Clustering Behaviour of Physics-Informed Neural Networks: Inverse Modeling of An Idealized Ice Shelf

Yunona Iwasaki
Department of Physics
Princeton University
Princeton, NJ 08544
yiwasaki@princeton.edu

Ching-Yao Lai
Department of Geosciences
Princeton University
Princeton, NJ 08544
cylai@princeton.edu

Abstract

We investigate the use of Physics-Informed Neural Networks (PINNs) for ice shelf hardness inversion, focusing on the effect of the relative weighting between equation and data components in the PINN objective function on its predictive performance. In the objective function we use a hyperparameter γ which adjusts the relative priority given to the fit of the PINN to known physical laws and its fit to the training data. We train the PINN with a range of γ, and training data with varying magnitudes of injected noise. We find that the PINN solutions converge to two different clusters in the prediction error space; one cluster corresponds to accurate, "low-error" solutions, while the other consists of "high-error" solutions that were likely trapped in a local minimum of the PINN objective function and fit poorly to the ground truth datasets. We call this the PINN *clustering behaviour*, which persists for a wide range of γ, noise level, and even with clean data. Using k-means clustering, we filter out the PINN solutions in the high-error clusters. The accuracy of the solutions in the low-error cluster varies with γ and the data noise. We find that the value of γ that minimizes the error of PINN-predicted ice hardness varies significantly with the data noise. With the optimal choice of γ, the PINN can remove the noise in the data and successfully predict the noise-free velocity, thickness and the ice hardness. The clustering phenomenon is observed for a wide range of parameter settings and is of practical, as well as theoretical interest.

1 Introduction

Ice shelves are the floating extensions of grounded ice sheets. They buttress upstream glaciers and play a crucial role in slowing ice flow into the ocean [4, 3, 6]. *Ice hardness* (B) is a fundamental material property that governs ice shelf dynamics, but cannot be directly measured. Here, we explore a novel approach, the Physics Informed Neural Network (PINN) [12], for hardness inversion of an idealized ice shelf. Physics Informed Neural Networks (PINN) [12] are a class of differential equation solvers that have demonstrated effectiveness in wide-ranging physical problems[16, 13, 1, 2] [11] [5] [14]. PINNs incorporate knowledge of the physical equations governing natural phenomena as a regularizer for a more traditional neural network predicting relations between variables of interest. Enforcing known physics equations during neural network training is demonstrated to reduce the amount of training data required [12, 13, 16] and to make PINNs more robust to noisy data [12, 13, 1]. PINNs work well when data points are available on an irregular grid [13, 9]. However, the effect of the relative weighting of the observations and the governing physics equations on the predictive performance of PINNs is poorly understood.

In this paper, we include a weighting parameter γ in the PINN objective function and investigate its role on PINN performance in the context of ice shelf modeling. We generated synthetic data by

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

specifying a hardness profile with an analytical solution for velocity $u(x)$ and thickness $h(x)$, then adding different levels of Gaussian noise. Then, we trained a PINN on clean and noisy datasets for a range of γ-values in order to quantify the relationship between γ, noise level, and predictive performance. This work is a step towards the optimization of PINNs for ice shelf hardness inversion under more realistic conditions; furthermore, it offers insight into the role of γ on the general PINN framework.

2 Governing equations of ice shelf flow

In this paper, we use a one-dimensional version of the Shallow-Shelf Approximation (SSA) [8, 10] to model an idealized problem in which the ice shelf spreads unidirectionally along the x-axis, i.e. the ice shelf dynamics are characterized by velocity $u(x)$, thickness $h(x)$, and ice hardness $B(x)$. Equations (1) summarize the final form of the equations and boundary conditions used to train the PINNs.

$$2\nu^\star B\left(\frac{du}{dx}\right)^{\frac{1}{n}} = h, \quad u(0) = 1, \quad h(0) = h_0. \tag{1}$$

B, u, and h have been rescaled by characteristic values of hardness, and thus represent dimensionless, velocity, and thickness, respectively [15]; $\nu^\star$ is a constant that encapsulates all of these re-scaling constants. These equations can be solved for analytically when $B(x) = k$, where k is a constant. A constant hardness profile is the simplest case for testing the application of PINN to hardness inversion, making it a natural starting point. Following the analytical solution derived by van der Veen (1986) [7, 17], we obtain the solution (ground truth) of velocity and thickness profiles for $B(x) = 1,$ and $\nu^\star = \frac{1}{2}$:

$$u(x) = \frac{h_0 + x}{h(x)}, \quad h(x) = \left(1 + \frac{h_0^{n+1}\left(h_0^{-n-1} - 1\right)}{\left(h_0 + x\right)^{n+1}}\right)^{-\frac{1}{n+1}} \tag{2}$$

