How to Select Physics-Informed Neural Networks in the Absence of Ground Truth: A Pareto Front-Based Strategy

Zhao Wei* [1 2] **Jian Cheng Wong*** [2 3] **Nicholas Wei Yong Sung** [1 2] **Abhishek Gupta** [4] **Chin Chun Ooi** [1 2]
Pao-Hsiung Chiu [2] **My Ha Dao** [2] **Yew-Soon Ong** [3]

Abstract

Physics-informed neural networks (PINNs) have been proposed as a potential route to inverse modelling or mesh-free alternative to numerical methods for partial differential equations (PDEs). However, these problems typically lack ground truth, making selection of more accurate PINN models difficult, especially with processes such as hyper-parameter tuning. This is exacerbated as PINNs need to balance multiple objectives, comprising the governing PDEs, associated boundary/initial conditions, and/or point data. Under this multi-objective optimization framework, the ideal PINN solution is one that achieves zero loss across all components although this is not typical, resulting in a Pareto set of models. Nonetheless, there are objectively-preferred models based on congruence to unknown ground truth. In this context, we propose a Pareto front-based analysis to help identify better performing models. First, an approximation to the Pareto set of solutions with minimal PINN loss is constructed for different balances of loss weights. A loss weight located on the convex bulge of the Pareto front is then selected to rescale the training loss across all solutions. Across our experiments, this rescaling demonstrates a strong correlation between the rescaled PINN loss and mean squared error (MSE) relative to simulated ground truth, thereby illustrating potential PINN model selection.

*Equal contribution [1]Centre for Frontier AI Research, Agency for Science, Technology and Research, 1 Fusionopolis Way, 16-16 Connexis, Singapore [2]Institute of High Performance Computing, Agency for Science, Technology and Research, 1 Fusionopolis Way, 16-16 Connexis, Singapore [3]School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore [4]School of Mechanical Sciences, Indian Institute of Technology, Goa, India. Correspondence to: Zhao Wei <Wei_Zhao@cfar.a-star.edu.sg>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023.

1. Introduction

Deep neural networks (DNNs) are able to model complex relationships, especially in domains such as computer vision and natural language processing where massive data-sets are available (Nguyen et al., 2015; Otter et al., 2020). However, such large data-sets are not available for most physical systems, and use of physical laws and prior physical knowledge can be important to improve model performance (Cuomo et al., 2022; Elhamod et al., 2022). In particular, PINNs have been proposed as a means of synergizing data and theory for solving many engineering problems, especially inverse problems, in recent years (Raissi et al., 2019; Das & Tesfamariam, 2022; Karniadakis et al., 2021; Cai et al., 2021).

In prior work, PINNs have been used to model PDEs governing real-world physical phenomena by training DNNs to minimize a physics-informed loss comprising initial conditions (ICs), boundary conditions (BCs), and the governing PDEs itself (Mao et al., 2020; Chiu et al., 2022). PINN training has correspondingly been formulated as a multi-objective problem (Jin, 2006), as the satisfaction of ICs, BCs, and PDEs can be treated as separate objectives (Heldmann et al., 2023; De Wolff et al., 2022; Rohrhofer et al., 2021; Bischof & Kraus, 2021; Strelow et al., 2023). However, in most practical scenarios, the ground truth is not *a priori* known. Hence, the quality of any trained PINN solution can only be judged based on the training loss (Peng et al., 2020). As the PINN training loss is comprised of multiple components, the trade-off between different loss components is highly complex, and prior literature has shown that PINNs are susceptible to a mis-match between training loss and predictive performance. A lower training loss does not necessarily imply a PINN solution with lower MSE relative to the ground truth (Gao et al., 2022; Fang, 2022). This bears similarity to a common machine learning problem in which one needs to evaluate model performance in the absence of ground truth (Veldanda et al., 2023). Notably, this issue is compounded when more components exist, e.g., multiple governing equations, or the inverse formulation, whereby one might have data for various field variables.

While some have analyzed the Pareto front and convergence

performance of PINN solutions, including the impact of hyper-parameters (e.g., loss weights, learning rate, activation function) (Rohrhofer et al., 2021), no prior work to our knowledge has systematically evaluated the extent to which PINN models with the lowest training loss can be assumed to provide the lowest MSE relative to ground truth. This is a key hurdle in PINN model selection, especially given how empirical hyper-parameter tuning is a common part of training.

Hence, in this work, we propose a strategy for comparing PINN model performance based on insights from constructing a Pareto front during PINN training. This can be useful for selecting the best model hyper-parameters during training. Note that our proposed strategy is complementary to the self-adaptive loss weighting schemes for PINNs (McClenny & Braga-Neto, 2020; Wang et al., 2021) because those strategies improve training convergence, but do not address model selection in the context of a mis-match between training loss and model performance. Analysis of correlations of training loss and MSE from PINN models trained using different hyper-parameters show that the Pareto front can provide insights to maximize the correlation between PINN training loss and MSE, thereby minimizing the impact of any mismatch between training loss and MSE in the absence of ground truth.

