude that for modeling problems with single or multiple priors, if the relationship between any of the prior errors and the empirical error is nonlinear or unknown, all the priors should be equally weighted in the validation performance index to select the most generalizable model.

VI. CONCLUSION

This work examines the impact of weighting single and multi-prior loss and validation error functions on five case studies using a simple benchmark modeling problem. We quantify and analyze benchmark NN and PGNN models to demonstrate the shortcomings of utilizing sensitivity studies to set the loss and validation error weights. To overcome these shortcomings, we propose (1) a multi-prior correlation-based adaptive weighting metric for choosing the prior-error weights during the learning step and (2) a multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step. The proposed metrics aim to balance the interplay between the competing empirical and prior errors in the loss function and the validation performance index function. The proposed weighting metrics are not limited to PGNN discussed in this work and apply to data-driven modeling frameworks with competing loss and validation errors. Our analysis and proposed methods provide new insights into competing priors' impact on training and validation PGNNs and offer an empirical RMSE performance improvement of up to 21.493%.

Future work includes (1) testing and utilizing the significance of the correlation coefficient, (2) exploring the use of more robust correlation measures, (3) analyzing the effects of the NN model structure (number of hidden layers and hidden units), and (4) analyzing the effects of the training data set size and the complexity of the modeling data.

ACKNOWLEDGMENT

This work was supported by The University of Sheffield and the UK EPSRC Future Manufacturing Hub - Manufacture using Advanced Powder Processes (MAPP) through grant Grant EP/P006566/1.

REFERENCES

[1] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2318–2331, 2017.

[2] C. M. Childs and N. R. Washburn, "Embedding domain knowledge for machine learning of complex material systems," *MRS Communications*, vol. 9, no. 3, pp. 806–820, 2019.

[3] A. Y. Sun, B. R. Scanlon, Z. Zhang, D. Walling, S. N. Bhanja, A. Mukherjee, and Z. Zhong, "Combining physically based modeling and deep learning for fusing grace satellite data: Can we learn from mismatch?" *Water Resources Research*, vol. 55, no. 2, pp. 1179–1195, 2019.

[4] S. Itani, M. Rossignol, F. Lecron, and P. Fortemps, "Towards interpretable machine learning models for diagnosis aid: A case study on attention deficit/hyperactivity disorder," *PloS one*, vol. 14, no. 4, p. e0215720, 2019.

[5] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (pgnn): An application in lake temperature modeling," *arXiv preprint arXiv:1710.11431*, 2017.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "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, pp. 686–707, 2019.

[7] T. de Wolff, H. Carrillo, L. Martí, and N. Sanchez-Pi, "Towards optimally weighted physics-informed neural networks in ocean modelling," *arXiv preprint arXiv:2106.08747*, 2021.

[8] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient pathologies in physics-informed neural networks," *arXiv preprint arXiv:2001.04536*, 2020.

[9] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, and V. Kumar, "Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 558–566.

[10] M. Elhamod, J. Bu, C. Singh, M. Redell, A. Ghosh, V. Podolskiy, W.-C. Lee, and A. Karpatne, "Cophy-pgnn: Learning physics-guided neural networks withcompeting loss functions for solving eigenvalue problems," *arXiv preprint arXiv:2007.01420*, 2020.

[11] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan, "Incorporating prior domain knowledge into deep neural networks," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 36–45.

[12] S. Mowlavi and S. Nabi, "Optimal control of pdes using physics-informed neural networks," *arXiv preprint arXiv:2111.09880*, 2021.

[13] D. Liu and Y. Wang, "Multi-fidelity physics-constrained neural network and its application in materials modeling," *Journal of Mechanical Design*, vol. 141, no. 12, 2019.

[14] S. A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture," *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113959, 2021.

[15] Z. Xiang, W. Peng, X. Zheng, X. Zhao, and W. Yao, "Self-adaptive loss balanced physics-informed neural networks for the incompressible navier-stokes equations," *arXiv preprint arXiv:2104.06217*, 2021.

[16] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 794–803.

[17] A. A. Heydari, C. A. Thompson, and A. Mehmood, "Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions," *arXiv preprint arXiv:1912.12355*, 2019.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] R. Bischof and M. Kraus, "Multi-objective loss balancing for physics-informed deep learning," *arXiv preprint arXiv:2110.09813*, 2021.

[20] C. Enăchescu, "Approximation capabilities of neural networks," *JNAIAM*, vol. 3, no. 3-4, pp. 221–230, 2008.

[21] J. C. Brigham and W. Aquino, "Surrogate-model accelerated random search algorithm for global optimization with applications to inverse material identification," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 45-48, pp. 4561–4576, 2007.

[22] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: appropriate use and interpretation," *Anesthesia & Analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

Chapter 4

Paper Three (application): In Situ porosity localization in laser additive manufacturing using photodiode sensor measurements: a prior-guided neural network approach

4.1 Paper Three Summary

Paper three is an application paper that utilizes the methods proposed in paper one (Chapter 2) for a classification problem surrounding in-situ porosity localization predictions in laser powder bed fusion (PBF). The proposed methods are assessed through the porosity localization percentage classification accuracy on high-strength aluminum A20X single-material cubes sintered using various process parameters via a randomized Box Behnken design with three factors and three replicates. Below is a summary of the paper's contributions.

- **In-situ detection and localization of porosities using a standardization-friendly MLP NN, photodiode data, and machine-independent physics-driven features.**
  - Thus far, melt pool imaging has been considered one of the most critical monitoring methods in PBF for porosity detection [43, 44]. This work shows that photodiode sensory data is sufficient for porosity detection and localization in metal PBF. Relying exclusively on photodiode sensory data for porosity detection and localization

simplifies the required hardware, data acquisition, data processing, and ML models.

- **A framework that given a three-dimensional sintering coordinate can make in-situ porosity localization predictions via (1) for the first time in literature, a local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe and within-layer sensory data and (2) a local 3-dimensional neighborhood of inter-layer sensory data.**
  - The benefit of utilizing within hatch stripe for in-situ porosity localization is quicker predictions without the need for surrounding hatch stripes or preceding/succeeding layer data. The capability to make porosity localization predictions exclusively within stripe data facilitates quicker within-layer defect mitigation via altering the sintering process settings of the surrounding hatch stripes. Utilizing within hatch stripe sensory data improves the classification accuracy of porosities with a spherical equivalent diameter (SED) $\leq$50 µm, relative to inter-layer data. The classification percentage improvements of using within-layer data relative to inter-layer data for porosities with an SED $\leq$50 is NN 18.14%, PGNN 19.62%, $PGNN_1$ 19.66%, and $PGNN_2$ 18.24%.
  - Utilizing local three-dimensional neighborhoods of inter-layer sensory data (multiple z layers) for porosity detection and localization enables capturing the laser spot overlap and melt pool remelting zone effects. Relative to using fewer PBF samples ($d < 100$), the benefit of utilizing a three-dimensional neighborhood of sensory data for in-situ porosity localization are: (1) an improved classification accuracy of porosities with an SED $\geq$50 µm, (2) improved the classification accuracy of the nominal samples and the false positive rates, and (3) facilitates inter-layer porosity mitigation via altering the succeeding layers.
- **For the first time in the literature, a framework based on prior-guided neural networks to utilize the often abundant SLM nominal data in the form of a prior training loss and a prior validation error.**
  - When exclusively using within-layer data, the structurally-optimized PGNN models reduce the number of hidden units and outperform

the NN and the non-structurally optimized PGNN models. The improved performance of the structurally-optimized PGNN models when using within hatch stripe data highlights the benefits of (1) optimizing the number of hidden units and (2) utilizing the abundant nominal (porosity-free) data in the form of a prior loss function and the performance index model validation.

  - The structurally-optimized PGNN models also consistently reduce the log loss error standard deviation, which increases the confidence in the model's classification accuracy.

- **To the author's knowledge, this is the first work in the literature capable of the localization of porosities as small as 38.118 µm SED, smaller than 47 µm SED in [1]. To the author's knowledge, the proposed methods localization of porosities as small as 38.118 µm SED is also a more than a five-fold improvement on the smallest SED porosity localization via photodiode sensory data [38].**

  - Using exclusively within hatch stripe data, we detect and localize porosities with a SED smaller than 50 µm with a classification accuracy ranging from $78.71 \pm 1.54\%$ to $80.67 \pm 2.67\%$.

  - We note that successfully detecting porosities smaller than 50 µm SED in-situ using exclusively within hatch-stripe data is a significant step towards within-layer defect mitigation via altering the sintering process settings of the surrounding hatch stripes.

The significance of this work in the field of metal powder bed fusion metal additive manufacturing includes the following findings:

- In metal additive manufacturing, controlling the porosity formation and microstructure is essential for dictating the sintered part properties. Porosity is one of the most significant concerns for part durability, where high cycle and low cycle fatigue (LCF) properties tend to be substandard compared to conventional manufacturing processes[45]. The current work in the literature is limited by the need for three-dimensional neighborhoods of sensory data from multiple sintered layers before making porosity predictions. Three-dimensional neighborhoods of data require collecting sensory data across several layers before making a porosity localization prediction which prohibits within-layer porosity mitigation.

Detecting and localizing porosity after multiple sintered layers have succeeded may hinder the possibility and quality of mitigating porosities. In this work, for the first time in the literature, we proposed a framework capable of making in-situ porosity localization prediction via exclusively local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe and within-layer sensory data. Our proposed framework of utilizing within hatch stripe and within-layer sensory data improves the classification accuracy of porosities with an SED $\leq$50 µm relative to the typical literature use of inter-layer data. The capability to make micro-porosity (SED $\leq$50 µm) localization predictions exclusively within stripe and layer data facilitates quicker within-layer defect mitigation via altering the sintering process settings of the surrounding hatch stripes. The capability of within-layer defect localization and mitigation can improve both the quality of AM parts and the AM compliance with the reliability and safety certification requirements of industries such as the aerospace industry [46].

- Our proposed framework of prior-guided neural networks makes use of the often abundant nominal data in the form of a prior loss, enabling the machine learning structure to be more compliant with the process physics. The proposed PGNN models consistently reduced the log loss error standard deviation, which increases the confidence in the model predictions. Data-driven models that are more compliant with the process physics and can make predictions with higher confidence improve AM compliance with the reliability and safety certification requirements.

- Thus far, melt pool imaging has been considered one of the most critical monitoring methods in PBF for porosity detection [43, 44]. This work shows that photodiode sensory data is sufficient for porosity detection and localization in metal PBF. Relying exclusively on photodiode sensory data for porosity detection and localization simplifies the required hardware and associated costs, data acquisition, data processing, and ML models.

- The proposed framework of using within hatch-stripe data reduces the costs associated with data storage since there is no need to store multi-dimensional and multi-layer sensory data.

Further research might explore whether ML frameworks could be standardized across materials and sintering machines for porosity detection and localization. The question is where the line should be drawn between ML

frameworks for generalizable applications of low-quantity/high-variety products and tailored high-quantity/low-variety products. This study shows that for a given sintered coordinate, the sintering processes parameters and sensor measurements from up to 0.15 mm in three-dimension can have relevant information on the quality (porosity-presence) of said coordinate. If the porosity status of a given coordinate is affected by at least three neighboring hatch stripes and the succeeding three layers, is online defect mitigation sufficiently possible? Or should porosity detection methods be primarily used for offline process optimization and post-processing quality documentation?

A limitation of this work is that the model's predictions granularity is discrete, and there is a need for a model per d samples considered. A future outlook is to develop a dynamic model that can update the probability per coordinate on a rolling basis as new data is produced.

4.2 Paper Three

In-situ porosity prediction in metal powder bed fusion additive manufacturing using spectral emissions: a prior-guided machine learning approach

Mohamed Atwya[1*] and George Panoutsos[1]

[1*]Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK.

*Corresponding author(s). E-mail(s): Matwya1@sheffield.ac.uk;
Contributing authors: g.panoutsos@sheffield.ac.uk ;

Abstract

Numerous efforts in the additive manufacturing literature have been made toward in-situ defect prediction for process control and optimization. However, the current work in the literature is limited by the need for multi-sensory data in appropriate resolution and scale to capture defects reliably and the need for systematic experimental and data-driven modeling validation to prove utility. For the first time in literature, we propose a data-driven neural network framework capable of in-situ microporosity ($<$ **50µm** spherical equivalent diameter) localization for laser powder bed fusion via an exclusively one-dimensional local neighborhood of spectral emissions sensory data (within hatch stripe). We further propose using prior-guided neural networks to utilize the often-abundant nominal data in the form of a prior loss, enabling the machine learning structure to comply more with process physics. The proposed methods are validated via rigorous experimental data sets of high-strength aluminum A205 parts, repeated k-fold cross-validation, and prior-guided validation. Using exclusively within hatch stripe data, we detect and localize porosity with a spherical equivalent diameter (SED) smaller than 50 **µm** with a classification accuracy ranging from **78.71 ± 1.54%** to **80.67 ± 2.67%**. This is the first work in the literature demonstrating in-situ localization of porosities as small as **38.12 µm** SED ($\mathbf{0.03x10^{-3}}$ $\mathbf{mm^3}$) and is more than a five-fold improvement on the smallest SED

porosity localization via spectral emissions sensory data in the literature. In-situ localizing micro-porosity using exclusively within hatch-stripe data is a significant step towards within-layer defect mitigation, advanced process feedback control, and compliance with the reliability certification requirements of industries such as the aerospace industry.

Keywords: Prior-guided neural network, Machine learning, Structure optimization, Metal laser powder bed fusion, Porosity

1 Introduction

Manufacturers must deliver new products to the market across shorter time intervals due to several reasons including globalization, the rapid introduction of new technologies, and shorter product life cycles Bellgran and Säfsten (2009); Chryssolouris (2013); Ishikura (2001). The increasing demand for rapid, flexible, and cost-effective manufacturing results in a driving force towards novel rapid manufacturing technologies such as AM Hendricks and Singhal (2008); Conner et al (2014). Academic and industrial investments in advancing the AM technology continue to rise with the aim of revolutionizing manufacturing.

There are currently over 20 recognized AM processes with different layer sintering methods Norazman and Hopkinson (2014). AM technology can generally be classified according to the raw material's state used by the process; liquid-based processes (e.g., Stereolithography), solid-based processes (e.g., Fused Deposition Modeling, Robocasting), and powder-based processes (e.g., Laser Sintering, High Speed Sintering). Following the American Society for Testing and Materials (ASTM) classification, single-step AM technology for metallic materials include sheet lamination, laser powder bed fusion (L-PBF), and directed energy deposition (DED) ASTM et al (2015).

At present, AM technologies generally encounter the same challenges regardless of the specific process employed. The main challenges include relatively high unit costs due to expensive raw materials and machinery, part-part and machine-machine repeatability, and reject rates for finished parts Conner et al (2014); Tofail et al (2017). Improving the cost-related AM limitations and ensuring repeatable sufficient AM part quality (mechanical properties and geometry), require improving both the underlying AM technology and the quality control framework of the process Gao et al (2015). In the literature, process modeling, monitoring, and control strategies have been deployed to overcome some of these drawbacks.

2 Literature Review

In metal additive manufacturing, controlling the porosity formation and microstructure is essential for dictating the sintered part mechanical properties. Porosity is one of the most significant concerns for part durability, where high cycle and low cycle fatigue (LCF) properties tend to be substandard compared to conventional manufacturing processesDebRoy et al (2018). In-situ monitoring and controlling the microstructure and defect formations in AM is necessary as it dictates the mechanical properties of the sintered parts. The microstructure and defect formation in AM processes is very sensitive to slight deviations in process parameters Clymer et al (2017). Therefore, AM processes have tight operating windows for different materials. Sintering outside the experimentally-validated operating windows is either unexplored or associated with defects such as lack of fusion, porosity, and key-holing Dass and Moridi (2019).

AM technologies such as DED of metal incorporate many aspects including 3-D part design, material selection, manufacturing, and quality evaluation. Laser based AM technologies typically form α' martensite microstructure as a result of fast cooling and therefore offer higher tensile strength and lower ductility. However, EBM manufactured parts tend to have lower tensile strength and higher ductility due to exhibiting $\alpha - \beta$ microstructure from the slower cooling in a vacuum atmosphere. Irrespective of the material, fine-grained microstructures (columnar, mix of columnar and equiaxed, and equiaxed) are usually observed in AM. Grains in mixed and equiaxed microstructures are typically on the length scale of a deposited layer. While columnar microstructures can epitaxially grow over several layers (in the build direction).

A single AM metal part can have several microstructural features and porosity types that dictate its mechanical properties. The microstructure and defects are formed in-situ and primarily depend on the process parameters and material used Galarraga et al (2016). For example, a higher laser scanning speed can lead to an increase of the HAZ, which can lead to lack of fusion porosities and the formation of larger grain sizes and more α generations Wang et al (2021); Ge et al (2019). It is possible to perform to use design of experiment (DOE) methods to estimate the process-property-performance (PPP) linkage and find the optimal parameter settings that minimize defects and achieve the desired microstructre and mechanical properties. However, AM is a multiple-input-multiple-output multi-physics process which makes it costly and time-consuming to rely on DOE alone. Additionally, the highly non-linear dynamics of AM results in a large uncertainty of the DOE surface response mapping in areas that are untested (i.e. extrapolation) Childs and Washburn (2019).