3 PINN experiment setup

We are interested in the ability of a trained PINN to predict the correct velocity, thickness, and hardness profiles $u(x)$, $h(x)$, and $B(x)$, for any x in the spatial domain $x \in [0, 1]$, after being trained on finite, noisy observations of only $u(x)$ and $h(x)$ described by Equation (2). *We note that no data on $B(x)$ is given to the PINN during training;* thus, the PINN prediction of $B(x)$ is obtained purely by adding the physics constraints in the objective function (Equation (6)). We first generate a clean dataset $\{u(x_i), h(x_i)\}_{i=1}^{k}$ by evaluating (2) at $k = 401$ x-locations, equally spaced over the whole spatial domain. Then, we create a noisy dataset $\{u_{obs}(x_i), h_{obs}(x_i)\}_{i=1}^{k}$ (see details below).

Next, we trained a simple feedforward neural network $f_\Theta : \mathbb{R}^2 \to \mathbb{R}^3$ with two 5-unit hidden layers. f_Θ is trained with potentially noisy observations of velocity and thickness data [$u_{obs}(x_i)$, $h_{obs}(x_i)$] and generates a prediction [$\hat{u}(x_i)$, $\hat{h}(x_i)$, $\hat{B}(x_i)$].[1] The PINN objective function takes the form

$$J(\Theta) \equiv \gamma\, E(\Theta) + (1 - \gamma)\, D(\Theta) \tag{3}$$

where

$$D(\Theta) = \frac{1}{k}\sum_{i=1}^{k}\left(\left(\hat{u}(x_i) - u_{obs}(x_i)\right)^2 + \left(\hat{h}(x_i) - h_{obs}(x_i)\right)^2\right) \tag{4}$$

and

$$E(\Theta) = \frac{1}{c}\sum_{j=1}^{c}\left(\left|\left|\hat{h}_j - 2\nu^\star \hat{B}_j\left(\frac{d\hat{u}}{dx}\right)_j^{\frac{1}{n}}\right|\right|^2\right) \tag{5}$$

Note in Equation 5 we use the subscript j to denote the evaluation each quantity at *collocation point* $\{x_j\}_{j=1}^{c}$. Collocation points specify the coordinates at which to evaluate how accurately the current f_Θ obeys the known PDE.; they are not restricted to points $\{x_i\}_{i=1}^{k}$ in the domain where observations are available. In terms of $D(\Theta)$ and $E(\Theta)$, the PINN objective function is

$$J(\Theta) \equiv \gamma\, E(\Theta) + (1 - \gamma)\, D(\Theta) \tag{6}$$

[1] The source code is available at `https://github.com/YaoGroup/pinn_clusters`

We refer to $E(\Theta)$ as the *equation loss*, and $D(\Theta)$ as the *data loss* [12]. PINNs leverage automatic differentiation (autodiff) [12] to evaluate derivatives of $\hat{u}_\Theta(x, t)$ involved in the equation loss. Because the prediction of $B(x)$ relies on the physics equation (the $E(\Theta)$ term in the objective function (6)), we are interested in the role of the hyperparameter γ on the predictive performance of a PINN trained on datasets with varying *noise levels* (defined below). We introduce three new performance metrics B_{err}, u_{err}, and h_{err}:

$$B_{err} = \frac{1}{k}\sum_{i=1}^{k}\left|\left|\hat{B}(x_i) - B(x_i)\right|\right|^2, u_{err} = \frac{1}{k}\sum_{i=1}^{k}\left|\left|\hat{u}(x_i) - u(x_i)\right|\right|^2, h_{err} = \frac{1}{k}\sum_{i=1}^{k}\left|\left|\hat{h}(x_i) - h(x_i)\right|\right|^2 \tag{7}$$

where u, h, and B are ground truth values. We conducted systematic experiments in order to quantify the relationship between γ, *noise level*, and the three performance metrics defined above. We conducted 501 experimental "trials" for each combination of *noise level* and γ-value. An experimental trial consists of the following steps (1)-(5):