2. Problem Setup

2.1. Physics-Informed Neural Networks

The PINN training loss ($\mathcal{L}_{\text{PINN}}$) consists of a weighted sum of three parts, IC loss ($\mathcal{L}_{\text{IC}}$), BC loss ($\mathcal{L}_{\text{BC}}$), and PDE residual loss ($\mathcal{L}_{\text{PDE}}$):

$$
\begin{aligned}
\mathcal{L}_{\text{PINN}} &= \lambda_{\text{IC}}\mathcal{L}_{\text{IC}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}} + \lambda_{\text{PDE}}\mathcal{L}_{\text{PDE}} \\
\mathcal{L}_{\text{IC}} &= \frac{1}{n_{\text{IC}}}\sum_{i=1}^{n_{\text{IC}}}\left|u_\theta\left(x_{\text{IC}}^{(i)}, t_{\text{IC}}^{(i)}\right) - u_{\text{IC}}^{(i)}\right|^2 \\
\mathcal{L}_{\text{BC}} &= \frac{1}{n_{\text{BC}}}\sum_{i=1}^{n_{\text{BC}}}\left|u_\theta\left(x_{\text{BC}}^{(i)}, t_{\text{BC}}^{(i)}\right) - u_{\text{BC}}^{(i)}\right|^2 \\
\mathcal{L}_{\text{PDE}} &= \frac{1}{n_{\text{PDE}}}\sum_{i=1}^{n_{\text{PDE}}}\left|f_\theta\left(x_{\text{PDE}}^{(i)}, t_{\text{PDE}}^{(i)}\right)\right|^2
\end{aligned}
\tag{1}
$$

where λ_{IC}, λ_{BC}, and λ_{PDE} are the weights that determine the balance between IC, BC, and PDE residual; $u_\theta(x,t)$ is the PINN prediction at point location (x,t); $\left\{\left(x_{\text{BC}}^{(i)}, t_{\text{BC}}^{(i)}\right), u_{\text{BC}}^{(i)}\right\}_{i=1}^{n_{\text{BC}}}$ and $\left\{\left(x_{\text{IC}}^{(i)}, t_{\text{IC}}^{(i)}\right), u_{\text{IC}}^{(i)}\right\}_{i=1}^{n_{\text{IC}}}$ are the specified BC and IC; and $\left\{x_{\text{PDE}}^{(i)}, t_{\text{PDE}}^{(i)}\right\}_{i=1}^{n_{\text{PDE}}}$ are the collocation points for calculating the PDE residuals $f_\theta(x,t)$.

2.2. PINN Hyper-Parameter Tuning Problem

Optimal selection of hyper-parameters (e.g., loss weights, learning rate) are required for good PINN model performance (Yan et al., 2022). Using the Korteweg-De Vries (KdV) problem as an example (a full description of the problem is provided in Appendix A), different hyper-parameters need to be evaluated during PINN training. Examples of these hyper-parameters are listed in Table 1.

Table 1. Hyper-parameter settings of PINN trained for a KdV problem.

PARAMETER	VALUE
PINN ARCHITECTURE	(x,t)-n_N-n_N-n_N-n_N-u
NODES OF EACH HIDDEN LAYER n_N	$\{5, 10, 20\}$
ACTIVATION FUNCTION	$\{\text{ReLU}, \text{Sigmoid}, \text{Tanh}\}$
PDE LOSS WEIGHT λ_{PDE}	$\{0.001, 0.01, 0.1, 1\}$
LEARNING RATE	$\{0.001, 0.01, 0.1\}$
MAXIMUM TRAINING ITERATION	120000
BATCH SIZE	15477 (IC: 77, PDE: 15400)

In Table 1, the numbers n_N's between input (x,t) and output u in the PINN architecture represent the number of nodes in each hidden layer, which can be varied. Similarly, the numbers in brackets corresponding to the batch size parameter represent the number of IC and PDE collocation points used during training, which can also be varied. Other hyper-parameters which can be varied include learning rate, choice of activation function, and the relative weights for the terms in the PINN training loss, e.g., λ_{PDE}. The value of λ_{IC} is set to 1 and there is zero BC term in this problem.

As shown in Eq. (1), a larger loss weight means putting more emphasis on the associated loss during training. Different combinations of loss weights can impact PINN training non-linearly. Moreover, it can be difficult to compare different PINN models, including with different hyper-parameter settings, based on their training loss across different loss weights. In particular, construction of the PINN training loss with different λ_{PDE} complicates the determination of a PINN model with minimal MSE based solely on their individual training loss. In this example, the Spearman correlation coefficient between PINN training loss and MSE across all the models (trained with different hyper-parameters) is 0.8454 and can be improved. Thus, it is helpful to have a strategy that can utilize a common loss term with a single λ_{PDE} for consistent comparison across all potential PINN models (and hyper-parameter combinations).