In Xia et al (2016) the authors highlight that the hatch spacing results in different heat-transfer behaviors and a larger hatch pacing reduces the peak temperature and temperature gradient of the molten liquid. The laser power, scanning speed, and hatch spacing, have a significant effect on the temperature

of the molten pool, the phase transitions, the thermal behavior, and defect formation Song et al (2012); Pei et al (2017).

To overcome the challenges associated with DOE, numerical and data-driven models can be utilized to provide insights and allow real-time process monitoring and control of the process parameters. Real-time process monitoring and control are desirable to achieve the desired microstructure, higher manufacturing repeatability, reduce defects, and potentially achieve functionally-graded AM capabilities. In the context of AM monitoring and control, there are three primary data collection and modeling themes in AM:

1. Thermal models and thermal data collection
2. Melt-pool geometry models
3. Numerical and data-driven process-property-performance linkage models

In the literature, data-driven models have been applied to the following parts of AM: design, in situ monitoring, and the process–property–performance linkage. Machine learning NN have been applied to estimate one of following properties at a time: dimensional accuracy, surface roughens, shrinkage, density, compressive strength, and porosity Sood et al (2010); Chen and Zhao (2015); Shen et al (2004); Garg et al (2014).

AM does not have any large open-source experimental data sets since it is expensive to collect training data and data labeling requires expertise knowledge Qi et al (2019). Utilizing ML NNs on physics-derived synthetic data sets without using domain knowledge to guide the ML learning process leads to poor extrapolation capabilities Popova et al (2017). If for example the ML algorithm happens to inaccurately extrapolate the process parameters in-situ due to data noise or outliers then the manufacturing process is likely to fail. Similarly, it would not be possible to use traditional ML algorithms in an offline setting to explore new materials, process parameters, and microstructures.

In Dass and Moridi (2019) the authors demonstrate AM operating windows for different materials across a plot of power feed rate versus linear heat input. Sintering outside the experimentally-validated operating windows is either unexplored or associated with defects such as lack of fusion, porosity, and key-holing. In Clymer et al (2017), the authors provide a theory-based operating window within the absorbed power versus beam scanning velocity space for 316L stainless steel to achieve the yield strength, density, etc. Beyond AM, other fields also function within nominal operating windows. Friction stir welding (FSW) is also bounded by operating windows to avoid weld flaws Kah et al (2015); Sattari et al (2012). For example, Fraser et al (2018) provides a theory-based and experimentally-validated (AA6061-T6 butt joints) process window for FSW. However, extrapolating beyond the operating windows via physics-consistent and interpretable ML models has the following impacts:

1. Knowledge discovery: exploring new materials, AM process parameters, and microstructures through reliable extrapolation and model interpretability.

2. Physics-consistency: within the process operating windows (i.e. training data), theory-guided ML models would perform more accurately than physics-based models and data-driven models.
3. Additive manufacturing capabilities: through knowledge discovery and physics-consistency, theory-guided ML models can be used in real-time for process control to facilitate achieving the desired microstructre, higher manufacturing repeatability, reduce porosities, and potentially achieve functionally-graded AM capabilities. Functionally-graded materials have inhomogeneous mechanical properties achieved through the microstructure undergoing a gradual change along some direction.

Notable prior-guided and data-driven Ml hybrid approaches for microstructure predictions include Jha et al (2018) and Popova et al (2017). Jha et al. Jha et al (2018) look at alloy design (rather than AM) and utilize machine learning (k-NN algorithm) to create meta-models from computationally-expensive CALPHAD solidification data (soft magnetic alloys). The combined ML-CALPHAD TGDM approach successfully captured relationships between the processing variables (composition, temperature, annealing time) and the alloy structure (mean radius and volume fraction).

Popova et al (2017) employed a prior-guided data-driven approach and used data from the SPPARKS kinetic Monte Carlo (kMC) simulation suite to develop a multivariate regression polynomial model that can estimate microstructure features as function of the AM process parameters. They use the chord length distribution (CLD) algorithm along with principal component analysis to quantify features and perform feature selection on the kMC generated microstructure data. The authors note that the normalized MAE for some of the predicted CLDs in the testing data set were significant (> 0.25 compared to roughly < 0.15 for other test data points). They discover through a visual plot of the training/testing data that the test cases with high MAE were outside the range of the training data set (extrapolation). The work in Popova et al (2017) was not validated against experimental data.

Fu et al. review the applications of ML to metal laser-based AM for defect detection Fu et al (2022). The authors point out the need for methods capable of dealing with varying sensor accuracy and spatio-temporal resolutions. They discuss that defect detection should focus on detecting and evaluating the defect impact rather than just detecting it, highlighting the importance of a complete process-property-performance modeling framework. They highlight the importance of feedback control and in-situ correction systems to mitigate the impact of the defects. In Lin et al (2022) the authors further argue that in order to achieve closed-loop control, it is necessary to employ a hybrid physics-guided data-driven model and give the example of physical-guided loss functions.

Guo et al (2022) discuss the challenges for physics-informed ML in metal AM. The challenges include limited data availability and a data imbalance where most data is considered compliant/defect-free. They bring attention to the challenges of data modality and the importance of developing data

preprocessing methods that can be run in an in-situ manner. Regarding data integrity, they highlight that AM sensor monitoring data is imperfect and can be corrupted with noise, missing data, and incorrect measurements.

In another review, Meng et al (2020) brings attention to the lack of research on uncertainty quantification in ML for AM. Uncertainty quantification and minimization for AM regression and classification process-property-performance linkage tasks are integral toward corrective feedback control and decision making.

In Gaja and Liou (2018) the authors can detect and differentiate between porosities and cracks on a per-layer basis using feature extraction on acoustic emission (AE) data. The porosities and cracks were generated by mixing Ti-6Al-4V powder with H13 tool steel powder in the LMD printing process. The data was collected from 15 mm long and 5 mm long deposits. The AE signal was recorded during sintering, and the feature extraction was performed at the end of the deposited layer. The data labels were acquired via unsupervised learning in Gaja and Liou (2017). The experiment resulted in 5 crack defects and 32 porosities. Using the 37 samples, the authors train a logistic regression model and a neural network classifier to predict whether the AE signals correspond to porosity or a crack defect. The models were validated on a second AE signal acquired under the same experimental conditions.

Liu et al. propose using machine learning regression with physics-informed features to separately predict the mean pore diameter, median pore diameter, max pore diameter, and pore spacing Liu et al (2021). For each regression task, for example, predicting the mean pore diameter, they categorize the porosity data into pass, flag, and fail clusters via domain knowledge. Per porosity cluster, they train each model individually for improved performance. Training models individually per porosity cluster is a means of dealing with the data imbalance in the skewed porosity size distribution. Although the authors use k-fold cross-validation, we raise the potential for their proposed framework to suffer over-fitting. Over-fitting can occur due to neglecting the information available in porosity-free data, splitting the regression outputs into one per model, and splitting each regression output into three sub-categories. Omitting the porosity-free data and splitting the data into several sub-categories leads to a tailored model that is less likely to be generalizable on new data. The authors propose a novel physics-informed pore generation explanation method. They visualize the relationship between the physics effects and porosity via physics-porosity correlation maps such as power-intensity versus maximum-pore-diameter. The pore generation explanation method provides porosity suppressing regions and porosity encouraging regions.

In Gobert et al (2018) the authors investigate the use of high resolution digital single-lens reflex (DSLR) cameras for defect detection in metal L-PBF. They collect multiple images where images one to three are captured following powder re-coating, and images four to eight are collected following the laser fusion step. Per three-dimensional neighborhood, they extract

multi-dimensional features that span multiple layers via 3D convolution filters. They classify neighborhoods as anomalous or nominal via linear SVM classification and ensemble classification schemes. Via cross-validation experiments, the results show an in situ defect detection accuracy greater than 80% for flaws with a spherical equivalent diameter (SED) larger than 47 µm. The authors also highlight the importance of assessing features through multiple layers to account for the melt pool remelting zone and accurately identify discontinuities.

In Jafari-Marandi et al (2019) the authors propose a cost-driven decision-making framework which uses in-situ melt pool images and considers the cost of the spatial distribution of microstructural defects. The proposed framework recommends correction actions based on the cost of the spatial distribution of the defects.

Okaro et al. use a Gaussian mixture model and an expectation-maximization algorithm in a semi-supervised framework Okaro et al (2019). Photodiode data is acquired during the L-PBF of 49 tensile test bars, and ultimate tensile strength tests (UTS) were then used to categorize each bar as faulty or acceptable. The authors apply feature extraction to the photodiode data and validate the models via 2-fold cross-validation. The results show a 77% classification success rate on the 'acceptable' parts. The semi-supervised approach utilizes the labeled UTS data but also leverages unlabeled data.

Tain et al. develop deep learning neural networks and a data fusion method to predict porosity in laser AM via melt pool thermal history data Tian et al (2021). The first model proposed is a convolutional NN (PyroNet) to correlate the in-situ pyrometry images with layer-wise porosity. A second long-term Recurrent convolutional Network (IRNet) is developed to correlate sequential thermal images with layer-wise porosity. The PyroNet and IRNet predictions are fused to improve the prediction accuracy of the layer-wise porosity. The proposed approach is validated via 6-fold cross-validation on Ti–6Al–4V thin-wall. The authors discuss and analyze the results per fold of the cross-validation. The author's method's are not tested on previously unseen test data set.

In Taherkhani et al (2022) the authors take on an unsupervised self-organizing map (SOM) learning method to detect defects from co-axial data emitted from the melt pool during L-PBF. They test the proposed methods on samples with different process parameters, intentional micro-voids, and normally occurring porosities. The predicted defects' position and size are compared to computed tomography scans via a volumetric segmentation method and a confusion matrix. The prediction algorithm achieved a 61% to 94% sensitivity rate and a 69% to 93% specificity rate. The authors note that the sensitivity and specificity depend highly on the process parameters. The SOM algorithm detected intentional defects ranging from 100 to 320 µm in diameter.

In Jayasinghe et al (2022) the authors apply unsupervised clustering methods on features extracted from photodiode data to classify the build density of

L-PBF sintered cubes with an accuracy of 93.54%. They also use supervised regression to predict build density with a RMS error of 3.65%.

In Snow et al (2022) the authors collect layer-wise imagery, multi-spectral emissions, and laser scan vector data of cylindrical L-PBF builds. X-ray computed tomography (XCT) and an automated defect recognition software were utilized to identify internal porosities. To create a balanced ML data set, a random nominal sample was selected for each flaw from the same part. Given a porosity/nominal coordinate, a $15 \times 15 \times 11$ local neighborhood of sensor data is extracted for each data sample. Each neighborhood corresponded to roughly 940 µm × 940 µm × 660 µm (11 layers in the z coordinate). Convolutional neural networks were trained to discriminate porosities from nominal through a 70% training and 30% validation data split. The proposed approach is tested on a previously unseen data set collected from an independent build. The authors' methods can detect porosities with a spherical equivalent diameter (SED) of 200 µm and larger with a 93.9 % classification accuracy.

We summarize the gaps in the literature of L-PBF data-driven in-situ porosity localization:

1. The limiting need for three-dimensional neighborhoods of sensory data from multiple sintered layers before making porosity predictions. Three-dimensional neighborhoods of data require collecting sensory data across several layers before making a porosity localization prediction which prohibits within-layer porosity detection and mitigation. Detecting and localizing porosity after multiple sintered layers have succeeded may hinder the possibility and quality of mitigating porosities.
2. Abundant information-rich porosity-free sensory data is often omitted to avoid an imbalanced data set.
3. Porosities are often simulated via mixing different powders and are larger than or equal to 100 µm SED. However, the porosity formation heat dynamics that occur by mixing powders are not necessarily the same porosity formation heat dynamics that occur when using a single powder. Gas porosities in L-PBF are typically smaller than 100 µm, and as small 10 µm SED King et al (2014); Choo et al (2019). Higher gas porosity counts result in crack initiation and propagation paths, resulting in a decrease of impact strength Girelli et al (2019); Kan et al (2022). Research should address the challenge of localizing naturally occurring micro-porosities (< 100µm).
4. The ML model validation in L-PBF porosity detection is often limited due to limited experimental data, poor training/testing/validation data splits, and the lack of use of methods such as repeated cross validation, multiple random model weight initialization, and prior-guided cross-validation.

We address the gaps discussed in metal AM porosity localization via proposing a framework with the following contributions:

1. A framework that given a three-dimensional sintering coordinate can make in-situ porosity localization predictions, for the first time in literature, via a local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe

and within-layer sensory data. The proposed framework also generalizes to making predictions via a local 3-dimensional neighborhood of inter-layer sensory data (multiple z layers).
2. A framework based on prior-guided neural networks to utilize the often abundant PBF nominal data to (1) improve the detection and localization of micro-porosities with an SED smaller than 50 µm and (2) increases confidence in the model predictions via reducing the classification log loss error standard deviation. The often abundant PBF nominal data is utilized in the form of a prior training loss and a prior validation error.
3. To the author's knowledge, this is the first work in the literature capable of the localization of naturally occurring porosities as small as 38.118 µm SED ($0.029x10^{-}3$ mm^3), smaller than 47 µm SED in Gobert et al (2018). To the author's knowledge, the proposed methods localization of porosities as small as 38.118 µm SED is also a more than a five-fold improvement on the smallest SED porosity localization via photodiode sensory data Snow et al (2022).

Our contributions are further supported via:

1. The proposed porosity detection and localization is done on the bulk core of the part and the border of the part which is often unused in the literature.
2. Decreasing the classification log loss error standard deviation via utilizing a portion of the excess porosity-free data in the form of prior loss function under the framework of prior-guided neural networks.
3. Thorough experimental validation and testing on single-material cubes sintered using various process parameters via a randomized Box Behnken design with 3 factors and 3 replicates.
4. Thorough model validation and testing via repeated k-fold cross-validation, prior-guided validation, and multiple random model weight initialization.

The remainder of the paper is organized as follows. Section 3 presents the problem formulation and methodology, including the data preprocessing, feature extraction, and machine learning. The Evaluation Section in 4 covers the machine learning model variants and the training, validation, and testing. In Section 5 we present the results and analysis. Finally, in the Conclusion Section 6 we discuss the proposed methods' findings, strengths, and limitations.

3 Methodology

3.1 Problem Formulation and Preliminaries

In this work we use the RenishawTM AM 500M PBF machine which uses a 500 W ytterbium fiber laser for sintering and the Renishaw InfiniAM spectral emissions systems for monitoring. The Renishaw InfiniAM spectral emissions systems includes three high precision co-axial single-channel photodiodes. The LaserVIEW system photodiode monitors the laser power. The MeltVIEW

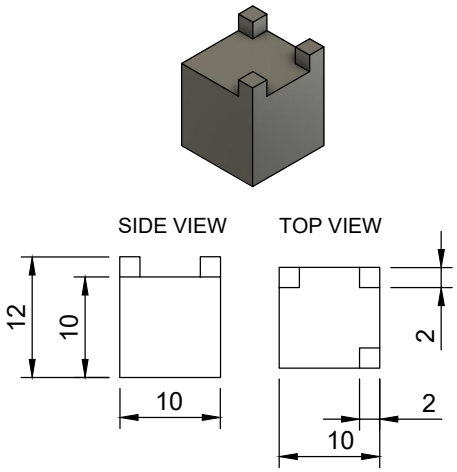
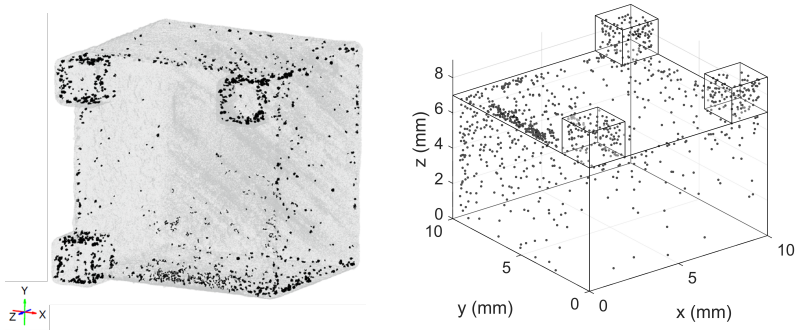


Fig. 1 The part design and dimensions (mm unit) examined in this work.



(a) (b)

Fig. 2 Porosity visualization of cube three: (a) The raw X-ray computed tomography data showing the porosity shape and volume and (b) the pre-processed dichotomous classification data 0, 1 data after removing 3.00 mm from the bottom of the cube along the z-plane to omit the bandsaw porosities/saw marks.

infrared light photodiode (1080 to 1700 nm) and the MeltVIEW visible light photodiode (700 to 1050 nm) monitor the melt-pool plume characteristics.

The part design is a cube with a 10.00 mm edge length and includes three smaller cubes at the top surface with a 2.00 edge length (Fig. 1). The three smaller cubes are geometric markers designed to facilitate alignment between the PBF and the XCT data.