(1) Create a noisy dataset $\{u_{obs}(x_i), h_{obs}(x_i)\}_{i=1}^{k}$ by adding a value randomly drawn from a Gaussian distribution centered at 0 to each data point in the clean dataset. The level of added noise is adjusted by modifying the standard deviation of the Gaussian distribution. We call the standard deviation of the Gaussian distribution used to generate a dataset as its *noise level*. (2) Choose $c = 201$ collocation points $\{x_j\}_{j=1}^{c}$. The points were initially drawn uniformly over the spatial domain $x \in [0, 1]$, then cubed in order to bias the points toward $x = 0$, where the gradient of both $u(x)$ and $h(x)$ are large. We also tested experiments using $c = 1001$, which produced qualitatively similar results. (3) Randomly initialize the parameters of the PINN. (4) Train the PINN using the objective function for the specified value of γ. Training was continued until convergence, or a maximum of 400,000 iterations of gradient descent using Adam optimizer with learning rate 0.001, followed by 200,000 iterations of LBFGS optimizer. (5) Evaluate B_{err}, u_{err}, and h_{err} of the PINN prediction.

Six *noise levels* were tested: 0.0 (clean data), 0.01, 0.05, 0.1, 0.2, and 0.3. For each noise level, thirteen values of γ were tested (in total $\sim$39k trials). Because we are interested in the *relative* weighting of the equation and data loss, the gammas were chosen so that the ratios $\frac{\gamma}{1-\gamma}$ would be logarithmically spaced over the range $[10^{-4}, 10^{8}]$. We ran trials on the Princeton Della-CPU Cluster; one trial with a single value of gamma and noise level takes approximately 10 minutes to complete.

4 Results

Bimodal prediction errors We examined u_{err} vs. B_{err} and h_{err} vs. B_{err} for every trial with a given $\frac{\gamma}{1-\gamma}$ and *noise level*. Figure 1 presents the distributions of trial error for noise level = 0.01; however, our results were qualitatively similar for all noise levels, including the trials using the clean data set.

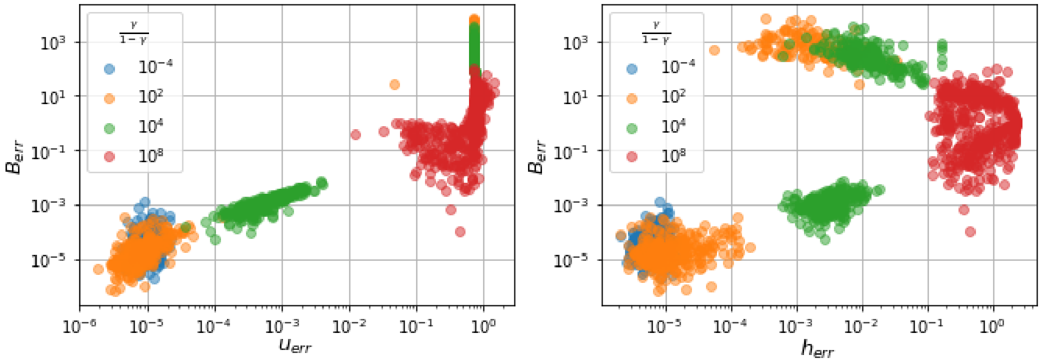

Figure 1: Correlation of B_{err} with u_{err} and h_{err} over 501 trials for various values of $\frac{\gamma}{1-\gamma}$ and noise level = 0.01. Left: u_{err} vs. B_{err}. Right: h_{err} vs. B_{err}.

Figure 1 suggests that B_{err}, u_{err}, and h_{err} are bimodally distributed on a logarithmic scale. The bimodal clustering behavior, with one cluster corresponding to very high error (greater than the average magnitude of the clean data) and the other corresponding to relatively low error, and very few trials in between, has not been systematically reported previously. Clustering persisted for repeated experiments with modifications such as significantly increased collocation points ($c = 1001$) and increased neural network width (up to 100 hidden units).