3. Pareto Front-Based Model Selection

As PINN models can be trained with different combination of loss weights, an approximation to the Pareto set of potential solutions with minimal PINN training loss across different loss weights can be constructed to identify high-quality non-dominated solutions with various balances of IC, BC, and PDE. According to Rohrhofer et al. (2021), the mapping between loss weights and the corresponding solution on the Pareto set is well-behaved when the Pareto front is convex, as small changes of loss weights have little influence on the converged solutions. Thus, *it is better to use a single loss weight, based on an analysis of solutions with minimal training loss on the convex bulge of the Pareto front, as a common basis for comparison across all PINN models, regardless of their training hyper-parameters*. The correlation between this rescaled training loss and MSE of all the solutions is calculated and discussed in Section 4, indicating that a PINN loss computed with this single rescaled loss weight can differentiate the models with lower MSE. Thus, this strategy allows one to compare across different PINN models in the absence of any ground truth, including in determining possible optimal hyper-parameters for training a PINN model.

The flowchart of the proposed Pareto front-based model selection strategy is shown in Figure 1 and described below:

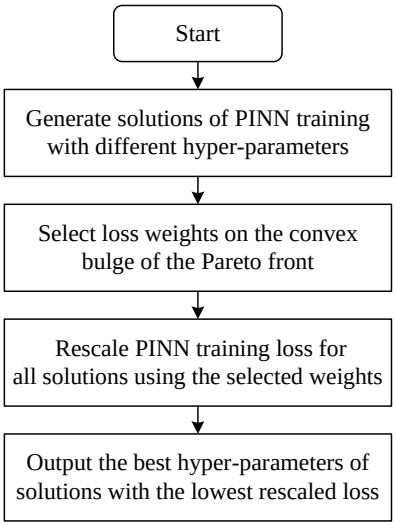


Figure 1. Flowchart of the proposed Pareto front-based model selection strategy.

Step 1: PINNs with different hyper-parameters are trained to generate potential solutions.

Step 2: A Pareto set of solutions with minimal PINN training loss for each combination of loss weights is constructed. Loss weights for PINN solutions with minimal training loss that are also located on the convex bulge of the Pareto front are selected as a common rescaling factor.

Step 3: A PINN loss for all generated solutions from Step 1 is re-computed per Eq. (2), based on the selected value from Step 2 (i.e., $\lambda_{\text{IC\_rescaled}}$, $\lambda_{\text{BC\_rescaled}}$, and $\lambda_{\text{PDE\_rescaled}}$).

$$\mathcal{L}_{\text{PINN\_rescaled}} = \lambda_{\text{IC\_rescaled}}\mathcal{L}_{\text{IC}} + \lambda_{\text{BC\_rescaled}}\mathcal{L}_{\text{BC}} + \lambda_{\text{PDE\_rescaled}}\mathcal{L}_{\text{PDE}} \quad (2)$$

Step 4: Solutions with the lowest $\mathcal{L}_{\text{PINN\_rescaled}}$ from Step 3 are the potential best models with good hyper-parameters.

4. Results

4.1. Strategy Effectiveness for Varied Hyper-Parameters in KdV Equation

As per the problem described in Section 2, multiple models trained with different hyper-parameters are presented to validate the effectiveness of the proposed Pareto front-based strategy for selecting the best-performing models. In the construction of the approximate Pareto front, all models obtained are considered, regardless of the difference being stochasticity in the neural network training process, or differences in individual hyper-parameters. A scatter-plot of $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{PDE}}$ from PINN models trained with different hyper-parameters is shown in Figure 2.

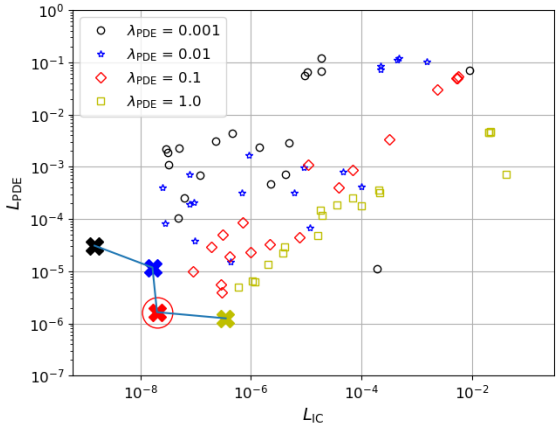


Figure 2. Distribution of $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{PDE}}$ for PINN models trained with different hyper-parameters for KdV equation. The Pareto front of solutions with minimal PINN training loss across different λ_{PDE} ($\lambda_{\text{IC}} = 1$) is plotted as a solid line. The cross with specific color represents the solution with minimal PINN training loss across all models for different combinations of loss weights. The red circle represents the solution with minimal MSE across all the models.

It can be seen from Figure 2 that the solution with PINN training loss trained with $\lambda_{\text{PDE}} = 0.1$ is on the convex

Table 2. PINN architecture and training configurations for test problems.