A total of 45 cubes were sintered using high strength aluminum A205. The hatch spacing $\mathcal{H}$ and layer thickness $\mathcal{L}_{\mathrm{t}}$ are a constant 0.05 mm across the cubes. The border power, border speed, and hatch offset $\mathcal{H}_{\mathrm{o}}$ were varied following a randomized Box Behnken design with 3 factors and 3 replicates. The values utilized for the independent process parameter variables are:

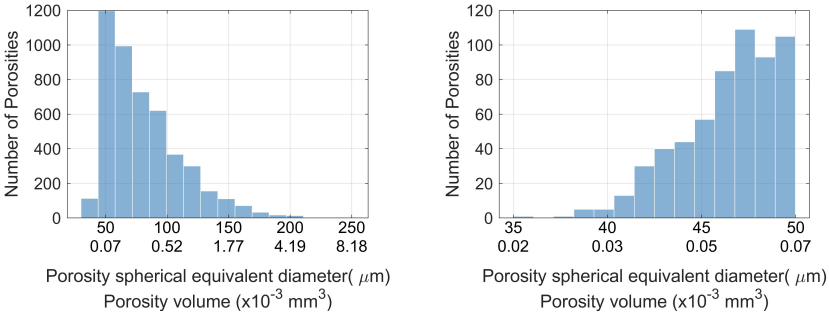


Fig. 3 The XCT porosity spherical equivalent diameter distribution across the 44 cubes in (a) and (b) a zoom in on the distribution of spherical equivalent diameters smaller than 50 µm.

1. Border Power: 100, 300, and 500 W
2. Border hatch Offset (a) : 0.00, 0.05, and 0.10 mm
3. Border Speed: 50, 175, 300 mm/s

3.2 X-ray computed tomography data

In Fig. 2 we see a visualization of the XCT porosities in cube three. The raw XCT data for each cube includes x coordinate, y coordinate, z coordinate, and porosity volume. The coordinates are in inches, and the porosity volume is in mm^3. We combine the text files into one matrix and label each sample with its respective cube index. The x, y, and z coordinates are converted from inches to mm. The porosity volume across the 44 cubes is right-skewed as seen in Fig. 3. Below is a list of the porosity volume statistics across the 44 cubes.

1. Number of porosities: 4733
2. Total porosity volume: 1.977 mm^3
3. Porosity volume mean: 4.177e-04 mm^3 (layer thickness: 500.00e-04 mm)
4. Porosity volume standard deviation: 6.099e-04 mm^3
5. Porosity volume median: 2.02e-04 mm^3
6. Porosity volume IQR [0.25, 0.75]: [0.094e-03, 0.475e-03] mm^3
7. Minimum porosity volume (SED): 23.00e-04 mm^3 (35.28 µm)
8. Maximum porosity volume(SED): 83.00e-04 mm^3 (251.00 µm)

After sintering, the cubes were sawed off the base printing plate via an automatic bandsaw. The automatic bandsaw removed approximately 2 mm from each cube (saw kerf). After sawing, the bandsaw leaves a diagonal pattern of porosities/saw marks on the base of the cubes. We remove 1 mm of the XCT data from the bottom of the cube along the z-pane to omit the saw marks. A total of 3.00 mm from the bottom of each cube along the z-plane are omitted from the study. Seven large artifact porosity samples from the XCT scans were

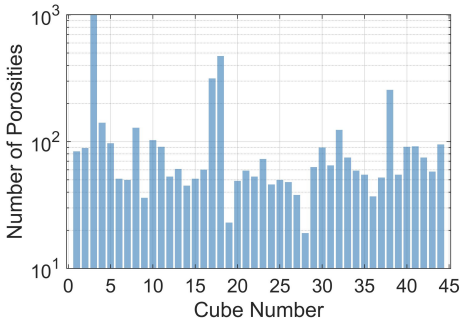


Fig. 4 The XCT porosity volume distribution per cube.

outside the cubes' edges in the three-dimensional scan images (cubes 30, 35, 37, and 38). The seven artifact samples were omitted from the study.

The zero position of the XCT data is in the cube's center with respect to the x, y, and z planes, excluding the three upper features. We shift the x, y, and z coordinates by adding [5, 5, 8] mm, such that x and y span [0, 10] and z spans [0, 9]. Due to the cube base unevenness and the non-smooth border surface, there are 1 or 2-dimensional geometric translational errors in the XCT centering of some cubes. Cubes with translational errors can be visually and computationally detected via searching for porosity locations outside the cube edge ranges of [0, 10] for the x and y plane and [0, 9] for the z plane. Note that the search for translational error in the XCT data must also separately consider the three smaller feature cubes. The cubes with translational error identified are cubes number 2, 3, 5, 10, 17, 18, 19, 20, 23, 30, 33, 37, and 40. Cubes with a translational error are shifted and centered around [5.0, 5.0, 4.5] along the x, y, and z planes. The number of porosities per cube is shown in Fig. 4.

3.3 PBF sensor data

The raw data acquired from the RenishawTM AM 500M machine and the Renishaw InfiniAM spectral emissions systems are 241 files containing the following features in columns: Time stamp in us, duration in us ($\underline{t}$), x and y coordinates of the laser in mm ($\underline{\acute{x}}$ and $\underline{\acute{y}}$), laserVIEW normalized photodiode laser power ($\underline{p}$), MeltVIEW photodiode intensities ($\underline{q}$ and $\underline{r}$). The 241 files correspond to the 241 layers sintered. The raw files are compiled into one matrix, and the z coordinate data is added as a feature ($\underline{\acute{z}}$). Each sample (row-wise) is assigned its corresponding cube number for reference.

We remove 3.00 mm of the machine/sensor data from the bottom of the cube along the z-pane ([0.00, 3.00] on the z-axis) to match the XCT. For ease of data readability, we shift each cube's x, y, and z coordinates, such that x and y span the [0.00, 10.00] mm range and z coordinates span the [0.00, 9.00] mm range. Note that the x and y coordinate data for layers 101 and 102 have

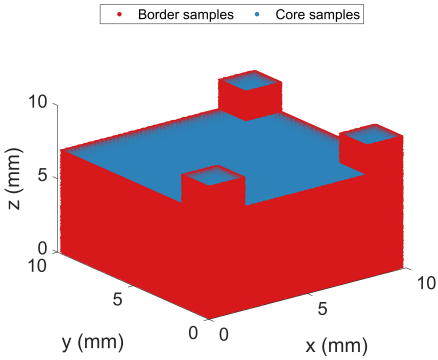


Fig. 5 A visualization of cube 5 showing the result of the border labeling procedure.

a geometry translation error and have been shifted to match the other layers. The geometry translation error is expected to be due to the acquisition system.

Due to the sensor acquisition rate, there are occasions where multiple consecutive samples are acquired at the same x and y coordinates. For example, if a given x and y coordinate is sintered for 40 us, two sensor readings spanning 20 us each could be acquired. Consecutive samples per cube per z-axis layer with the same x and y coordinates are combined into one sample. The duration feature is summed, and the three sensor data features are averaged. The cube index feature is also updated accordingly. After combining samples with repeated x and Y coordinates, the total number of samples is reduced by 1,758,037 samples from 267,867,956 to 266,109,919. The mean number of samples per cube is 5.953e+06 and the mean number of samples per 1.00 mm^3 is 8.222e+03.

3.4 Feature Addition

To improve the the models' prediction accuracy we compute/create additional features including a border label, hatch offset, scan speed, and the energy density.

3.4.1 Border Labeling

The core of the 45 cubes is sintered with the same parameters, but the border sintering parameters vary. It is necessary to have a border label for each sample to avoid a data imbalance when training the NN models. We clarify that border labels are only needed when training and would not be needed when running a trained model in-site. The border labeling procedure per PBF sample is discussed in Appendix A. In Fig. 5 we demonstrate the border labels for cube 5 across all the 180 layers.

3.4.2 Hatch Spacing

The hatch spacing feature, $\underline{w}$, in meters is created via the cube number and border label. The hatch spacing for the cube cores is set as 0.050 mm. The hatch spacing for the cube border is set as the hatch offset value, a, depending on the cube number. The border samples of cubes with a null hatch offset, $a = 0$, are assigned a hatch spacing of the single track border spot size 0.080 mm.

3.4.3 Scan Speed

The scan speed, $\underline{v}$, in meters per second is calculated via the duration and the x and y coordinates (Eq. 1).

$$\underline{v}_i = \frac{\sqrt{\Delta \acute{\underline{y}}_i^{\,2} + \Delta \acute{\underline{x}}_i^{\,2}}}{\underline{t}}, \qquad i \in \{1, \cdots, I-1\}, \tag{1}$$

where $\underline{v}$ is the scan speed (m/s) and $\underline{t}$ is the scan duration (s).

3.4.4 Energy Density

A unitless energy density, e, is calculated via the scan speed, hatch spacing, LaserVIEW system photodiode feature, and the layer thickness (Eq. 2). The energy density distribution across the core and border samples is shown in Fig. 6 and clarifies why border labeling is important to avoid a data imbalance when training NN models. The assumptions made in the calculation of the energy density via Eq. 2 are covered in Prashanth et al (2017). Although used widely in the literature, the energy density calculation in Eq. 2 may be miss-representative when used for PBF process parameter optimization. In this work, the assumptions made for the energy density calculation in Eq. 2 are less relevant in terms of porosity detection and localization. Albeit a more accurate energy density measure may improve porosity detection and localization.

$$\underline{e}_i = \begin{cases} 0 \;\; for \;\; \underline{v}_i = 0 \\ \frac{\underline{p}_i}{b\,\underline{v}_i\,\underline{w}_i} \;\; for \;\; \underline{v}_i \neq 0 \end{cases} i \in \{1, \cdots, I\} \tag{2}$$

where $\underline{e}$ is a unitless energy density.

3.5 Porosity-free Coordinates

The PBF data is imbalanced with most of the samples being porosity-free. To create a balanced Bernoulli distributed classification data set, we need to identify 4733 porosity-free coordinates in the 45 cubes to match the $C = 4733$ XCT porosity samples. Additionally, we note the data imbalance in the porosity locations where most of the porosities are in the cube core relative to the cube borders. The porosity location imbalance is significant since the core samples have the same sintering settings while the border sintering settings vary across cubes. For a balanced data set, it is necessary to ensure an equal

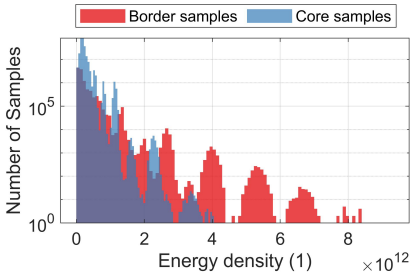


Fig. 6 The unitless energy density distribution across the core and border samples.

number of core and border samples between the porosity and porosity-free coordinates.

We partition the PBF data of each cube into 0.25 mm^3 blocks. A block is identified and handled via its center coordinates. We use the XCT porosity locations $[\breve{x}, \breve{y}, \breve{z}]$ to identify the porosity and porosity-free blocks. The border label $\underline{u}$ is used to identify the border blocks relative to core blocks. According to the 0.25 mm^3 block boundaries, the XCT samples can be divided into 3304 core and 1429 border porosities. A block is considered a border block if it contains one or more border samples according to $\underline{u}$. Similarly, according to the 0.25 mm^3 block boundaries, we identify 4050 and 4212 core and border block coordinates that are porosity-free across all 45 cubes. Therefore, there are 182250 and 189540 porosity-free core and border block coordinates and cube number combinations.

We randomly pick 3304 core and 1429 border porosity-free block coordinates and cube numbers to represent our porosity-free data. We define matrix $\boldsymbol{B}$ including the porosity-free block coordinates and cube numbers and the XCT coordinates and cube numbers. The matrix $\boldsymbol{B}$, therefore, has 9466 rows and 4 columns containing the x, y, and z coordinates and cube numbers. We randomly pick an additional 3304 core and 1429 border porosity-free block coordinates and cube numbers (matrix $\boldsymbol{C}$) to represent a prior data-set. The prior data set use is discussed in subsection 3.7. When populating the prior data set $\boldsymbol{C}$, we ensure that non of the samples in $\boldsymbol{B}$ are repeated in $\boldsymbol{C}$ to avoid data leakage. For matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ we define vectors $\underline{y} \in 0, 1$ and $\underline{z} \in 0, 1$ where porosities are assigned value 1 and porosity-free coordinates are assigned 0.

Ideally, a balanced data set would have an equal amount of porosity and porosity-free samples per combination of process parameters. As seen in Fig. 4 the porosity count distribution varies between the cubes, and some process parameter combinations result in more XCT porosity samples. Repeating the experiments several times to have an approximately equal number of porosity samples per combination of process parameters is time costly and requires expert knowledge. However, we note the choice taken here to have an approximately equal number of porosity-free samples across the cubes to improve the data imbalance.

3.6 Feature Extraction

Feature extraction is a process of dimensionality reduction which reduces a set of raw or pre-processed data to a more manageable groups for processing and modeling tasks. The objective of this subsection is to determine which PBF samples should be used to represent the 9466 porosity and porosity-free coordinates and cube numbers ($\boldsymbol{B}$). The number of PBF samples is $I =$ 266,109,919. Given that the largest porosity volume is 83.00e-04 mm^3 and the layer thickness and hatch spacing are 500.00e-04 mm, the largest porosity is smaller than the Renishaw PBF data resolution. Assuming that each sample in $\boldsymbol{B}$ corresponds to the nearest PBF sample in terms of the x, y, and z three-dimensional euclidean distance creates two challenges. The first challenge is that it would be challenging to correctly detect the PBF sample closest to the XCT sample due to geometric and dimensional errors when mapping the XCT scans on the PBF data. The second challenge is that the sintered part properties and presence of porosity at a specific coordinate are affected by the surrounding hatch paths and the preceding and succeeding layers due to the laser spot overlapping and the melt pool remelting zone (Section 1).

Instead of using the pre-processed PBF data set ($I = 266{,}109{,}919$ samples), we propose a framework where we use feature extraction measures on the d closest PBF samples to each XCT sample in terms of the three-dimensional euclidean distance. In Eq. 3 we define the matrix $\boldsymbol{A}_{ic}$ the three-dimensional euclidean distance between each XCT sample and every PBF sample. We define $h(\boldsymbol{A}_{ic}, d)$ where $c \in \{1, \cdots, 2C\}$ $\forall i \in \{1, \cdots, I\}$ as the smallest d elements in each column $\boldsymbol{A}_{ic}$. From $h(\boldsymbol{A}_{ic}, d)$ we find the d closest PBF samples to each XCT sample in terms of the euclidean distance $\boldsymbol{A}_{ic}$. Accordingly, for each sample in $\boldsymbol{B}$ we have d representative PBF samples. The same procedure is repeated for the prior matrix $\boldsymbol{C}$ porosity-free block coordinates and cube numbers. We note that utilizing the a pre-defined three-dimensional sized neighborhood of PBF samples is flawed in terms of a ML framework as it would lead to a different number of PBF samples per neighborhood due to the scan speed and hatch spacing/offset varying between cubes.

$$\begin{aligned} &\boldsymbol{A}_{ic} = \sqrt{(\underset{\sim}{x}_i - \boldsymbol{B}_{c1})^2 + (\underset{\sim}{y}_i - \boldsymbol{B}_{c2})^2 + (\underset{\sim}{z}_i - \boldsymbol{B}_{c3})^2}, \\ &i \in \{1, \cdots, I\}, c \in \{1, \cdots, 2C\}, \end{aligned} \tag{3}$$

where $\boldsymbol{A}_{ic}$ is the three-dimensional euclidean distance between each PBF and XCT sample x, y, and z coordinates.

We choose to set the three spectral emission sensors, scan speed, hatch spacing, and energy density as the machine learning input variables of interest. For each d representative PBF sample coordinate per coordinate in $\boldsymbol{B}$ and $\boldsymbol{C}$ we utilize the mean, variance and skewness as feature extraction methods on the spectral emission sensors, scan speed, hatch spacing, and energy density. The supervised machine learning input data set consists of (1) the mean, variance, and skewness of of the three spectral emission sensors ($\underset{\sim}{p}$, $\underset{\sim}{q}$, and $\underset{\sim}{r}$) and (2) the

mean and variance of the scan speed, hatch spacing, and energy density ($\underline{v}$, $\underline{w}$, and $\underline{e}$).

3.7 Supervised Machine Learning

The input (extracted features) and target labeled data presents a Bernoulli distributed classification problem with a nominal dichotomous target $0, 1$ where 0 indicates porosity-free and 1 indicates a porosity. To model the non-linear relationship between the input and target we use the universal approximator non-linear in the parameter two-layer feed-forward multi-layer perceptron (MLP) model. The MLP in this work consists of 15 input neurons, one hidden layer, and an output layer with one neuron. We use non-linear rectified linear unit activation function for the hidden layer and a logistic activation function for the output layer.

The NN cost function in this work consists of an empirical loss function $\mathcal{L}_{\mathrm{e}}$ and a regularization loss function $\mathcal{L}_{\mathrm{r}}$ (Eq. 4). To utilize the prior dataset, we use a prior-knowledge-guided neural network as in Atwya and Panoutsos (2022); Liu and Wang (2019); Muralidhar et al (2018); Jagtap et al (2020). The class of prior-knowledge-guided neural networks in Atwya and Panoutsos (2022); Liu and Wang (2019); Muralidhar et al (2018); Jagtap et al (2020) incorporate prior-based loss functions to the cost function of a vanilla NN. We add the prior knowledge-based loss function $\mathcal{L}_{\mathrm{p}}$ to the cost function in Eq. 5. We direct the reader to Atwya and Panoutsos (2019) for more information on using feature extraction for multi-layer perceptron classification. Paper Atwya and Panoutsos (2022) covers the structure optimization of prior-knowledge guided neural networks and the impact of data complexity, hyperparameters, and the data set size.