k-means clustering We quantitatively characterize the bimodal distribution of PINN prediction error using k-means clustering. For each combination of $\frac{\gamma}{1-\gamma}$ and noise level, we have a corresponding dataset $\{\epsilon_i\}_{i=1}^{501}$, $\epsilon_i = \langle u_{err_i}, h_{err_i}, B_{err_i} \rangle$, where u_{err_i}, h_{err_i}, and B_{err_i} denote the values of the performance metrics introduced in (7) for the i-th training trial. For all combinations of $\frac{\gamma}{1-\gamma}$ and noise level, we run k-means clustering with $k = 2$ in 3D space in terms of transformed coordinates $\langle \log_{10}(u_{err}), \log_{10}(h_{err}), \log_{10}(B_{err}) \rangle$. We denote data points assigned to the cluster center with the *lower* value of B_{err} as the *low-error cluster*; the data points assigned to the other cluster center are referred to as the *high-error cluster*. The PINN solutions in the low-error and high-error cluster roughly corresponds, respectively, to approximately < 10% and > 100% relative error in all three predictive variables.

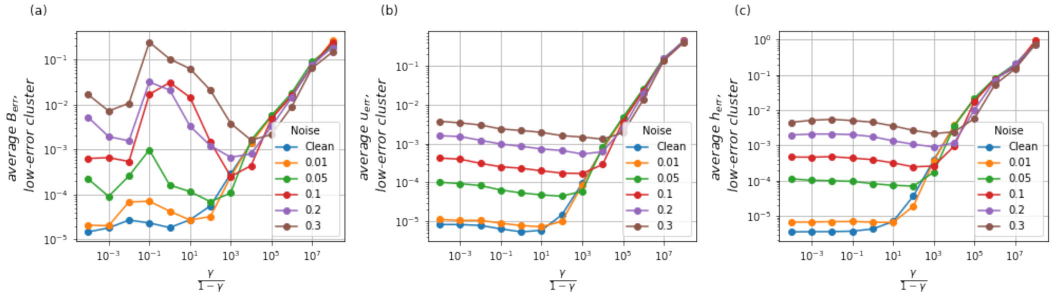

Figure 2: Average error in predictive variables for low-error clusters, for all noise levels and γ. (a) Average B_{err} in low-error clusters. (b) Average u_{err} in low-error clusters. (c) Average h_{err} in low-error clusters.

Low-error cluster statistics We focus our analysis on u_{err}, h_{err}, and B_{err} for low-error clusters. In Figure 1, B_{err}, u_{err}, and h_{err} appear to be correlated on a log-log scale; within the low-error cluster, small u_{err} and h_{err} correspond to small B_{err}, while large u_{err} and h_{err} correspond to large B_{err}. In Figure 2, we plot the average B_{err}, u_{err}, and h_{err} in low-error clusters for each tested noise level and γ. We observe that for all noise levels, average u_{err} and h_{err} in low-error clusters generally decrease as $\frac{\gamma}{1-\gamma}$ decreases, with predictive performance tapering for sufficiently small $\frac{\gamma}{1-\gamma}$. However, we notice that the plot of the low-error cluster means for B_{err} for each *noisy* dataset has a clear minimum.

5 Concluding Remarks

We systematically investigate the performance of PINNs applied to the inverse problem of 1D ice shelf flow. We discover an unexpected bimodal distribution (clustering) of PINN predictions in the prediction error space. There exists a "low-error mode" corresponding to PINN solutions with approximately < 10% relative error in all three predictive variables, and a "high-error mode" with approximately > 100% relative error. This bimodal behavior suggests that the loss landscape of the PINN objective function with a regularizer enforcing the governing equation has two local minima, one corresponding to relatively low predictive error, and the other corresponding to high predictive error. Using k-means clustering we successfully filter out the high-error solutions. The distribution of PINN prediction errors in the low-error cluster are highly dependent on $\gamma \in [0, 1]$ and the noise level of the data. The optimal weight of the equation loss $\gamma = \gamma^*_{low}$ that minimizes average error over low-error solutions depends strongly on the noise level of the training data. For data without noise and with noise level $= 0.3$, the optimal weights are $\gamma^*_{low} \approx 0.0001$ and $\gamma^*_{low} \approx 0.9999$, respectively. Weighting the equation and data loss components equally is a sufficient strategy only in the case of clean or very low-noise data; for datasets with higher levels of noise, equal weighting reduced the accuracy of the PINN predictions. Due to the broad applications of PINNs in scientific machine learning, significant work is needed to further explore the clustering behaviour of PINN predictions, the loss landscape of PINNs as a function of γ, and the dependence of optimal γ on the data noise.