PROBLEM	PINN ARCHITECTURE	BATCH SIZE	MAXIMUM TRAINING ITERATION	LEARNING RATE
CONVECTION-DIFFUSION EQUATION	x-10-10-10-u	10000 (BC: 2, PDE: 9998)	100000	0.001
LINEARIZED BURGERS' EQUATION	(x,t)-10-10-10-u	51657 (IC: 257, PDE: 51400)	100000	0.1
KDV EQUATION	(x,t)-8-8-8-8-u	15477 (IC: 77, PDE: 15400)	120000	0.01
LID-DRIVEN CAVITY	(x,y)-40-20-20 -[20-20-20-u, 20-20-20-v, 20-20-20-p]	2700 (BC: 200, PDE: 2500)	100000	0.005

Notes: The numbers between input and output represent the number of nodes in each hidden layer. For example, x-10-10-10-u means the PINN architecture has three hidden layers with 10 nodes between its input x and output u. For the first three benchmarks, the Tanh activation function with Glorot uniform distribution for weights initialization is used while the learning rate is fixed (Sung et al., 2022). For the lid-driven cavity, the Sin activation function with He uniform distribution for weights initialization is used while a cosine decay learning rate is used with an initial value of 0.005. A linear activation function is used in the output layer for all problems. The numbers in brackets for batch size represent the number of IC or BC and PDE collocation points. Full batch is used for training. All PINN models are implemented in the JAX framework and are run on an Intel Xeon Gold 6336Y 2.40GHz CPU workstation with a NVIDIA RTX A5000 GPU.

bulge of the Pareto front of all solutions with minimal PINN training loss trained across different λ_{PDE}. This particular value is deemed effective at balancing both $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{PDE}}$ for this problem. Thus, $\lambda_{\text{PDE}} = 0.1$ is used for rescaling the solutions obtained from different hyper-parameters.

When λ_{PDE} is 0.1, the rescaled PINN loss and MSE have high Spearman correlation coefficient of **0.9841**, which is a large increase (**16.41%**) from the previous value of **0.8454**. In addition, for this particular example, the PINN solution with minimal MSE has the same λ_{PDE} as that selected through our proposed strategy, further emphasizing the effectiveness of the strategy for model selection.

4.2. Comparisons for Different Loss Weights

We further seek to demonstrate the generalizability of the proposed Pareto front-based strategy on a wider range of problems. We thus select four problems spanning different physics (i.e., convection-diffusion equation, linearized Burgers' equation, KdV equation, and lid-driven cavity) (Sung et al., 2022; Chiu et al., 2022; Wong et al., 2021) and evaluate its ability to identify better models based on the correlation between the computed rescaled loss and the MSE (relative to ground truth obtained from high fidelity numerical simulations). Additional details on the physics for these problems are provided in Appendix A while the PINN architecture and training configurations are listed in Table 2.

This emulates a practical scenario whereby one seeks to tune the loss weights (i.e., λ_{BC} (λ_{IC}) and λ_{PDE} in Eq. (1) to find a more accurate PINN model. λ_{BC} (λ_{IC}) is set to 1, and λ_{PDE} spans 1e-3 to 1e3 (which is a sufficiently broad range according to our prior experience). Five different initializations are used for training under each setting and the distributions of $\mathcal{L}_{\text{BC}}$ ($\mathcal{L}_{\text{IC}}$) and $\mathcal{L}_{\text{PDE}}$ for the lid-driven cavity problem are shown in Figure 3. Similar figures for the other problems are presented in Appendix B Figure 5.

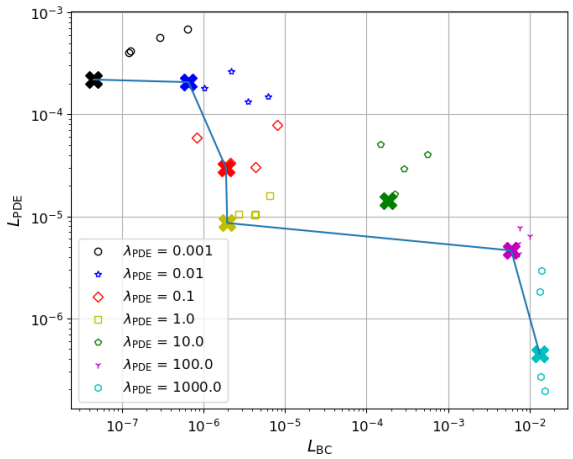


Figure 3. Loss distribution of different runs for the lid-driven cavity problem. The cross represents the solution with minimal PINN loss across different runs for each λ_{PDE}.

The Pareto front of solutions with minimal PINN loss across each combination of loss weights is plotted as a solid line in Figure 3. Next, the loss weight corresponding to the minimal PINN training loss that is located on the convex Pareto front is selected for rescaling the PINN training loss across all training runs. These loss weights are selected as they provide the best balance of $\mathcal{L}_{\text{BC}}$ ($\mathcal{L}_{\text{IC}}$) and $\mathcal{L}_{\text{PDE}}$. In the neighborhood of the convex bulge, a small improvement in one of the loss components is typically accompanied by a large

Table 3. Comparison of Spearman correlation coefficients obtained between MSE and the proposed rescaled PINN loss (rescaled-PINN), unscaled PINN loss (unscaled-PINN), and residual (Res).