The framework utilized leads to three hyper-parameters: regularization loss weight ($\rho_{\mathrm{r}} \geq 0$), prior-based loss weight ($\rho_{\mathrm{p}} \geq 0$), and the hidden unit number (Eq. 4 and 5). The work in Atwya and Panoutsos (2022) shows that structure-optimized prior-knowledge-guided neural networks are not affected by non-adaptive regularization and prior-based loss weights. In this work, we utilize the structure-optimization methods in Atwya and Panoutsos (2022) and set the regularization and prior-based loss weights as constants. We discuss the model variants tested in Section 4.

$$\mathcal{L}_{NN} = \mathcal{L}_{\mathrm{e}} + \rho_{\mathrm{r}}\mathcal{L}_{\mathrm{r}} \tag{4}$$

where $\mathcal{L}_{\mathrm{e}}$ is the empirical loss , $\mathcal{L}_{\mathrm{r}}$ is the regularization loss, and ρ_{r} is the regularization loss weight.

$$\mathcal{L}_{PGNN} = \mathcal{L}_{\mathrm{e}} + \rho_{\mathrm{r}}\mathcal{L}_{\mathrm{r}} + \rho_{\mathrm{p}}\mathcal{L}_{\mathrm{p}}, \tag{5}$$

where $\mathcal{L}_{\mathrm{p}}$ is the prior loss and ρ_{p} is the prior-based loss weight.

4 Evaluation

As explained in Subsection 3.6 we apply feature extraction on the d closest PBF samples to each XCT sample in terms of the three-dimensional euclidean distance. To examine the impact of d on the classification performance we train and test each model variant on $d = [5, 10, 20, 40, 60, 80, 100]$. The three-dimensional euclidean distance average and standard deviation of each value in d is shown in Fig. 7.

We note that $d = 5$ represents a framework where in-situ porosity localization has been achieved via exclusively one-dimensional hatch stripe data and two-dimensional surrounding hatch stripes (laser spot overlap). The implementation of $d = 5$, does not utilize a three-dimensional neighborhood of data (i.e. does not make use of preceding/succeeding layers). Using $d \geq 10$ utilizes a three-dimensional neighborhood of sensory data to in-situ capture the inter-layer dynamics and the effects of the surrounding hatch stripes (laser spot overlap) and preceding/succeeding layers (melt pool remelting zone).

We split each data set in d where half the data set is used to train and validate the models and the second half is used to test the models. The second half of the data is referred to as the out-of-sample-data is reserved and only used to test the model performance after training and validation. Before splitting the data, the input and target row order is randomized.The minimum and maximum porosity volumes in the training/validation and testing data splits are provided below. Note that the minimum porosity volumes provided below are for reference and exclude the porosity-free samples.

- Training/validation data minimum porosity volume (SED): 23.00e-04 mm^3 (35.28 µm)
- Training/validation data maximum porosity volume (SED): 62.72e-04 mm^3 (228.81 µm)
- Testing data minimum porosity volume (SED): 29.00e-04 mm^3 (38.12 µm)
- Testing data maximum porosity volume (SED): 82.80e-04 mm^3 (251.00 µm)

Below is a list of the model variants utilized. The number of hidden units optimization and use of the prior error in the validation performance index of $PGNN_1$ and $PGNN_2$ is performed as in Atwya and Panoutsos (2022). The number of hidden unit coarse search was performed across a linearly spaced vector of 8 values within $[30, 100]$. The coarse search is followed by a fine search with a vector of 10 linearly spaced points around the best-estimated number of hidden units value $m^* \pm 10$. We use the same model training, validation, and testing framework as in Atwya and Panoutsos (2022).

- NN: A vanilla NN with 100 hidden units (no prior usage in the loss or validation performance index).
- PGNN: A PGNN with 100 hidden units (no prior usage in the validation performance index).
- $PGNN_1$: A PGNN where the number of hidden units is optimized via a line search (no prior usage in the validation function).

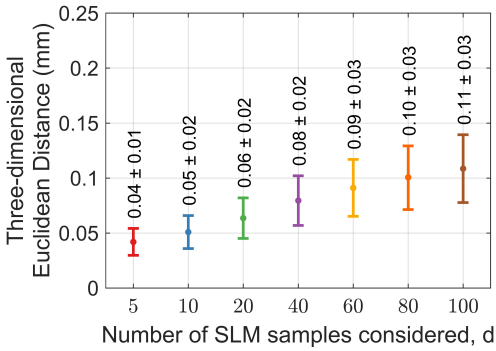


Fig. 7 The number of PBF samples considered (d) for feature extraction per porosity and porosity-free coordinate and the equivalent average three-dimensional euclidean distance of the PBF samples considered.

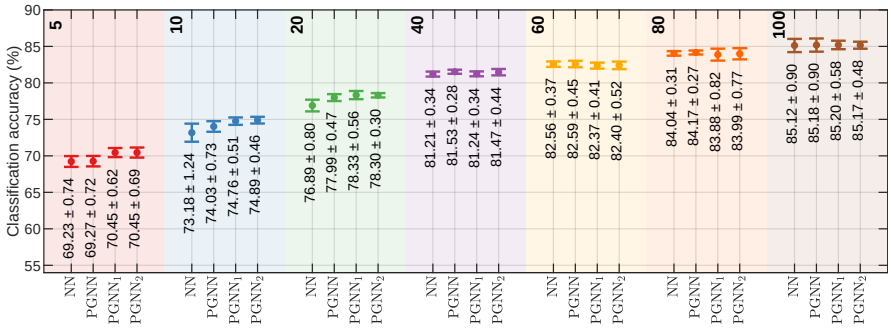


Fig. 8 The classification accuracy mean and standard deviation of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

- PGNN_2: A PGNN where the number of hidden units is optimized via a line search and the prior error is utilized in the validation performance index.

4.1 Model Training

The empirical $\mathcal{L}_e$ and prior $\mathcal{L}_p$ loss were computed via the cross-entropy cost function (Eq. 6 and 7. The regularization loss $\mathcal{L}_r$ was computed via L2 regularization (Eq. 8). The regularization weight was set as $1e-3$ for the NN model. The regularization and prior weights were set as $1e-3$ and $1e-3$ for the PGNN models. Each input feature was transformed via z-score standardization (i.e. zero mean and unit standard deviation). The NN and PGNN model weights were initialized via the He initialization method He et al (2015). The loss functions in Eq. 5 and 4 were minimized via the scaled conjugate gradient optimization algorithm (SCG) to find the optimal model weights. The SCG algorithm was allowed a maximum of 1000 iterations to find the solution.

$$\mathcal{L}_{\mathrm{e}} = -\sum_{i=1}^{I} \left(\underline{\hat{y}}_i \ln \left(\underline{y}_i \right) + \left(1 - \underline{\hat{y}}_i \right) \ln \left(1 - \underline{y}_i \right) \right), \tag{6}$$

where I is the number of data points and $\hat{y}_i$ and y_i are the prediction and target for the i^{th} input vector.

$$\mathcal{L}_{\mathrm{p}} = -\sum_{i=1}^{I} \left(\underline{\hat{z}}_i \ln \left(\underline{z}_i \right) + \left(1 - \underline{\hat{z}}_i \right) \ln \left(1 - \underline{z}_i \right) \right), \tag{7}$$

where I is the number of data points and $\hat{z}_i$ and z_i are the prediction and target for the i^{th} input vector of the prior data set.

$$\mathcal{L}_{\mathrm{r}} = \frac{1}{2} \underline{w}^T \underline{w}, \tag{8}$$

where $\underline{w}$ is a vector of the model weights.

4.2 Model Validation and Testing

To validate the models, we use three-time-repeated 5-fold cross-validation (CV). The 5-fold CV is repeated three times for every randomly initialized set of weights using re-divided data subsets. The empirical and prior-based CV errors were measured via the cross-entropy function. The mean and standard deviation of the input features in the training data set were used to normalize (z-score standardization) the input features in the testing and prior data sets. Following training and validation, the models' classification performance is tested on the OOS data set. Note that the OOS data sets were not used in the training/validation steps.

5 Results and Analysis

Across the 16 random model weight random initializations per model, we assess the mean and standard deviation of the classification accuracy, log loss error, and the area under the curve. We apply Youden's Index cutoff threshold to the model predictions providing a trade-off between the hit and false-alarm rates Atwya and Panoutsos (2019). Using Youden's Index cutoff threshold, we report the median and interquartile ranges for the true positive count, true negative count, false positive count, false negative count, precision, recall, and F-score. We note that Youden's Index cutoff threshold is the decision-making step where a decision is made on the porosity status of a sintered coordinate based on the classifier's predicted probability. Utilizing the standard Youden's Index cutoff threshold is also a decision that sensitivity and specificity are equally important when making classifications.

In Fig. 8 we see the model variants classification accuracy across the varying number of PBF samples d per ML sample. As the number of PBF samples compressed increases the classification accuracy increases of each model variant

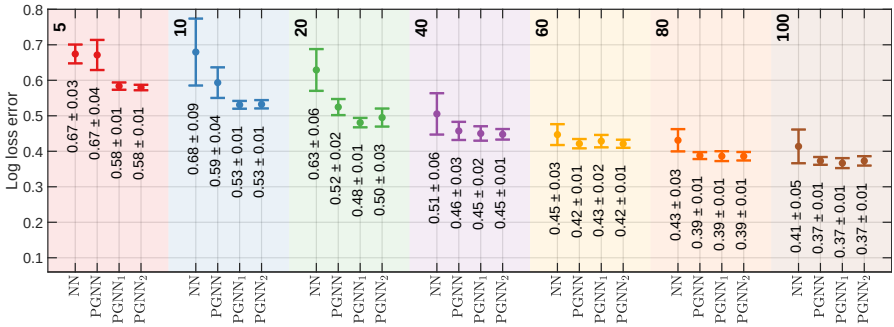


Fig. 9 The log loss error mean and standard deviation of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

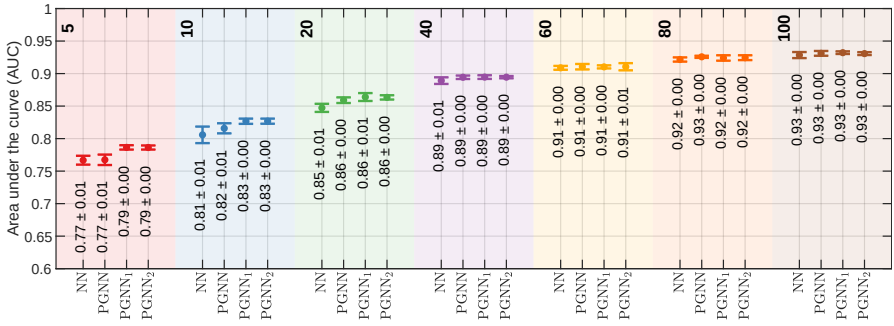


Fig. 10 The area under the curve mean and standard deviation of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

increases from a lowest accuracy of 69.23±0.74% to 85.20±0.58%. An increase in the number of compressed PBF samples d improves the classification accuracy due to:

1. An increase in the information captured per ML sample via feature extraction.
2. The model's are less affected by the geometric and dimensional errors when mapping the XCT scans on the PBF data.
3. The model's capture the effects of the surrounding hatch paths and the preceding and succeeding layers (i.e., the laser spot overlapping and the melt pool remelting zone).

The PGNN variants, specifically PGNN_2, outperform the vanilla NN by more than 1% in the mean classification accuracy and reduce the standard deviation when d is equal to 5, 10, and 20. For $d = 40$ and larger values of d, the model variants perform similarly in terms of the mean classification accuracy. The drop in classification accuracy improvements for $d = 40$ and larger values

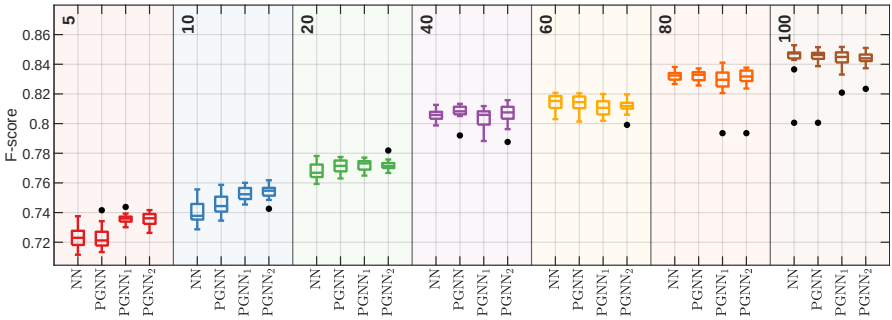


Fig. 11 The F-score box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample. The box plot shows the median, interquartile range, minimum, and maximum values. Outliers are marked as black dot, and an outlier is a value that is more than 1.5 times the interquartile range away from the bottom or top of the box.

of d is consistent with the results in Atwya and Panoutsos (2022) which show that as the data set size (or data information) increases, the impact on the empirical errors reduces.

The NN/PGNN models output the probability of an instance belonging to one of the two classes. The log loss score falls in the range $[0, \text{inf}]$ and measures how close the models output is to the corresponding true value. A smaller the log loss score corresponds to the model predicted probability being closer to the true class. The model variants log loss score across the varying number of PBF samples d per ML sample is shown in Fig. 9. The log loss score results are consistent with the classification results. One key feature in the log loss score results is the reduction in the standard deviation of the PGNN variants relative to the NN model across all values of d. The smaller log loss score standard deviation of the PGNN variants provides a higher confidence in the models' predictions.

The AUC value falls in the range $[0.5, 1]$ and is a measure of a classifier's ability to distinguish between classes. The higher the AUC, the better the model's performance at distinguishing between the two classes. The model variants AUC across the varying number of PBF samples d per ML sample is shown in Fig. 10. The AUC results are consistent with the classification results.

The F-score $[0, 1]$ is defined as the harmonic mean of the model's precision and recall. In this study, we use the standard F-score where the precision and recall are equally weighted. An F-score of 1 implies a perfect model.The lowest F-score median and interquartile ranges is 0.72 [0.713, 0.742] at $d = 5$. The largest F-score median and interquartile ranges is 0.847 [0.800, 0.853] at $d = 100$. As our data set is balanced, the F-score results and findings are in agreement with the classification accuracy, log-loss, and AUC results.

The precision results shown in Fig. 12 show which proportion of positive identifications was actually correct. The recall results shown in Fig. 13 show proportion of actual positives was identified correctly. Improving precision

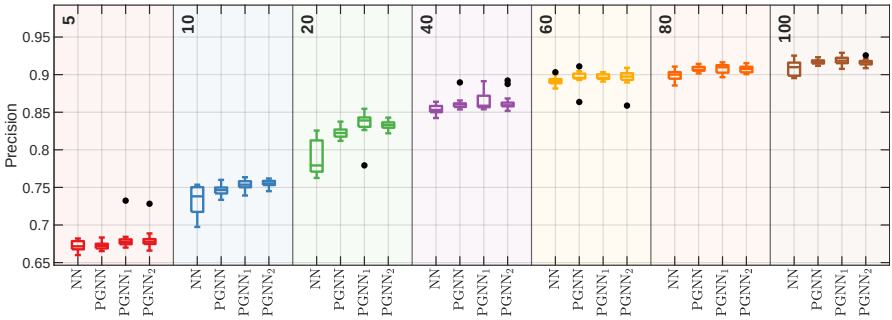


Fig. 12 The precision box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

typically reduces recall and vice versa. The precision and recall results show that while $d = 5$ results in an increase in the proportion of actual positives identified, a larger d value significantly improves the proportion of positive identifications that was actually correct. In other words, $d = 5$ will detect more true porosities, but will often miss-classify porosity-free coordinates as a porosity. The poor precision at $d = 5$ is hypothesized to be the model's potentially picking up on what might have been a porosity, but was corrected via the effects of the surrounding hatch paths and the preceding and succeeding layers. Similarly, the improved precision as d increases is likely due to the ML samples being more informative and capturing the effects of of the surrounding hatch stripes and the preceding and succeeding layers. The true and false positive/negative results are provided in Appendix B.

A box plot of the optimal number of hidden units selected by PGNN_1 and PGNN_2 is shown in Fig. 14. The NN and PGNN models had 100 hidden units. In comparison, PGNN_1 and PGNN_2 reduce the number of hidden units to the median range $[20, 65]$ and perform similarly or outperform the NN and PGNN models. A smaller median number and quantile range of hidden units is selected for $d \leq 40$. At $d \geq 40$, we see a larger median and quantile range in the selected number of hidden units. Although we see higher quantile ranges at $d \geq 40$, PGNN_1 and PGNN_2 reduced the classification log loss error standard deviation and increased the confidence in the predicted probabilities. By optimizing and reducing the number of hidden units and incorporating the prior-consistency error in the validation performance index PGNN_1 and PGNN_2 (1) avoid overfitting, (2) reduce the classification log loss error standard deviation for all d values, and (3) improve the classification accuracy for $d = \{5, 10, 20\}$.