6 Impact Statement

Understanding ice-shelf dynamics is not only a fundamental geophysical question, but an urgent area of research that is essential for predicting the effects of global warming on sea-level rise. We use physics-informed neural networks (PINNs) to predict ice hardness, a material property that governs ice shelf dynamics, but is challenging to measure. We show that, despite the absence of training data of ice hardness, PINNs can successfully infer ice hardness purely based on physical laws. In this paper, we systematically investigate the effect of the relative weighting of data and equation components of the PINN objective function in the context of an ice hardness inversion problem. With hundreds of trials for each experiment, we discovered an unexpected bimodal distribution of predictive errors across training trials (which we refer to as "clustering behavior") that persists for a wide range of the relative weight given to the equation loss, the number of collocation points, and PINN width. Thus, in addition to serving as a preliminary step for deep-learning based ice hardness inversion, our work highlights critical future research directions for successful PINN implementation in numerous potential scientific applications.

References

[1] S. Cai et al. "Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks". In: *Journal of Fluid Mechanics* 915 (2021).

[2] S. Cai et al. "Physics-informed neural networks (PINNs) for fluid mechanics: A review". In: *Acta Mechanica Sinica* (2022), pp. 1–12.

[3] J.J. Fürst et al. "The safety band of Antarctic ice shelves". In: *Nature Climate Change* 6.5 (2016), pp. 479–482.

[4] G.H. Gudmundsson. "Ice-shelf buttressing and the stability of marine ice sheets". In: *The Cryosphere* 7.2 (2013), pp. 647–655.

[5] E. Haghighat et al. "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 379 (2021), p. 113741.

[6] C.Y. Lai et al. "Vulnerability of Antarctica's ice shelves to meltwater-driven fracture". In: *Nature* 584.7822 (2020), pp. 574–578.

[7] D.R. MacAyeal. "EISMINT: Lessons in ice-sheet modeling". In: *Department of Geophysical Sciences, University of Chicago, Chicago, IL* 1832 (1997), p. 1839.

[8] D.R. MacAyeal. "Large-scale ice flow over a viscous basal sediment: Theory and application to ice stream B, Antarctica". In: *Journal of Geophysical Research: Solid Earth* 94.B4 (1989), pp. 4071–4087.

[9] Z. Mao, A.D. Jagtap, and G.E. Karniadakis. "Physics-informed neural networks for high-speed flows". In: *Computer Methods in Applied Mechanics and Engineering* 360 (2020), p. 112789.

[10] L. Morland. "Unconfined ice shelf flow". In: *Proceedings of Workshop on The Dynamics of the West Antarctic Ice Sheet (1987)*. 1987, pp. 99–116.

[11] G.P. Pun et al. "Physically informed artificial neural networks for atomistic modeling of materials". In: *Nature communications* 10.1 (2019), pp. 1–10.

[12] 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". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991.

[13] M. Raissi, A. Yazdani, and G.E. Karniadakis. "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations". In: *Science* 367.6481 (2020), pp. 1026–1030.

[14] M. Rasht-Behesht et al. "Physics-Informed Neural Networks (PINNs) for Wave Propagation and Full Waveform Inversions". In: *Journal of Geophysical Research: Solid Earth* 127.5 (2022), e2021JB023120.

[15] J. Sola and J. Sevilla. "Importance of input data normalization for the application of neural networks to complex industrial problems". In: *IEEE Transactions on Nuclear Science* 44.3 (1997), pp. 1464–1468. DOI: 10.1109/23.589532.

[16] L. Sun et al. “Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data”. In: *Computer Methods in Applied Mechanics and Engineering* 361 (2020), p. 112732.

[17] C.J. van der Veen. “Numerical modelling of ice shelves and ice tongues”. In: *Annales geophysicae. Series B. Terrestrial and planetary physics*. Vol. 4. 1. 1986, pp. 45–53.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **In Section 2, we explain the simplifications we used in order to implement a ice-shelf model compatible with the PINN framework.**
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **See footnote in Section 3.**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **This is explained in the description of an experimental trial in Section 3.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] **The premise of this work was the *distribution of results* over numerous (500+) repeated trials. Therefore, we account for uncertainty by characterizing the full distribution of results.**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] **See end of Section 3.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]