PROBLEM	CORRELATION (RESCALED-PINN)	CORRELATION (UNSCALED-PINN)	CORRELATION (RES)
CONVECTION-DIFFUSION EQUATION	0.9546	0.8840	-0.4597
LINEARIZED BURGERS' EQUATION	0.9762	0.9422	-0.8561
KDV EQUATION	0.9737	0.6927	-0.9692
LID-DRIVEN CAVITY	0.9227	0.4546	-0.4773

deterioration in the other. Based on the constructed Pareto fronts, the selected λ_{PDE} values for convection-diffusion equation, linearized Burgers' equation, KdV equation and lid-driven cavity problem are determined to be 0.01, 0.1, 0.1, and 1, respectively.

Spearman correlation coefficients are calculated and reported in Table 3, and illustrate the improvement in correlation when a single, carefully chosen $\lambda_{\text{PINN\_rescaled}}$ is used for rescaling. This improvement is particularly large for the more non-linear problems (e.g. KdV and Lid-driven cavity). In addition, we compare the results obtained using a residual convergence metric typically used in numerical methods to assess convergence. This was done by using the PINN to output predictions across a regular grid and evaluating the residual with a finite difference-based stencil. The correlation between PINN loss and MSE is similarly listed in Table 3. While this strategy also requires no *a priori* ground truth, we note that the correlations obtained remain poor, rendering this strategy unsuitable for PINNs. Further details are provided in Appendix C.

While we propose a Pareto front-based selection strategy in this work, we can also calculate the loss for all generated PINN models using a single, chosen λ_{PDE} value as per Eq. (2) based on a single arbitrary value. Hence, the Spearman correlation coefficients for 7 different loss computations (spanning 1e-3 to 1e3) and MSE are plotted in Figure 4 for the lid-driven cavity problem. It can be seen from Figure 4 that it is important to choose an appropriate scaling factor to maximize correlation between the PINN loss and MSE, thereby further illustrating the importance of our Pareto front-based model selection approach.

From our experiments, the proposed strategy allows better PINN models with lower MSE to be identified across multiple problems. Hyper-parameters of selected solutions can potentially also be optimal for the problems.

5. Conclusions

In this work, a Pareto front-based PINN model selection strategy is proposed which can be used for purposes such as identifying potential optimal hyper-parameters. Building on the multi-objective nature of the PINN training, a

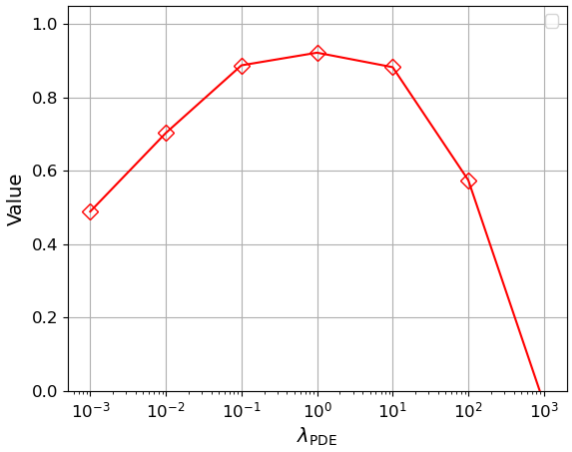


Figure 4. Correlation between rescaled PINN loss and MSE for the lid-driven cavity problem.

Pareto set of possible PINN models trained across different settings with minimal loss can be constructed. Solutions that map to the convex bulge of the Pareto front, which are relatively more well-behaved, suggest potential loss weights that best balance the various components of the PINN training loss. This can then be used to construct a rescaled loss across all the PINN models for selection of better PINN models or optimal hyper-parameters. Our results show that the rescaled PINN loss is indeed highly correlated with the MSE relative to simulated ground truth, and that similarly good correlations are observed for all four benchmark problems tested (i.e., convection-diffusion equation, linearized Burgers' equation, KdV equation, and lid-driven cavity). To conclude, our study showed that the proposed Pareto front-based model selection strategy is effective for attaining better PINN models in the absence of *a priori* knowledge of the ground truth. Nonetheless, we note that this is a first exploration of the possibility for identifying better performing models using insights from the Pareto front and that further extensions to reduce the computational cost and handle more complicated multi-component objective functions are still critically needed.

Broader Impact

Model selection and hyper-parameter tuning of PINNs are a tough task because PINNs are typically used in situations without ground truth, complicating evaluation of PINN model performance. This strategy is expected to be particularly useful as PINNs have also frequently been proposed as a potential mesh-free alternative to traditional numerical methods with particular promise for high-dimensional PDE problems. However, use of PINNs in these problems is problematic as model selection and hyper-parameter tuning must be done without access to ground truth. Thus, the selection of potential solutions with minimal MSE is a fundamental problem. This paper proposes a new strategy to select models based on maximizing the correlation between PINN training loss and MSE by leveraging the Pareto front generated by re-formulating PINNs as a multi-objective problem. Note that the correlation between PINN loss and MSE as obtained by our proposed strategy is consistently high across different problems according to the experimental results, which means that this approach may be generally applicable, although more verification needs to be done. In addition, the PINN loss of the experiments in this work only has two components, i.e., BC (IC) loss and PDE loss. When more components such as PDE losses of multiple governing equations and data loss are considered for training, the implementation of the strategy need to be further investigated.