Table 1 shows the classification accuracy average and standard deviation of the testing data across the porosity spherical equivalent diameter using within-layer data ($d = 5$).Table 2 shows the classification accuracy average and standard deviation of the testing data the varying porosity spherical equivalent diameter using inter-layer data ($d = 100$).

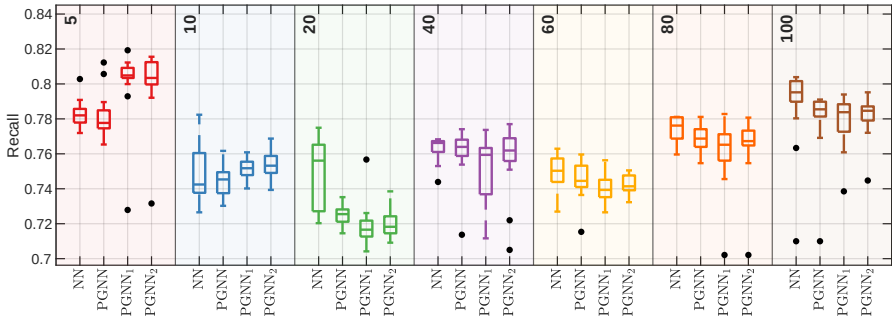


Fig. 13 The recall box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

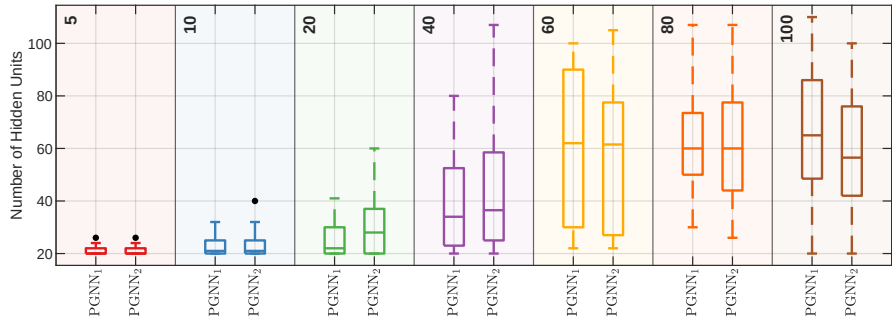


Fig. 14 A box plot of optimal number of hidden units selected by PGNN_1 and PGNN_2 across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

From Table 1, using exclusively within-layer data ($d = 5$) to detect and localize porosities with an SED smaller than 50 µm achieves the following accuracy: NN $78.71 \pm 1.54\%$, PGNN $78.88 \pm 2.25\%$, PGNN_1 $80.67 \pm 2.67\%$, and PGNN_2 $80.55 \pm 3.11\%$. From Table 2, using three-dimensional neighborhoods of inter-layer data ($d = 100$) to detect and localize porosities with an SED smaller than 50 µm achieves the following accuracy: NN $67.98 \pm 4.34\%$, PGNN $66.57 \pm 3.86\%$, PGNN_1 $66.47 \pm 2.69\%$, and PGNN_2 $67.32 \pm 2.68\%$.

From Table 1, using exclusively within-layer data ($d = 5$) to detect and localize porosities with an SED smaller than 50 µm achieves the following accuracy: NN $75.67 \pm 0.90\%$, PGNN $75.54 \pm 1.10\%$, PGNN_1 $77.73 \pm 1.76\%$, and PGNN_2 $77.65 \pm 1.52\%$. From Table 2, using three-dimensional neighborhoods of inter-layer data ($d = 100$) to detect and localize porosities with an SED larger than 50 µm achieves the following accuracy: NN $80.31 \pm 2.07\%$, PGNN $79.63 \pm 1.71\%$, PGNN_1 $79.54 \pm 1.35\%$, and PGNN_2 $79.60 \pm 0.98\%$.

Quantitatively, when exclusively utilizing within-layer data ($d = 5$) the NN and PGNN variants perform better for porosities with an SED smaller

than 50 µm and larger than 150 µm (Table 1). We hypothesize that porosities with a SED $\leq$50 µm occur primarily due to within hatch stripe disturbances and are less likely to form due to surrounding hatch stripes and the preceding/succeeding layers. Hence, utilizing the within-layer and within-stripe data ($d = 5$) is hypothesized to perform better on porosities with a SED $\leq$50 µm. The classification accuracy performance drop between 50 and 150 µm SED utilizing the within-layer and within-stripe data ($d = 5$) is hypothesized to be due to the lack of sensory information on the melt pool re-melting zone and the laser spot overlap. Therefore, porosities in the $[50, 150]$ µm SED range likely occur due to inter-layer dynamics and are better detected/localized via three-dimensional neighborhoods of sensory data. The classification accuracy performance improvement at $\geq$150 µm SED utilizing the within-layer and within-stripe data ($d = 5$) is likely due to relatively major process disturbances that are localized with a hatch stripe.

Quantitatively, when utilizing a three-dimensional neighborhood of inter-layer data ($d = 100$) the NN and PGNN variants perform better for porosities with an SED $\geq$50 µm (2). The improved classification accuracy when inter-layer data ($d = 100$) for porosities with a SED $\geq$50 µm is consistent with the results and conclusions of using within-layer data ($d = 5$). Using statistical moments on a large number of three-dimensional inter-layer data ($d = 100$) likely results in a loss of information on the porosities with a SED $\leq$50 µm. Perhaps applying a form of outlier detection and weighting mechanism before feature extraction on samples within a three-dimensional neighborhood of sensory data can improve the detection of the porosities with a SED $\leq$50 µm. The classification accuracy performance improvement between 50 and 250 µm SED utilizing the inter-layer data ($d = 100$) is hypothesized to be due to the porosities between 50 and 250 µm SED affecting the heat flow dynamics significantly enough to be captured via the photodiode sensory data in surrounding hatch stripes and preceding/succeeding layers.

The smallest porosity in the data set 35.284 µm SED ($0.023x10^{-}3$ mm^3) was randomly assigned to the training and validation portion of the data and was correctly detected and localized across all the model variants and their randomly initialed weights at $d = 5$. However, at $d = 100$, the 35.284 µm SED porosity was classified correctly as follows: NN $81.25 \pm 40.31\%$, PGNN $50.00 \pm 51.64\%$, PGNN_1 $43.75 \pm 51.23\%$, and PGNN_2 $43.75 \pm 51.23\%$. The classification accuracy degradation for the 35.284 µm SED porosity using inter-layer data ($d = 100$) further supports our hypothesis that porosities with an $\leq$ 50 µm SED occur due to hatch stripe disturbances. The classification accuracy improvement at $d = 5$ for porosities with a SED leq50 µm highlights the importance of models capable of within-layer data predictions in defect mitigation. Models that rely on three-dimensional neighborhoods of sensory data will only detect a porosity with a SED leq50 µm after at least one if not more succeeding layers have been sintered. Detecting and localizing a porosity after multiple sintered layers have succeeded may hinder the possibility and quality of mitigating said porosity.

Table 1 The classification accuracy average and standard deviation of the testing data across the porosity spherical equivalent diameter using within-layer data ($d = 5$). The average and standard deviation are across the 16 random model weight initialization.

Porosity	Number	$d = 5$			
SED (µm)	of Samples	NN	PGNN	$PGNN_1$	$PGNN_2$
0 (nominal)	2316	62.12 ± 0.85	62.31 ± 0.71	62.48 ± 2.95	62.58 ± 2.72
[38, 40]	5	80.00 ± 16.33	80.00 ± 16.33	73.75 ± 14.08	75.00 ± 13.67
[40, 42]	16	81.25 ± 3.95	80.08 ± 6.54	87.11 ± 2.77	87.50 ± 3.95
[42, 44]	35	80.71 ± 4.49	80.18 ± 4.36	80.36 ± 4.02	80.54 ± 4.81
[44, 46]	49	82.40 ± 4.14	83.16 ± 4.91	88.78 ± 3.94	89.16 ± 3.85
[46, 48]	101	74.88 ± 2.58	74.69 ± 2.82	78.16 ± 4.00	76.98 ± 4.27
[48, 50]	87	79.74 ± 2.84	80.53 ± 3.65	78.38 ± 3.23	78.88 ± 2.84
[50, 60]	454	74.12 ± 1.11	74.04 ± 1.61	76.64 ± 2.31	76.58 ± 2.27
[60, 70]	360	73.30 ± 1.74	73.19 ± 1.68	74.76 ± 2.36	75.03 ± 1.57
[70, 80]	297	73.91 ± 1.70	73.78 ± 2.04	77.25 ± 2.15	77.36 ± 2.56
[80, 90]	228	80.00 ± 1.91	80.02 ± 1.56	81.47 ± 2.19	81.72 ± 2.01
[90, 100]	220	73.15 ± 1.89	72.44 ± 1.93	74.15 ± 2.67	73.84 ± 1.76
[100, 150]	467	77.89 ± 1.06	77.81 ± 0.94	79.75 ± 1.60	79.20 ± 2.06
[150, 200]	89	81.74 ± 3.00	82.03 ± 2.05	84.69 ± 1.87	84.55 ± 1.81
[200, 250]	8	82.81 ± 7.74	83.59 ± 7.53	85.16 ± 5.04	85.16 ± 5.04
251.002	1	87.50 ± 34.16	87.50 ± 34.16	93.75 ± 25.00	93.75 ± 25.00

Table 2 The classification accuracy average and standard deviation of the testing data the varying porosity spherical equivalent diameter using inter-layer data ($d = 100$). The average and standard deviation are across the 16 random model weight initialization.

Porosity	Number	$d = 100$			
SED (µm)	of Samples	NN	PGNN	$PGNN_1$	$PGNN_2$
0 (nominal)	2316	91.71 ± 1.11	92.63 ± 0.39	92.76 ± 0.65	92.52 ± 0.45
[38, 40]	5	30.00 ± 10.33	26.25 ± 9.57	25.00 ± 8.94	26.25 ± 9.57
[40, 42]	16	53.13 ± 7.57	51.17 ± 7.30	50.39 ± 6.24	52.73 ± 6.02
[42, 44]	35	62.68 ± 6.96	62.86 ± 6.43	63.75 ± 5.39	64.64 ± 5.20
[44, 46]	49	67.35 ± 3.73	64.92 ± 2.81	64.93 ± 3.35	65.43 ± 3.91
[46, 48]	101	64.60 ± 5.02	63.49 ± 4.73	63.24 ± 3.09	64.23 ± 2.60
[48, 50]	87	79.31 ± 4.38	77.73 ± 4.32	77.51 ± 3.38	78.09 ± 3.04
[50, 60]	454	71.78 ± 3.00	70.80 ± 2.83	70.87 ± 2.37	71.20 ± 1.96
[60, 70]	360	80.54 ± 2.25	80.00 ± 1.89	79.77 ± 1.74	80.14 ± 1.30
[70, 80]	297	80.00 ± 1.90	79.50 ± 1.52	79.69 ± 1.24	79.61 ± 1.04
[80, 90]	228	78.87 ± 2.45	77.74 ± 2.03	77.44 ± 1.95	77.33 ± 1.62
[90, 100]	220	86.45 ± 1.67	85.82 ± 1.60	85.77 ± 1.31	86.05 ± 1.10
[100, 150]	467	84.46 ± 1.97	83.99 ± 1.61	83.87 ± 1.31	83.63 ± 0.97
[150, 200]	89	90.03 ± 1.92	89.75 ± 2.00	89.26 ± 2.17	88.62 ± 1.87
[200, 250]	8	85.16 ± 9.38	82.81 ± 7.74	81.25 ± 7.91	81.25 ± 6.46
251.002	1	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00

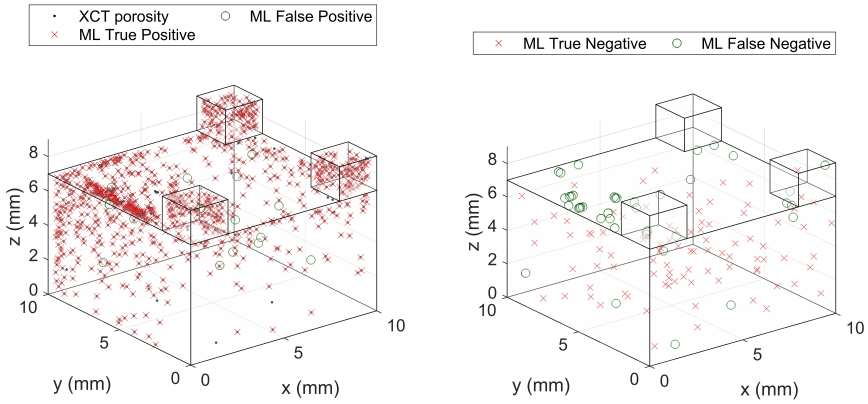


Fig. 15 A visualization of the XCT porosities and the vanilla NN predictions using inter-layer data ($d = 100$) for cube three: (a) true positive and false positive predictions and (b) true negative and false negative predictions.

At $d = 5$, using within-layer data, PGNN_1 and PGNN_2 outperform the NN and PGNN models (Table 1). The improved performance of PGNN_1 and PGNN_2 at $d = 5$ highlights the benefits of optimizing the NN number of hidden units and utilizing the prior data set in the model validation performance index. The improved performance of PGNN_1 and PGNN_2 at $d = 5$ also shows the benefits of utilizing the abundant nominal (porosity-free) data as a prior loss function. At $d = 100$, using inter-layer data, the NN and PGNN variants perform similarly, which shows that as the data information increases, the impact on the prior loss function reduces Atwya and Panoutsos (2022).

In Tables 1 and 2, relative to $d = 5$, the $d = 100$ models significantly improve the classification accuracy of the nominal samples. The $d = 100$ models' significant improvement of the classification accuracy of the nominal samples is due to the increase in the data knowledge relative to $d = 5$. The improvement at $d = 100$ also indicates that porosities may be mitigated through the preceding layers without altering the process parameters. The difference between $d = 5$ and $d = 100$ in correctly classifying the nominal samples highlights the importance of monitoring the preceding/succeeding layers' effects on the presence of porosity at a given coordinate.

Fig. 15 shows the training/validation and testing samples predictions for cube three using $d = 100$. Cube three process parameters were a border scan speed of 50.00 mm/s, a null hatch offset, and a border laser power of 500.00 W. Cube three laser energy density is too high due to the high laser power, low border scan speed, and null hatch offset. A high energy density is associated with keyhole porosities where droplets are ejected a high velocity out of the melt pool. The raid cooling and solidification of the molten hatch stripe often results in spherical and small porosities.

6 Conclusion

This work examines the use of photodiode sensory data in a machine learning framework for PBF in-situ porosity detection and localization. We propose a framework that can make in-situ porosity localization predictions via (1) to the author's knowledge, for the first time in literature, a local 1-dimensional and 2-dimensional neighborhood of within hatch stripe and within-layer sensory data and (2) a local 3-dimensional neighborhood of inter-layer sensory data (multiple z layers). To the author's knowledge, this is the first work in the literature capable of the localization of porosities smaller than 47 µm SED and as small as 38.118 µm SED with a classification accuracy of 80.00±16.33 using exclusively within hatch stripe and within layer sensory data Gobert et al (2018). The proposed method's localization of porosities as small as 38.118 µm SED is also a more than a five-fold improvement on the smallest SED porosity localization via photodiode sensory data in the literature Snow et al (2022).

Thus far, melt pool imaging has been considered one of the most critical monitoring methods in PBF for porosity detection Jafari-Marandi et al (2019); Tian et al (2021). This work shows that photodiode sensory data is sufficient for porosity detection and localization in metal PBF. Relying exclusively on photodiode sensory data for porosity detection and localization simplifies the required hardware, data acquisition, data processing, and ML models.

To the authors' knowledge, the proposed option (1) is the first time in the literature where in-situ porosity localization has been achieved exclusively within hatch stripe and within layer sensory data ($d = 5$). The benefit of utilizing within hatch stripe and within layer sensory data ($d = 5$) for in-situ porosity localization is quicker predictions without the need for surrounding hatch stripes or preceding/succeeding layer data. The capability to make porosity localization predictions exclusively within stripe and layer data facilitates quicker within-layer defect mitigation via altering the sintering process settings of the surrounding hatch stripes. Utilizing within hatch stripe and within layer sensory data ($d = 5$) improves the classification accuracy of porosities with an SED $\leq$50 µm, relative to inter-layer data ($d = 100$). The classification percentage improvements of using within layer data ($d = 5$) relative to inter-layer data ($d = 100$) for porosities with an SED $\leq$50 is NN 18.14%, PGNN 19.62%, $PGNN_1$ 19.66%, and $PGNN_2$ 18.24%. The limitations of utilizing within layer sensory data ($d = 5$) are a large false positive rate and a poor classification performance for porosities with an SED $\geq$50 µm, relative to using inter-layer data (2). The capability to make micro-porosity (SED $\leq$50 µm) localization predictions exclusively within stripe and layer data facilitates quicker within-layer defect mitigation via altering the sintering process settings of the surrounding hatch stripes. The capability of within-layer defect localization and mitigation can improve the quality of AM parts and improve AM compliance with the reliability and safety certification requirements of industries such as the aerospace industry Blakey-Milner et al (2021).