Acknowledgments

This research is supported by A*STAR under the AME Programmatic project: Explainable Physics-based AI for Engineering Modelling & Design (ePAI) [Award No. A20H5b0142].

References

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

Cai, S. Z., Mao, Z. P., Wang, Z. C., Yin, M. L., and Karniadakis, G. E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

Chiu, P. H., Wong, J. C., Ooi, C. C., Dao, M. H., and Ong, Y. S. CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. *Computer Methods in Applied Mechanics and Engineering*, 395:114909, 2022.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

Das, S. and Tesfamariam, S. State-of-the-art review of design of experiments for physics-informed deep learning. *arXiv preprint arXiv:2202.06416*, 2022.

De Wolff, T., Lincopi, H. C., Martí, L., and Sanchez-Pi, N. MOPINNs: An evolutionary multi-objective approach to physics-informed neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 228–231, 2022.

Elhamod, M., Bu, J., Singh, C., Redell, M., Ghosh, A., Podolskiy, V., Lee, W. C., and Karpatne, A. CoPhy-PGNN: Learning physics-guided neural networks with competing loss functions for solving eigenvalue problems. *ACM Transactions on Intelligent Systems and Technology*, 13(6):1–23, 2022.

Fang, Z. W. A high-efficient hybrid physics-informed neural networks based on convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5514–5526, 2022.

Gao, H., Zahr, M. J., and Wang, J. X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, 2022.

Heldmann, F., Berkhahn, S., Ehrhardt, M., and Klamroth, K. PINN training using biobjective optimization: The trade-off between data loss and residual loss. *Journal of Computational Physics*, pp. 112211, 2023.

Jadoui, M., Blondeau, C., Martin, E., Renac, F., and Roux, F.-X. Comparative study of inner–outer Krylov solvers for linear systems in structured and high-order unstructured CFD problems. *Computers & Fluids*, 244:105575, 2022.

Jin, Y. C. *Multi-objective machine learning*, volume 16. Springer Science & Business Media, 2006.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S. F., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Mao, Z., Jagtap, A. D., and Karniadakis, G. E. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.

McClenny, L. and Braga-Neto, U. Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*, 2020.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

Otter, D. W., Medina, J. R., and Kalita, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2020.

Peng, W., Zhou, W. E., Zhang, J., and Yao, W. Accelerating physics-informed neural network training with prior dictionaries. *arXiv preprint arXiv:2004.08151*, 2020.

Raissi, M., Perdikaris, P., and 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.

Rohrhofer, F. M., Posch, S., and Geiger, B. C. On the pareto front of physics-informed neural networks. *arXiv preprint arXiv:2105.00862*, 2021.

Strelow, E. L., Gerisch, A., Lang, J., and Pfetsch, M. E. Physics informed neural networks: A case study for gas transport problems. *Journal of Computational Physics*, 481:112041, 2023.

Sung, N. W. Y., Wong, J. C., Chiu, P. H., Gupta, A., Ooi, C. C., and Ong, Y. S. Jax-accelerated neuroevolution of physics-informed neural networks: Benchmarks and experimental results. *arXiv preprint arXiv:2212.07624*, 2022.

Veldanda, A. K., Brugere, I., Dutta, S., Mishler, A., and Garg, S. Hyper-parameter tuning for fair classification without sensitive attribute access. *arXiv preprint arXiv:2302.01385*, 2023.

Wang, S. F., Teng, Y. J., and Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., et al. High-order CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

Wong, J. C., Gupta, A., and Ong, Y. S. Can transfer neuroevolution tractably solve your differential equations? *IEEE Computational Intelligence Magazine*, 16(2):14–30, 2021.

Yan, C., Vescovini, R., and Dozio, L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. *Computers & Structures*, 265:106761, 2022.

A. Definition of Problems

A.1. Korteweg–De Vries (KdV) Equation

KdV equation is a nonlinear partial differential equation which describes the evolution of phenomena such as wave propagation. The differential equation of the physical system and corresponding IC are described in Eq. (3) and Eq. (4), respectively.

$$\begin{aligned} &f = \frac{\partial u(x,t)}{\partial t} + v_1 u(x,t)\frac{\partial u(x,t)}{\partial x} + v_2 \frac{\partial^3 u(x,t)}{\partial x^3} = 0 \\ &v_1 = 1,\ v_2 = 0.001,\ x \in [0, 1.5],\ t \in [0, 2] \end{aligned} \tag{3}$$

$$\begin{aligned} &u(x,0) = \frac{3c_1}{\cosh^2[a_1(x - x_1)]} + \frac{3c_2}{\cosh^2[a_2(x - x_2)]} \\ &a_1 = \frac{1}{2}\sqrt{\frac{c_1}{v_2}},\ a_2 = \frac{1}{2}\sqrt{\frac{c_2}{v_2}},\ c_1 = 0.3,\ c_2 = 0.1, \\ &x_1 = 0.4,\ x_2 = 0.8,\ v_1 = 1,\ v_2 = 0.001 \end{aligned} \tag{4}$$

A.2. Convection-Diffusion Equation

The steady-state convection-diffusion equation describes the steady-state behavior when particles, energy, or other physical quantities are transported in a convective-diffusive physical system. The differential equation of the physical system and corresponding BCs are described in Eq. (5) and Eq. (6), respectively.

$$\begin{aligned} &f = v\frac{\mathrm{d}u(x)}{\mathrm{d}x} - k\frac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} = 0 \\ &v = 6,\ k = 1,\ x \in [0, 1] \end{aligned} \tag{5}$$

$$u(0) = 0,\ u(1) = 1 \tag{6}$$

A.3. Linearized Burgers' Equation

Linearized Burgers' equation is a simplified partial differential equation that is common across multiple areas in applied mathematics. The differential equation of the physical system and corresponding IC are described in Eq. (7) and Eq. (8), respectively.

$$\begin{aligned} &f = \frac{\partial u(x,t)}{\partial t} + v_1\frac{\partial u(x,t)}{\partial x} - v_2\frac{\partial^2 u(x,t)}{\partial x^2} = 0 \\ &v_1 = 1,\ v_2 = 0.02,\ x \in [-1.5, 6.5],\ t \in [0, 2] \end{aligned} \tag{7}$$

$$u(x,0) = m\mathrm{e}^{-(kx)^2},\ k = 2,\ m = 10 \tag{8}$$

A.4. Lid-Driven Cavity

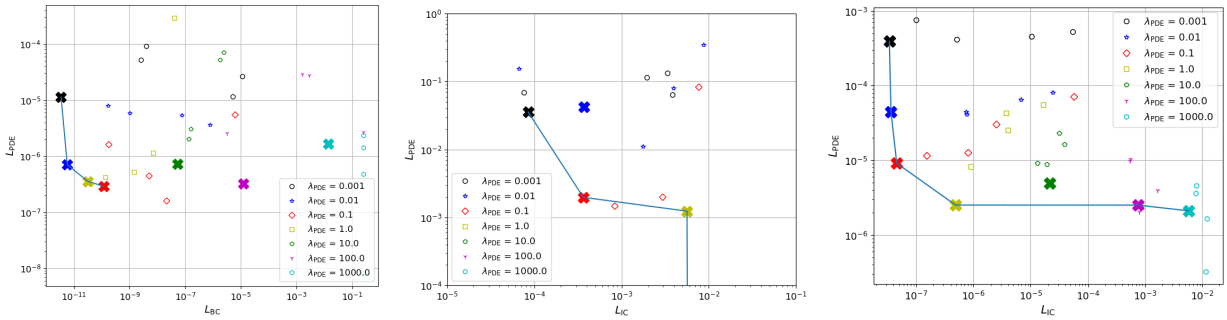
Lid-driven cavity is a well-known benchmark for viscous, incompressible fluid flow. The governing equations of lid-driven cavity are the steady-state, incompressible N-S equations. The differential equations of the physical system and corresponding BCs are described in Eq. (9) and Eq. (10), respectively

$$\begin{aligned} &f_1 = \frac{\partial u(x,y)}{\partial x} + \frac{\partial v(x,y)}{\partial y} = 0 \\ &f_2 = u(x,y)\frac{\partial u(x,y)}{\partial x} + v(x,y)\frac{\partial u(x,y)}{\partial y} + \frac{\partial p(x,y)}{\partial x} - \frac{1}{\mathrm{Re}}\Big[\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2}\Big] = 0 \\ &f_3 = u(x,y)\frac{\partial v(x,y)}{\partial x} + v(x,y)\frac{\partial v(x,y)}{\partial y} + \frac{\partial p(x,y)}{\partial y} - \frac{1}{\mathrm{Re}}\Big[\frac{\partial^2 v(x,y)}{\partial x^2} + \frac{\partial^2 v(x,y)}{\partial y^2}\Big] = 0 \\ &x \in [0, 1],\ y \in [0, 1] \end{aligned} \tag{9}$$

$$
\begin{aligned}
u(x,1) &= 1,\ v(x,1) = 0 \\
u(x,0) &= 0,\ v(x,0) = 0 \\
u(0,y) &= 0,\ v(0,y) = 0 \\
u(1,y) &= 0,\ v(0,y) = 0
\end{aligned}
\tag{10}
$$

where $\mathrm{Re} = 200$ is the Reynolds number.

B. Scatter-Plots for Section 4.2



(a) Convection-diffusion equation (b) Linearized Burgers' equation (c) KdV equation

Figure 5. Loss distributions for each test problem. The cross represents the solution with minimal PINN loss across the five different runs for each value of λ_{PDE}.