Utilizing local three-dimensional neighborhoods of inter-layer sensory data (multiple z layers, $d \geq 10$) for porosity detection and localization enables capturing the laser spot overlap and melt pool remelting zone effects. Relative to using fewer PBF samples ($d < 100$), the benefit of utilizing a three-dimensional neighborhood of sensory data ($d = 100$) for in-situ porosity localization are: (1) an improved classification accuracy of porosities with an SED $\geq$50 µm, (2) improved the classification accuracy of the nominal samples and the false positive rates, and (3) facilitates inter-layer porosity mitigation via altering the succeeding layers. The limitation of utilizing a three-dimensional neighborhood of inter-layer data ($d = 100$) include a poor classification performance for porosities with an SED $\leq$50 µm, relative to using within-layer data (2). Three-dimensional neighborhoods of data require collecting sensory data across several layers before making a porosity localization prediction which prohibits within-layer porosity mitigation. Detecting and localizing a porosity after multiple sintered layers have succeeded may hinder the possibility and quality of mitigating porosities.

When exclusively using within-layer data ($d = 5$), PGNN_1 and PGNN_2 reduce the model's number of hidden units and outperform the NN and PGNN models (Table 1). The improved performance of PGNN_1 and PGNN_2 at $d = 5$ highlights the benefits of (1) optimizing the number of hidden units and (2) utilizing the abundant nominal (porosity-free) data in the form of a prior loss function and the performance index model validation. As d increases, the training data become more informative and representative of the function, reducing the need for additional prior information. Hence, as As d increases, the PGNN models are less likely to improve the classification accuracy relative to the vanilla NN, in agreement with Atwya and Panoutsos (2022). However, the PGNN models consistently reduced the log loss error standard deviation across all the values of d, which increases the confidence in the model predictions.

A possible application of the proposed framework is to in-situ sequentially run models with varying granularity (d). Running a model with $d = 5$ can identify porosities more aggressively at the cost of a higher false positive count. However, sequentially as the available number of PBF samples (d) increases, the models with a larger d granularity can be utilized to update the porosity probability predictions and alter the process control to minimize porosity size/occurrence. Another application of the proposed framework is to run offline simulation and optimization algorithms to find the optimal spectral emission ranges for a given application-dependent criterion, such as a maximum number of porosities, maximum total porosity volume, or the maximum number of porosities for a given volume.

Further research might explore whether ML frameworks can be standardized across materials and sintering machines for porosity detection and localization. The question raised is where should the line be drawn between ML frameworks for generalizable applications of low-quantity/high-variety products and tailored high-quantity/low-variety products? This study shows that for a given sintered coordinate, the sintering processes parameters and sensor

measurements from up to 0.15 mm in three-dimension can have relevant information on the quality (porosity-presence) of said coordinate (Fig. 7). If the porosity status of a given coordinate is affected by at least three neighboring hatch stripes and the succeeding three layers, is online defect mitigation sufficiently possible? Or should porosity detection methods be primarily used for offline process optimization and post-processing quality documentation?

Running the classification models on each PBF sample in-situ might produce multiple false positive triggers in the coordinates nearest to the porosity coordinate. Future work includes running the models offline with varying d values on all the PBF samples of a sintered part to analyze the occurrence of false positives and propose a filtering method. A limitation of this work is that the model's predictions granularity is discrete, and there is a need for a model per d samples considered. A future outlook is to develop a dynamic model that can update the probability per coordinate on a rolling basis as new data is produced.

Supplementary information. Four supplementary appendices accompany this article.

Acknowledgments. The authors thank the Advanced Manufacturing Research Centre (AMRC) North West for assistance with data generation and acquisition.

Declarations

- **Funding:** This work was supported by the UK EPSRC Future Manufacturing Hub - Manufacture using Advanced Powder Processes (MAPP) through Grant EP/P006566/1 and the University of Sheffield's EPSRC Impact Acceleration Award, under project 'Machine Learning digital twin for defect-free additive manufacturing', 2022.
- **Conflict of interest/Competing interests:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- **Authors' contributions: Mohamed Atwya:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization.**George Panoutsos:** Conceptualization, Writing - review & editing, Super-vision, Funding acquisition.

Appendix A PBF data border labeling

This appendix details the border labeling procedure of the PBF sensory data. We demonstrate the sintering path, a 0.100 mm and null hatch offset in Fig. A1 and A2. Core samples can have a 0 gradient scan path, border samples can have a non-zero gradient, and due to the hatch offset, the last core and first border samples are not differentiable. As can be seen in Fig. A1, A2, and A3

finding the border samples is not as simple as searching for 0 gradient scan path.

To differentiate border samples from core samples, we split the procedure into four parts corresponding to the main cube and the three feature cubes. When searching for the border samples, we assess each layer per cube separately. The x and y coordinates per layer per cube are extracted via the z coordinate and cube number. In the scan path, the border is sintered after the core. The border samples are, therefore, chronologically at the end of the data set of each layer per cube. Therefore, the challenge is finding the first border sample per cube per layer since all the samples following are border samples.

We define the scan path absolute gradient per cube per layer in Eq. A1, A2, and A3. We define the three-sample moving average mean of the scan path absolute gradient in Eq. A4.

$$\begin{aligned} &\Delta\underline{\acute{x}}_i(j,k) = |\underline{\acute{x}}_{i+1}(j,k) - \underline{\acute{x}}_i(j,k)|, \\ &i \in \{1,\cdots,I-1\}, j \in \{1,\cdots,J\}, k \in \{1,\cdots,K\}, \end{aligned} \tag{A1}$$

where $\underline{\acute{x}}$ is the PBF scan path x coordinates, $J \in \mathbb{N}_1$ is the number of layers, $K \in \mathbb{N}_1$ is the number of cubes, $I \in \mathbb{N}_1$ is the number of samples in layer j cube k.

$$\begin{aligned} &\Delta\underline{\acute{y}}_i(j,k) = |\underline{\acute{y}}_{i+1}(j,k) - \underline{\acute{y}}_i(j,k)|, \\ &i \in \{1,\cdots,I-1\}, k \in \{1,\cdots,K\}, j \in \{1,\cdots,J\}, \end{aligned} \tag{A2}$$

where $\underline{\acute{y}}$ is the PBF scan path y coordinates.

$$\underline{f}'_i = \begin{cases} 0 \text{ for } \Delta\underline{\acute{x}}_i = 0 \vee \Delta\underline{\acute{y}}_i = 0 \\ \frac{\Delta\underline{\acute{y}}_i}{\Delta\underline{\acute{x}}_i} \text{ for } \Delta\underline{\acute{x}}_i \neq 0 \wedge \Delta\underline{\acute{y}}_i \neq 0, \end{cases} \tag{A3}$$

where $\underline{f}'_i$ is the scan path gradient.

$$\underline{g}_i = \begin{cases} \frac{1}{3}\sum_{j=i}^{i+2} \underline{f}'_j \text{ for } i = 1 \\ \frac{1}{3}\sum_{j=i-1}^{i+1} \underline{f}'_j \text{ for } i \in \{2,\cdots,I-2\} \\ \frac{1}{3}\sum_{j=i-2}^{i} \underline{f}'_j \text{ for } i = I-1 \end{cases} \tag{A4}$$

where $\underline{g}_i$ is the moving average mean of the scan path absolute gradient.

Let $\underline{u} \in \{0,1\}^c$ be a logical label for each PBF sample where 1 represents a border sample, 0 represents a core sample, and $c \in \mathbb{N}_1$ is the number of PBF samples. The strategy proposed to find the first border sample per cube per layer is:

1. Find u^*_m, where $m \in \mathbb{N}_1$ is the index of a core sample sufficiently close to the border samples.
2. Find u^*_n, where $n \in \mathbb{N}_1$ is the index of a border sample (not necessarily the first border sample).
3. Search between u^*_m and u^*_n for u^*_p, where $p \in \mathbb{N}_1$ is the index of the sample most likely to be the first border sample.

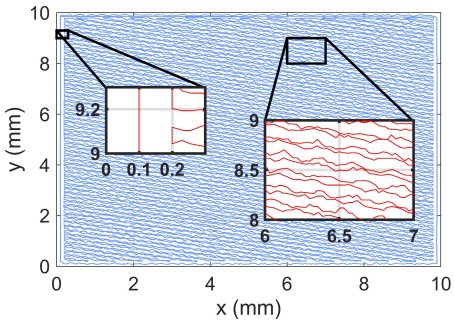


Fig. A1 Cross-sectional view of cube 2 at layer 140 showing the 0.1 mm hatch offset and the scan path.

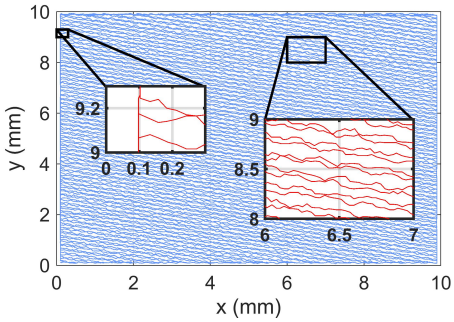


Fig. A2 Cross-sectional view of cube 3 at layer 140 showing the null hatch offset and the scan path

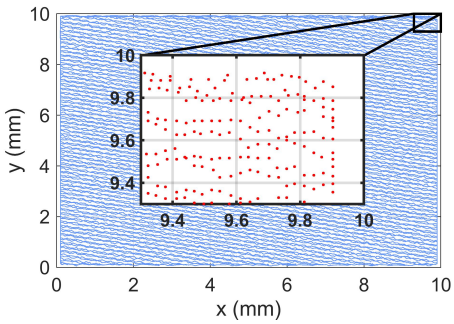


Fig. A3 Cross-sectional view of cube 41 at layer 140 showing the null hatch offset and the overlap between core and border samples.

On average, cubes with the same scan speed have the same number of border samples. The faster the scan speed, the fewer border samples, and vice versa. Through empirical testing, cubes with 300 mm/s scan speed have at least 98% core samples per layer per cube. Cubes with 175 mm/s scan speed have at least 97% core samples per layer per cube. Cubes with 50 mm/s

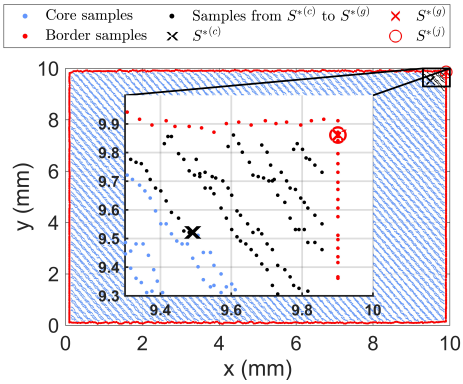


Fig. A4 Cross-sectional view of cube 5 at layer 1 showing the border labeling steps and the final border labels in red.

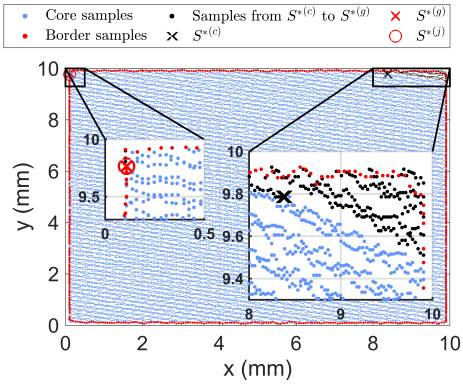


Fig. A5 Cross-sectional view of cube 41 at layer 140 showing the border labeling steps and the final border labels in red.

scan speed have at least 92% core samples per layer per cube. We define the core sample as sufficiently close to the border samples according to these core sample percentages. For example, for cubes with a 300 mm/s scan, we consider the first sample after 98% of the samples to be the core sample sufficiently close to the border samples u_m^*. The proposed strategy eliminates the majority of the core samples and the possibility of incorrectly selecting a core sample with a 0 gradient scan path as the first border sample. For the feature cubes, u_m^* is chronologically after 90%, 85%, and 60% of the samples per cube per layer for the scan speeds 300 mm/s, 175 mm/s, and 50 mm/s, respectively.

The first scan path moving-average-mean absolute gradient $\underline{u}_i = 0.000$ is considered a border sample u_n^*, but not necessarily the first border sample. We find u_p^* the sample most likely to be the first border sample in Eq. A5. If $\underline{u}_j < \left(\mu_{\underline{u}_j} - 3\sigma_{\underline{u}_j}\right)$ does not exist then, $u_p^* = u_n^*$.

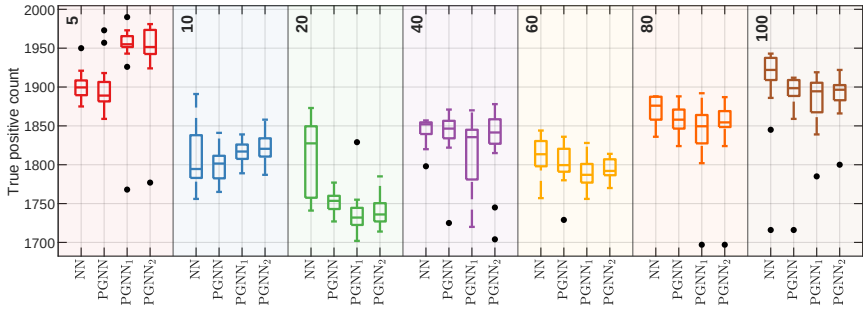


Fig. B6 The true positive count box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

$$\begin{aligned} u_p^* = & \underset{j \in h}{\arg\min} && j \mid \underline{u}_j < \left(\mu_{\underline{u}_j} - 3\sigma_{\underline{u}_j}\right) \\ & \text{subject to} && h \in \{m, m+1, \cdots, n\}. \end{aligned} \tag{A5}$$

For each sample per cube per layer in $\underline{u}(j,k)$ we use the identified index p from Eq. A5 to define whether a sample is a core or a border sample. Indices greater than or equal to p are considered the border. The same process is repeated per cube per layer for the three small feature cubes. An example of the border labeling process for cube 5 layer 1 (hatch offset 0.050 mm) is shown in Fig. A4. Fig. A5 shows the border labeling process for cube 41 layer 140 (null hatch offset), which has a different scan path relative to cube 5.

Appendix B Extended classification results

Fig. B6 and B7 show the true positive and false positive counts, where a positive refers to a porosity. There is no qualitatively clear trend in the true positive count versus the number of compressed PBF samples d. However, at the smallest value $d = 5$ the NN and PGNN variants have a relatively high true positive count (Fig. B6). Similarly at $d = 5$, the NN and PGNN variants have the highest false positive count (Fig. B7). Fig. B8 and B9 show the true negative and false negative counts. The true/false negative counts mirror the true/false positive count. The true and false positive/negative trends in Fig. B6, B7, B8, and B9 are in line with the analysis of the precision and recall trends (Subsection 5).

References

ASTM, ISO, et al (2015) Astm iso/astm52900-15 standard terminology for additive manufacturing–general principles–terminology. https://doi.org/https://doi.org/10.1520/ISOASTM52900-15

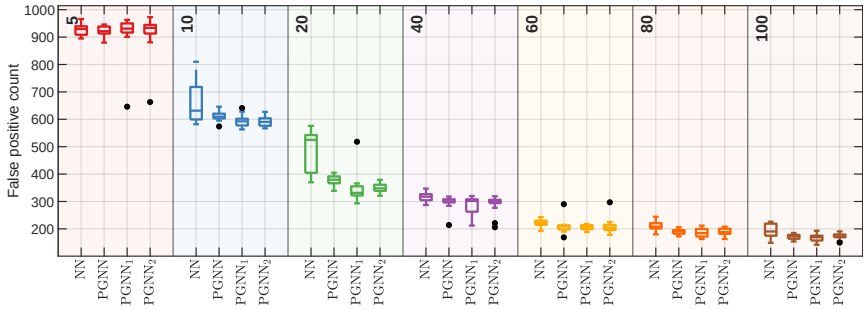


Fig. B7 The false positive count box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

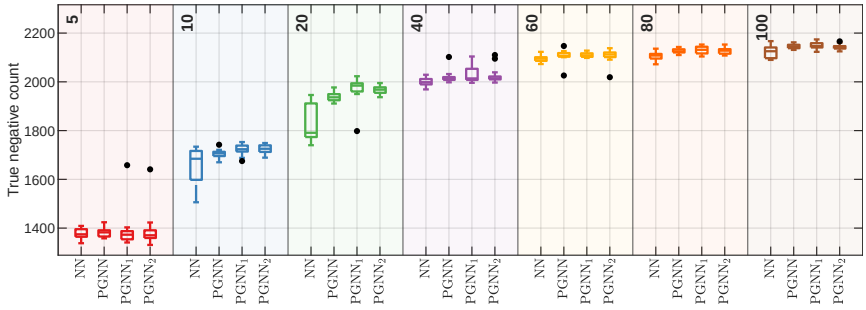


Fig. B8 The true negative count box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

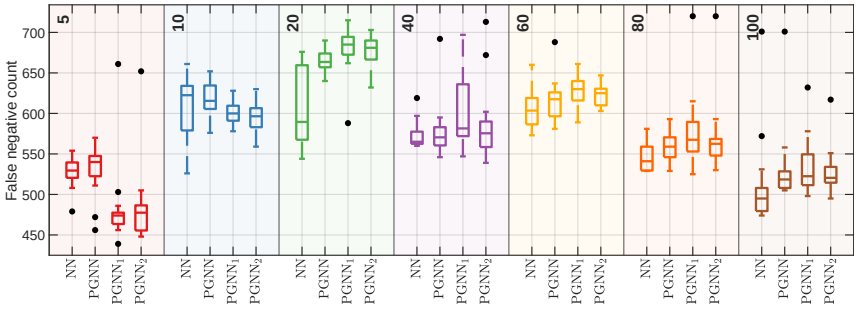


Fig. B9 The false negative count box plot of each model variant across the 16 random model weight initialization. The results are color coded and labeled according to the number of compressed PBF samples d per ML sample.

Atwya M, Panoutsos G (2019) Transient thermography for flaw detection in friction stir welding: a machine learning approach. IEEE Transactions on Industrial Informatics 16(7):4423–4435

Atwya M, Panoutsos G (2022) Structure optimization of prior-knowledge-guided neural networks. Neurocomputing 491:464–488

Bellgran M, Säfsten EK (2009) Production development: design and operation of production systems. Springer Science & Business Media, https://doi.org/10.1007/978-1-84882-495-9, URL https://www.springer.com/gb/book/9781848824942

Blakey-Milner B, Gradl P, Snedden G, et al (2021) Metal additive manufacturing in aerospace: A review. Materials & Design 209:110,008

Chen H, Zhao YF (2015) Learning algorithm based modeling and process parameters recommendation system for binder jetting additive manufacturing process. In: ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection

Childs CM, Washburn NR (2019) Embedding domain knowledge for machine learning of complex material systems. MRS Communications 9(3):806–820

Choo H, Sham KL, Bohling J, et al (2019) Effect of laser power on defect, texture, and microstructure of a laser powder bed fusion processed 316l stainless steel. Materials & Design 164:107,534

Chryssolouris G (2013) Manufacturing systems: theory and practice. Springer Science & Business Media, URL https://www.springer.com/gb/book/9780387256832

Clymer DR, Cagan J, Beuth J (2017) Power–velocity process design charts for powder bed additive manufacturing. Journal of Mechanical Design 139(10):100,907

Conner BP, Manogharan GP, Martof AN, et al (2014) Making sense of 3-d printing: Creating a map of additive manufacturing products and services. Additive Manufacturing 1:64–76

Dass A, Moridi A (2019) State of the art in directed energy deposition: From additive manufacturing to materials design. Coatings 9(7):418

DebRoy T, Wei H, Zuback J, et al (2018) Additive manufacturing of metallic components–process, structure and properties. Progress in Materials Science 92:112–224

Fraser K, Kiss L, St-Georges L, et al (2018) Optimization of friction stir weld joint quality using a meshfree fully-coupled thermo-mechanics approach. Metals 8(2):101

Fu Y, Downey AR, Yuan L, et al (2022) Machine learning algorithms for defect detection in metal laser-based additive manufacturing: a review. Journal of Manufacturing Processes 75:693–710

Gaja H, Liou F (2017) Defects monitoring of laser metal deposition using acoustic emission sensor. The International Journal of Advanced Manufacturing Technology 90(1):561–574

Gaja H, Liou F (2018) Defect classification of laser metal deposition using logistic regression and artificial neural networks for pattern recognition. The International Journal of Advanced Manufacturing Technology 94(1):315–326

Galarraga H, Lados DA, Dehoff RR, et al (2016) Effects of the microstructure and porosity on properties of ti-6al-4v eli alloy fabricated by electron beam melting (ebm). Additive Manufacturing 10:47–57

Gao W, Zhang Y, Ramanujan D, et al (2015) The status, challenges, and future of additive manufacturing in engineering. Computer-Aided Design 69:65 – 89. https://doi.org/10.1016/j.cad.2015.04.001

Garg A, Tai K, Savalani M (2014) State-of-the-art in empirical modelling of rapid prototyping processes. Rapid Prototyping Journal 20(2):164–178

Ge P, Zhang Z, Tan Z, et al (2019) An integrated modeling of process-structure-property relationship in laser additive manufacturing of duplex titanium alloy. International Journal of Thermal Sciences 140:329–343

Girelli L, Giovagnoli M, Tocci M, et al (2019) Evaluation of the impact behaviour of alsi10mg alloy produced using laser additive manufacturing. Materials Science and Engineering: A 748:38–51

Gobert C, Reutzel EW, Petrich J, et al (2018) Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. Additive Manufacturing 21:517–528

Guo S, Agarwal M, Cooper C, et al (2022) Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm. Journal of Manufacturing Systems 62:145–163

He K, Zhang X, Ren S, et al (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034

Hendricks KB, Singhal VR (2008) The effect of product introduction delays on operating performance. Management Science 54(5):878–892. https://doi.org/10.1287/mnsc.1070.0805

Ishikura H (2001) New product development and planning. International Journal of Manufacturing Technology and Management 3(3):238–250

Jafari-Marandi R, Khanzadeh M, Tian W, et al (2019) From in-situ monitoring toward high-throughput process control: cost-driven decision-making framework for laser-based additive manufacturing. Journal of Manufacturing Systems 51:29–41

Jagtap AD, Kawaguchi K, Karniadakis GE (2020) Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics 404:109,136

Jayasinghe S, Paoletti P, Sutcliffe C, et al (2022) Automatic quality assessments of laser powder bed fusion builds from photodiode sensor measurements. Progress in Additive Manufacturing 7(2):143–160

Jha R, Chakraborti N, Diercks DR, et al (2018) Combined machine learning and calphad approach for discovering processing-structure relationships in soft magnetic alloys. Computational Materials Science 150:202–211

Kah P, Rajan R, Martikainen J, et al (2015) Investigation of weld defects in friction-stir welding and fusion welding of aluminium alloys. International Journal of Mechanical and Materials Engineering 10(1):26. https://doi.org/10.1186/s40712-015-0053-8, URL https://link.springer.com/article/10.1186/s40712-015-0053-8

Kan WH, Chiu LNS, Lim CVS, et al (2022) A critical review on the effects of process-induced porosity on the mechanical properties of alloys fabricated by laser powder bed fusion. Journal of Materials Science 57(21):9818–9865

King WE, Barth HD, Castillo VM, et al (2014) Observation of keyhole-mode laser melting in laser powder-bed fusion additive manufacturing. Journal of Materials Processing Technology 214(12):2915–2925

Lin X, Zhu K, Fuh JYH, et al (2022) Metal-based additive manufacturing condition monitoring methods: from measurement to control. ISA transactions 120:147–166

Liu D, Wang Y (2019) Multi-fidelity physics-constrained neural network and its application in materials modeling. Journal of Mechanical Design 141(12)

Liu R, Liu S, Zhang X (2021) A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing. The International Journal of Advanced Manufacturing Technology 113(7):1943–1958

Meng L, McWilliams B, Jarosinski W, et al (2020) Machine learning in additive manufacturing: a review. Jom 72(6):2363–2377

Muralidhar N, Islam MR, Marwah M, et al (2018) Incorporating prior domain knowledge into deep neural networks. In: 2018 IEEE International Conference on Big Data (Big Data), IEEE, pp 36–45

Norazman F, Hopkinson N (2014) Effect of sintering parameters and flow agent on the mechanical properties of high speed sintered elastomer. Journal of Manufacturing Science and Engineering 136(6). https://doi.org/10.1115/1.4028482

Okaro IA, Jayasinghe S, Sutcliffe C, et al (2019) Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. Additive Manufacturing 27:42–53

Pei W, Zhengying W, Zhen C, et al (2017) Numerical simulation and parametric analysis of selective laser melting process of alsi10mg powder. Applied Physics A 123(8):1–15

Popova E, Rodgers TM, Gong X, et al (2017) Process-structure linkages using a data science approach: application to simulated additive manufacturing data. Integrating Materials and Manufacturing Innovation 6(1):54–68

Prashanth KG, Scudino S, Maity T, et al (2017) Is the energy density a reliable parameter for materials synthesis by selective laser melting? Materials Research Letters 5(6):386–390

Qi X, Chen G, Li Y, et al (2019) Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives. Engineering

Sattari S, Bisadi H, Sajed M (2012) Mechanical properties and temperature distributions of thin friction stir welded sheets of AA5083. International Journal of Mechanics and Applications 2(1):1–6. https://doi.org/10.5923/j.mechanics.20120201.01, URL http://article.sapub.org/10.5923.j.mechanics.20120201.01.html

Shen X, Yao J, Wang Y, et al (2004) Density prediction of selective laser sintering parts based on artificial neural network. In: International Symposium on Neural Networks, Springer, pp 832–840

Snow Z, Reutzel EW, Petrich J (2022) Correlating in-situ sensor data to defect locations and part quality for additively manufactured parts using machine learning. Journal of Materials Processing Technology 302:117,476

Song B, Dong S, Liao H, et al (2012) Process parameter selection for selective laser melting of ti6al4v based on temperature distribution simulation and experimental sintering. The international journal of advanced manufacturing technology 61(9):967–974

Sood AK, Ohdar RK, Mahapatra SS (2010) Parametric appraisal of fused deposition modelling process using the grey taguchi method. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 224(1):135–145. https://doi.org/10.1243/09544054JEM1565, URL https://doi.org/10.1243/09544054JEM1565, https://arxiv.org/abs/https://doi.org/10.1243/09544054JEM1565

Taherkhani K, Eischer C, Toyserkani E (2022) An unsupervised machine learning algorithm for in-situ defect-detection in laser powder-bed fusion. Journal of Manufacturing Processes 81:476–489

Tian Q, Guo S, Melder E, et al (2021) Deep learning-based data fusion method for in situ porosity detection in laser-based additive manufacturing. Journal of Manufacturing Science and Engineering 143(4)

Tofail SA, Koumoulos EP, Bandyopadhyay A, et al (2017) Additive manufacturing: scientific and technological challenges, market uptake and opportunities. Materials Today https://doi.org/10.1016/j.mattod.2017.07.001

Wang W, Ning J, Liang SY (2021) Prediction of lack-of-fusion porosity in laser powder-bed fusion considering boundary conditions and sensitivity to laser power absorption. The International Journal of Advanced Manufacturing Technology 112(1):61–70

Xia M, Gu D, Yu G, et al (2016) Influence of hatch spacing on heat and mass transfer, thermodynamics and laser processability during additive manufacturing of inconel 718 alloy. International Journal of Machine Tools and Manufacture 109:147–157

Chapter 5

Conclusion

In the literature, expertly tailored prior-guided NNs methods have successfully outperformed traditional NNs in terms of empirical performance and prior-consistency. However, the need to tailor PGNNs on per-application basis via expert knowledge and empirical experimentation limits the applications and objectivity of PGNNs [16, 19]. This work aimed to develop and statistically test a systematic framework to optimize the structure and hyperparameters of PGNNs while ensuring empirical generalizability and prior-consistency. We proposed the following contributions to PGNNs:

1. A statistical analysis on the empirical and prior-consistency error performance of PGNNs under varying settings, including the effects of random model weight initialization, data set size, hyperparameters, and the correlation between the empirical prior-consistency errors.

2. An analysis of the effects of the prior loss weights on the PGNN performance in the case of single versus multi-prior modeling problems.

3. A framework to optimize the number of hidden units via a line search and cross-validation using the empirical error to eliminate data-set/model-structure application dependency (PGNN_1).

4. We propose utilizing the prior errors as part of the cross-validation performance index to reinforce prior-consistency (PGNN_2)

5. A multi-prior correlation-based adaptive weighting metric for choosing the prior-error weights during the learning step (CoReg-PGNN_2 and CoReg-PGNN_3).

6. A multi-prior correlation-based weighting metric for choosing the prior-error weights during the cross-validation step (CoReg-PGNN_1).

The key finding of the methodologies work towards prior-guided neural networks are summarized below.

1. We found a statistically significant ($P < 0.05$) positive correlation between the empirical and the prior-consistency RMSEs for the three prior scenarios in two benchmark case studies. The positive correlation between the empirical and the prior-consistency RMSEs demonstrates the importance of developing prior-consistent NNs.

2. Optimizing the number of hidden units via a line search and repeated cross-validation using the empirical error ($PGNN_1$) eliminates data-set/model-structure application dependency and statistically significantly improves the empirical and prior-consistency RMSE means relative to the reference models.

    - A limitation of optimizing the number of hidden units via a line search $PGNN_1$ is the possibility of a larger hidden unit quantile range across random model weight initialization. The $PGNN_1$ limitation is hypothesized to be due to the data set size versus the function's complexity and the un-tuned loss weight hyperparameters.
    - A complication with constrained optimization is that it is a reactive technique. Traditional NN regularization (complexity control) methods such as L_1 and L_2 are agnostic to the prior-consistency of the learned model and thus will often prioritize reducing the prediction error at the cost of a physically inconsistent solution. Therefore, prior-based constraints reactively attempt to compete with the complexity control loss to minimize the prior-inconsistency. Accordingly, more work is needed to set the foundations of constrained optimization.

3. Using the prior-consistency error with the empirical error in the cross-validation performance index ($PGNN_2$) statistically significantly improves the empirical and prior-consistency RMSE mean and variance relative to the reference models.

    - By incorporating the prior-consistency error in the validation performance index, $PGNN_2$ re-enforces prior-consistency outperforms (empirical/prior error mean and variance) $PGNN_1$ relative to the reference models by selecting a suitable model candidate with a number of hidden units that avoids under-fitting or

over-fitting depending on the training/prior data landscape versus the data set size.

- The positive correlation between the empirical and the prior-consistency RMSEs discovered in this work further supports our proposal of using the prior-consistency error in the validation performance index of PGNNs (PGNN_2).

4. The data set size investigation for PGNN_1 and PGNN_2 has shown that PGNNs result in a 1-dimensional or 2-dimensional statistically significant difference ($P < 0.05$) in the multivariate empirical and prior-consistency means for convex and non-convex modeling problems up to a 2000 sample data set size.

    - The univariate analysis has shown a statistically significant difference ($P < 0.05$) in the empirical and prior-consistency errors for up to sample data set size 4000 and 2000, respectively.
    - The functions' complexity, the prior value, and the correlation between the empirical and the prior-consistency errors dictate the magnitude of the multivariate difference between data set sizes.

5. In the hyperparameter sensitivity investigation, we show that the proposed methods in PGNN_1 and PGNN_2 are not affected by the loss weight hyperparameters and consistently improve the mean empirical and prior-consistency RMSEs relative to the reference models.

6. For modeling problems with multiple priors, the empirical/prior-consistency performance is affected more significantly by the prior loss weights than the prior error weights in the validation PI. We hypothesize that the prior loss weights significantly impact the model performance relative to the validation error weights because the loss search space is larger than the validation PI search space.

    - For modeling problems with multiple priors, the proposed correlation-based adaptive prior loss weighting metric, along with the equally weighted sum of the empirical and prior errors as the cross-validation performance index (CoReg-PGNN_3), achieve the best empirical and prior-consistency RMSE improvements.
    - For modeling problems with a single prior and a lack of knowledge of the value of said prior, the proposed correlation-based prior loss adaptive weighting and weighted correlation-based model validation

(CoReg-PGNN$_4$), on average, achieve the best empirical and prior-consistency RMSE improvements.

7. When the empirical data is not fully-representative, a prior that is statistically correlated to empirical data can add value in terms of the empirical and prior-consistency performance of a data-driven framework such as PGNNs. However, as evident in this work, the methods used to incorporate the priors into a NN training/validation framework might be detrimental to the PGNN empirical performance. Given the proposed methods and results, we conclude that for modeling problems with single or multiple priors, if the relationship between any of the prior errors and the empirical error is nonlinear or unknown, all the priors should be equally weighted in the validation performance index to select the most generalizable model.

Regarding Chapter 4, in-situ monitoring and controlling the defect formations in AM is necessary as it dictates the mechanical properties of the sintered parts. Existing work in AM data-driven defect detection and localization has demonstrated empirical success in utilizing sensory and physics-driven data to detect porosities in three-dimensional data neighborhoods of a sintered part. However, the current work in the literature is limited by the need for inter-layer three-dimensional neighborhoods of sensory data, the need for several sensors, and poor experimental and model validation. This work aimed to develop a data-driven prior-guided neural network framework capable of in-situ porosity localization predictions for selective laser melting. We proposed the following contributions in our application of PGNNs to porosity localization:

1. In-situ detection and localization of porosities using standardization-friendly vanilla NN models, photodiode data, and machine-independent physics-driven features.

2. A framework that, given a three-dimensional sintering coordinate, can make in-situ porosity localization predictions via (1) to the author's knowledge, for the first time in literature, a local 1-dimensional and 2-dimensional neighborhoods of within hatch stripe and within-layer sensory data and (2) a local 3-dimensional neighborhood of inter-layer sensory data (multiple z layers).

3. A framework based on prior-guided neural networks to utilize the often abundant PBF nominal data in the form of a prior training loss and a prior validation error.