Based on the constructed Pareto fronts, the selected λ_{PDE} values for convection-diffusion equation, linearized Burgers' equation, KdV equation and lid-driven cavity problem are determined to be 0.01, 0.1, 0.1, and 1, respectively. For the linearized Burgers' equation problem, the PINN solution with the lowest MSE also corresponds to the λ_{PDE} as obtained by this Pareto front-based analysis, similar to the KdV problem. However, the PINN solution with the lowest MSE for the convection-diffusion and lid-driven cavity problems were obtained using a λ_{PDE} of 1.0 and 0.1 respectively.

C. Comparison to Residual Convergence Metric

To further illustrate the effectiveness of the proposed Pareto front-based model selection strategy, a residual-based convergence metric is also used for comparison. The residual is a conventional metric used to monitor convergence in numerical methods, including in domains such as computational fluid dynamics (Wang et al., 2013; Jadoui et al., 2022). When the residual decays to a small value, the solution is typically regarded as converged.

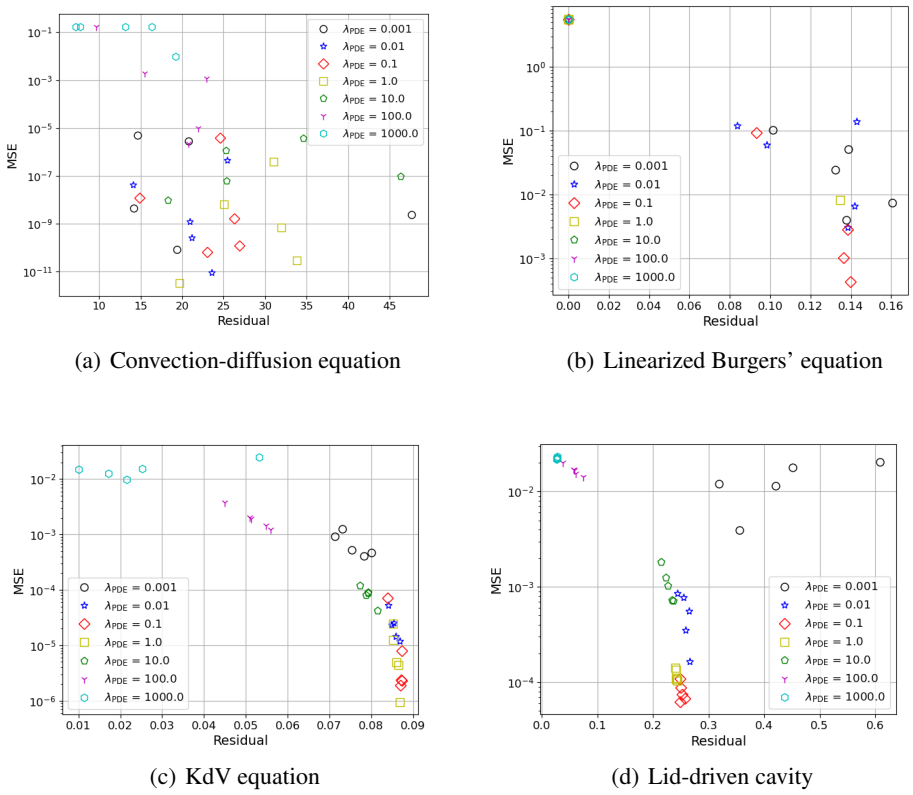
In this work, the residual of the physical system is calculated by Eq. (11)

$$
\begin{aligned}
&\text{Residual} = \frac{1}{n_s}\sum_{i=1}^{n_s}\left| f_{\text{residual}}\left(x_1, x_2, \cdots, x_d; \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \cdots, \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \cdots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \cdots\right)\right| \\
&\frac{\partial u}{\partial x_1} = \frac{u(x_1+\Delta h, x_2, \cdots, x_d) - u(x_1, x_2, \cdots, x_d)}{\Delta h} = D\_u(x_1) \\
&\frac{\partial^2 u}{\partial x_1^2} = \frac{D\_u(x_1+\Delta h) - D\_u(x_1)}{\Delta h} = DD\_u(x_1, x_1) \\
&\cdots
\end{aligned}
\tag{11}
$$

where n_s is the number of the discrete sample locations; $f_{\text{residual}}(\cdot)$ is the sum value of the ordinary or partial differential terms of the physical system in Eqs. (3), (5), (7), and (9), where the value of the ordinary or partial differential terms is calculated through the finite difference method using the PINNs' prediction (i.e., $u(x)$, $u(x,y)$, or $u(x,t)$).

The relationship between residual and MSE for all the solutions obtained for each benchmark problem are displayed in Figure 6. Spearman correlation coefficients obtained between MSE and the proposed rescaled PINN loss, unscaled PINN

loss, and residual are also calculated for comparison as listed in Table 3. As shown in Table 3, the correlation coefficients between rescaled PINN loss and MSE are high for all the benchmarks, whereas the unscaled PINN loss and MSE or the residual and MSE exhibit much lower correlations. This further illustrates the effectiveness of the proposed strategy for identifying solutions with lower error.



(a) Convection-diffusion equation

(b) Linearized Burgers' equation

(c) KdV equation

(d) Lid-driven cavity

Figure 6. Scatter-plot of residual and MSE for each benchmark problem.