4. To the author's knowledge, this is the first work in the literature capable of the localization of porosities as small as 38.118 µm SED ($0.029x10^{-}3$ mm^3), smaller than 47 µm SED in [1]. To the author's knowledge, the proposed methods localization of porosities as small as 38.118 µm SED is also a more than a five-fold improvement on the smallest SED porosity localization via photodiode sensory data [38].

The key findings of the application work are summarized below.

1. Thus far, melt pool imaging has been considered one of the most critical monitoring methods in PBF for porosity detection [43, 44]. This work shows that photodiode sensory data is sufficient for porosity detection and localization in metal PBF. Relying exclusively on photodiode sensory data for porosity detection and localization simplifies the required hardware, data acquisition, data processing, and ML models.

2. An increase in the number of compressed PBF samples d per ML sample improves the porosity classification accuracy due to the:

   (a) An increase in the information captured per ML sample via feature extraction.

   (b) The models are less affected by the geometric and dimensional errors when mapping the XCT scans on the PBF data.

   (c) The models capture the effects of the surrounding hatch paths and the preceding and succeeding layers (i.e., the laser spot overlapping and the melt pool remelting zone).

3. Utilizing the abundant information-rich nominal PBF data in the form of a prior loss and validation error reduces the log loss score standard deviation and provides higher confidence in the model's predictions. Optimizing and reducing the number of hidden units and incorporating the prior-consistency error in the validation performance index ($PGNN_1$ and $PGNN_2$) results in (1) avoiding over-fitting, (2) reducing the classification log loss error standard deviation for all d values, and (3) improving the classification accuracy when less sensory data is utilized.

4. When exclusively utilizing within hatch stripe data ($d = 5$), the model will detect more true porosities but will often miss-classify porosity-free coordinates as porosity. The poor precision at $d = 5$ is hypothesized to be the model's potentially picking up on what might have been porosity but was corrected via the effects of the surrounding hatch paths and the preceding and succeeding layers.

5. Quantitatively, when exclusively utilizing within-layer data ($d = 5$), the NN and PGNN variants perform better for porosities with an SED smaller than 50 µm and larger than 150 µm. We hypothesize that porosities with a SED $\leq$50 µm occur primarily due to within hatch stripe disturbances and are less likely to form due to surrounding hatch stripes and the preceding/succeeding layers. Hence, utilizing the within-layer and within-stripe data ($d = 5$) is hypothesized to perform better on porosities with a SED $\leq$50 µm. The classification accuracy performance drop between 50 and 150 µm SED utilizing the within-layer and within-stripe data ($d = 5$) is hypothesized to be due to the lack of sensory information on the melt pool re-melting zone and the laser spot overlap. Therefore, porosities in the $[50, 150]$ µm SED range likely occur due to inter-layer dynamics and are better detected/localized via three-dimensional neighborhoods of sensory data. The classification accuracy performance improvement at $\geq$150 µm SED utilizing the within-layer and within-stripe data ($d = 5$) is likely due to relatively significant process disturbances that are localized with a hatch stripe.

6. Quantitatively, when utilizing a three-dimensional neighborhood of inter-layer data ($d = 100$), the NN and PGNN variants perform better for porosities with an SED $\geq$50 µm. The improved classification accuracy when inter-layer data ($d = 100$) for porosities with a SED $\geq$50 µm is consistent with the results and conclusions of using within-layer data ($d = 5$). Using statistical moments on a large number of three-dimensional inter-layer data ($d = 100$) likely results in a loss of information on the porosities with a SED $\leq$50 µ. Perhaps applying a form of outlier detection and weighting mechanism before feature extraction on samples within a three-dimensional neighborhood of sensory data can improve the detection of the porosities with a SED $\leq$50 µ. The classification accuracy performance improvement between 50 and 250 µm SED utilizing the inter-layer data ($d = 100$) is hypothesized to be due to the porosities

between 50 and 250 µm SED affecting the heat flow dynamics significantly enough to be captured via the photodiode sensory data in surrounding hatch stripes and preceding/succeeding layers.

5.1 Future work

To further investigate our work on structure optimization of PGNNs (Chapter 2), an empirical investigation should compare the proposed PGNNs against a multi-layer perceptron with a fixed number of hidden units, L1 regularization, and L2 regularization (elastic net) where the regularization weights are optimized through methods such as a grid search or Bayesian Optimization. The weights and the number of hidden units of the structurally optimized PGNN variants should be examined against the weights of the elastic net to compare the feature selection effect of elastic nets against the proposed PGNN model capacity effects of the structure optimization and prior-guided cross-validation.

Our prior loss adaptive weighting methods (Chapter 3) does not consider the non-convexity of the empirical and prior loss landscapes. Where a prior is informative in the inference of output given an input, there is a statistical association between the empirical loss landscape and the prior loss landscape. If there is no statistical association between the empirical loss landscape and the prior loss landscape, then minimizing the prior loss will not affect the empirical loss. Priors that provide additional training samples with input features and target values (e.g., initial loss and boundary priors) are likely to have a loss landscape that is statically associated with the empirical loss landscape and is expected to result in empirical loss and testing improvements. Priors that are not statistically related to empirical loss result in a multi-objective optimization problem with uncorrelated objectives; the prior loss objective and the empirical loss objective are uncorrelated. For future work, updating the loss weighting metric such that the correlation is not only measured across the previous epochs but instead uses the loss landscape neighborhood to estimate (1) how many previous epochs to use and (2) take into account the prior loss value and gradient across the following possible epochs. Prior informativeness via entropy-based measures and prior informativeness-based weighting metrics could be usefully explored in further research.

A natural progression of the proposed prior loss adaptive weighting metric is to take into account the p-value and the number of samples (epochs) required for inference based on a Pearson correlation coefficient. Particularly, emphasis should be placed on investigating the impact of the prior loss weights during

the early training stage on the convergence stability and minima sharpness of the scaled conjugate gradient method.

Our proposed adaptive weighting methods which utilize Pearson's correlation coefficient would not be a suitable choice for applications where the variables are already known to have a nonlinear relation. Possible solutions include transforming the variables to have a normal distribution and linearizing the relationship between the variables. Another approach is to utilize other correlation metrics, such as Spearman's rank correlation or distance correlation. For example, Spearman's rank correlation can examine nonlinear monotonic relationships and does not require normally distributed variables.

Regarding our laser powder bed fusion porosity localization work in Chapter 4, additional features such as the porosity density per layer, the number of porosities on previous layers, and a geometry-independent distance-from-surface metric should be investigated in future work. Features that rely on the previous layers' porosity predictions should consider the resulting compound error. Distance-from-surface metrics should consider the neighborhood part volume and resulting energy density (i.e. thin protruding features versus relatively bulky features). The distance-from-surface metrics should also accommodate irregular features and curves to generalize on varying part geometries.

One research question yet to be addressed in the literature is what is the correct shape of the three-dimensional neighborhood of sensory data that is informative for a given porosity coordinate? Typically, in the literature, the porosity status for a given coordinate within a layer is predicted via all the data from the layer in question and a given number of preceding and succeeding layers. In our work, we proposed utilizing the closest d sensory samples to a given coordinate in terms of the three-dimensional euclidean distance, where d is a hyperparameter ranging from 5 to 100 samples. However, in the literature and our proposed work, a given sample's importance must be quantified within a three-dimensional porosity neighborhood. A greater focus on data fidelity could significantly improve the empirical accuracy of porosity localization and shed light on the porosity formation mechanism. A better understanding of data fidelity could also enable the detection of porosities before they occur and contribute towards Right First Time (RFT) manufacturing.

A general shortcoming of NNs and PGNNs is interpretability. With AM porosity detection and quantification as a motivator, further research might explore developing new prior incorporation frameworks to combine physics-based computational fluid dynamics models and PGNNs towards more interpretable model ling solutions.

Further research might explore whether ML frameworks could be standardized across materials, varying part geometry, and sintering machines for porosity detection and localization. The question raised is where should the line be drawn between ML frameworks for generalizable applications of low-quantity/high-variety products and tailored high-quantity/low-variety products? This study shows that for a given sintered coordinate, the sintering processes parameters and sensor measurements from up to 0.15 mm in three-dimension can have relevant information on the quality (porosity-presence) of said coordinate. If the porosity status of a given coordinate is affected by at least three neighboring hatch stripes and the succeeding three layers, is online defect mitigation sufficiently possible? Or should porosity detection methods be primarily used for offline process optimization and post-processing quality documentation?

Running the classification models on each PBF sample in-situ might produce multiple false positive triggers in the coordinates nearest to the porosity coordinate. Future work includes running the models offline with varying d values on all the PBF samples of a sintered part to analyze the occurrence of false positives and propose a filtering method. A limitation of this work is that the model's predictions granularity is discrete, and there is a need for a model per d samples considered. A future outlook is to develop a dynamic model that can update the probability per coordinate on a rolling basis as new data is produced.

A limitation of the porosity localization work in Chapter 4 is that it only predicts the probability of a porosity occurring in the range $[0, 1]$ but does not quantify the volume of the porosity. Further research should examine predicting the porosity volume to aid defect mitigation decision-making and part certification.

An over-arching major limitation in additive manufacturing research and application is the variations in the sensory systems across different sintering machines and the limited open-source data available. An open-source multi-material platform with open-source data collation and an online repository is necessary and would significantly speed up progress in the context of industrial applications, standardization, and research.

References

1. Gobert, C., Reutzel, E. W., Petrich, J., Nassar, A. R. & Phoha, S. Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Additive Manufacturing* **21,** 517–528 (2018).

2. Ghungrad, S., Gould, B., Wolff, S. & Haghighi, A. *Physics-Informed Artificial Intelligence for Temperature Prediction in Metal Additive Manufacturing: A Comparative Study* in *International Manufacturing Science and Engineering Conference* **85802** (2022), V001T01A008.

3. Atwya, M. & Panoutsos, G. Transient thermography for flaw detection in friction stir welding: a machine learning approach. *IEEE Transactions on Industrial Informatics* **16,** 4423–4435 (2019).

4. Karpatne, A. *et al.* Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* **29,** 2318–2331 (2017).

5. Hautier, G., Fischer, C. C., Jain, A., Mueller, T. & Ceder, G. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chemistry of Materials* **22,** 3762–3767 (2010).

6. Schmidt, V. *et al.* Visualizing the consequences of climate change using cycle-consistent adversarial networks. *arXiv preprint arXiv:1905.03709* (2019).

7. Zobeiry, N. & Humfeld, K. D. An Iterative Scientific Machine Learning Approach for Discovery of Theories Underlying Physical Phenomena. *arXiv preprint arXiv:1909.13718* (2019).

8. Schrodt, F. *et al.* BHPMF–a hierarchical Bayesian approach to gap-filling and trait prediction for macroecology and functional biogeography. *Global Ecology and Biogeography* **24,** 1510–1521 (2015).

9. Jia, X. *et al. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles* in *Proceedings of the 2019 SIAM International Conference on Data Mining* (2019), 558–566.

10. Keshavan, R. H., Montanari, A. & Oh, S. Matrix completion from a few entries. *IEEE transactions on information theory* **56,** 2980–2998 (2010).

11. Erhan, D., Courville, A., Bengio, Y. & Vincent, P. *Why does unsupervised pre-training help deep learning?* in *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), 201–208.

12. Chatterjee, S., Steinhaeuser, K., Banerjee, A., Chatterjee, S. & Ganguly, A. *Sparse group lasso: Consistency and climate applications* in *Proceedings of the 2012 SIAM International Conference on Data Mining* (2012), 47–58.

13. Liu, J., Wang, K., Ma, S. & Huang, J. Accounting for linkage disequilibrium in genome-wide association studies: A penalized regression method. *Statistics and its interface* **6,** 99 (2013).

14. Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference, and prediction* (Springer Science & Business Media, 2009).

15. Karpatne, A., Watkins, W., Read, J. & Kumar, V. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431* (2017).

16. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378,** 686–707 (2019).

17. Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A. & Ramakrishnan, N. *Incorporating Prior Domain Knowledge into Deep Neural Networks* in *2018 IEEE International Conference on Big Data (Big Data)* (2018), 36–45.

18. De Wolff, T., Carrillo, H., Martí, L. & Sanchez-Pi, N. Towards Optimally Weighted Physics-Informed Neural Networks in Ocean Modelling. *arXiv preprint arXiv:2106.08747* (2021).

19. Liu, D. & Wang, Y. Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling. *Journal of Mechanical Design* **141** (2019).

20. Bellgran, M. & Säfsten, E. K. *Production development: design and operation of production systems* `https://www.springer.com/gb/book/9781848824942` (Springer Science & Business Media, 2009).

21. Chryssolouris, G. *Manufacturing systems: theory and practice* ISBN: 978-1-4419-2067-6. `https://www.springer.com/gb/book/9780387256832` (Springer Science & Business Media, 2013).

22. Ishikura, H. New product development and planning. *International Journal of Manufacturing Technology and Management* **3,** 238–250. ISSN: 1741-5195 (2001).

23. Hendricks, K. B. & Singhal, V. R. The effect of product introduction delays on operating performance. *Management Science* **54,** 878–892 (2008).

24. Conner, B. P. *et al.* Making sense of 3-D printing: Creating a map of additive manufacturing products and services. *Additive Manufacturing* **1,** 64–76 (2014).

25. Tofail, S. A. *et al.* Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials Today* (2017).

26. Gao, W. *et al.* The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design* **69,** 65 –89. ISSN: 0010-4485 (2015).

27. Clymer, D. R., Cagan, J. & Beuth, J. Power–velocity process design charts for powder Bed additive manufacturing. *Journal of Mechanical Design* **139,** 100907 (2017).

28. Dass, A. & Moridi, A. State of the art in directed energy deposition: From additive manufacturing to materials design. *Coatings* **9,** 418 (2019).

29. Galarraga, H., Lados, D. A., Dehoff, R. R., Kirka, M. M. & Nandwana, P. Effects of the microstructure and porosity on properties of Ti-6Al-4V ELI alloy fabricated by electron beam melting (EBM). *Additive Manufacturing* **10,** 47–57 (2016).

30. Wang, W., Ning, J. & Liang, S. Y. Prediction of lack-of-fusion porosity in laser powder-bed fusion considering boundary conditions and sensitivity to laser power absorption. *The International Journal of Advanced Manufacturing Technology* **112,** 61–70 (2021).

31. Ge, P *et al.* An integrated modeling of process-structure-property relationship in laser additive manufacturing of duplex titanium alloy. *International Journal of Thermal Sciences* **140,** 329–343 (2019).

32. Childs, C. M. & Washburn, N. R. Embedding domain knowledge for machine learning of complex material systems. *MRS Communications* **9,** 806–820 (2019).

33. Sood, A. K., Ohdar, R. K. & Mahapatra, S. S. Parametric appraisal of fused deposition modelling process using the grey Taguchi method. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **224,** 135–145. eprint: `https://doi.org/10.1243/09544054JEM1565`. `https://doi.org/10.1243/09544054JEM1565` (2010).

34. Chen, H. & Zhao, Y. F. *Learning Algorithm Based Modeling and Process Parameters Recommendation System for Binder Jetting Additive Manufacturing Process* in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2015).

35. Shen, X., Yao, J., Wang, Y. & Yang, J. *Density prediction of selective laser sintering parts based on artificial neural network* in *International Symposium on Neural Networks* (2004), 832–840.

36. Garg, A, Tai, K & Savalani, M. State-of-the-art in empirical modelling of rapid prototyping processes. *Rapid Prototyping Journal* **20,** 164–178 (2014).

37. Liu, R., Liu, S. & Zhang, X. A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing. *The International Journal of Advanced Manufacturing Technology* **113,** 1943–1958 (2021).

38. Snow, Z., Reutzel, E. W. & Petrich, J. Correlating in-situ sensor data to defect locations and part quality for additively manufactured parts using machine learning. *Journal of Materials Processing Technology* **302,** 117476 (2022).

39. Atwya, M. & Panoutsos, G. Structure optimization of prior-knowledge-guided neural networks. *Neurocomputing* **491,** 464–488 (2022).

40. Schober, P., Boer, C. & Schwarte, L. A. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia* **126,** 1763–1768 (2018).

41. Puth, M.-T., Neuhäuser, M. & Ruxton, G. D. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits. *Animal Behaviour* **102,** 77–84 (2015).

42. Edelmann, D., Móri, T. F. & Székely, G. J. On relationships between the Pearson and the distance correlation coefficients. *Statistics & Probability Letters* **169,** 108960 (2021).

43. Jafari-Marandi, R., Khanzadeh, M., Tian, W., Smith, B. & Bian, L. From in-situ monitoring toward high-throughput process control: cost-driven decision-making framework for laser-based additive manufacturing. *Journal of Manufacturing Systems* **51,** 29–41 (2019).

44. Tian, Q., Guo, S., Melder, E., Bian, L., Guo, W., *et al.* Deep learning-based data fusion method for in situ porosity detection in laser-based additive manufacturing. *Journal of Manufacturing Science and Engineering* **143** (2021).

45. DebRoy, T. *et al.* Additive manufacturing of metallic components–process, structure and properties. *Progress in Materials Science* **92,** 112–224 (2018).

46. Blakey-Milner, B. *et al.* Metal additive manufacturing in aerospace: A review. *Materials & Design* **209,** 110008 (2021).