Adaptive collocation point sampling strategies for physics informed neural networks

J Florido*[1], P K Jimack[1], A Khan[2], H Wang[3]

[1] School of Computer Science, University of Leeds, Leeds, UK, LS2 9JT
[2] School of Civil Engineering, University of Leeds, Leeds, UK, LS2 9JT
[3] AI Centre, Department of Computer Science, University College London, London, UK, WC1E 6BT

Keywords: Physics-Informed Machine Learning, Adaptivity, Nonlinear PDEs, Convection-dominated problems

Abstract: Physics Informed Neural Networks (PINNs) have emerged as a machine learning (ML) framework designed to leverage prior knowledge of physics usually available in fluid dynamics and other scientific domains, in the form of well-established equations or systems of equations. These can be used to train the network in the form of a physics loss term which measures, and then attempts to minimise, the partial differential equation (PDE) residual, thereby solving the governing PDE. The PINNs approach provides greater flexibility by reducing the dependency on data and accurate boundary information, leveraging the additional physics loss term as measured at a set of collocation points distributed throughout the space-time domain. In this work, we explore improvements to the distribution of the set of collocation points via adaptive resampling, presenting different strategies used and focusing on their scalability as the number of spatial dimensions in the PDE increases. Results will be presented demonstrating potential advantages of adaptive sampling over a baseline that uses fixed collocation point distributions. We also show the limitations of using the residual alone for higher dimensional problems, for which this residual redistribution strategy is not always robust.

1. Introduction

In this paper, we describe Physics Informed Neural Networks (PINNs)[1], and how they can be used for solving nonlinear Partial Differential Equations (PDEs) by introducing prior knowledge of physics into a machine learning (ML) algorithm. A physics loss term is created by measuring how well the prediction satisfies the governing PDE (quantified as the ‘residual’) throughout the domain. This is done efficiently by computing the necessary derivatives of the solution via automatic differentiation at a finite number of ‘collocation points’ set throughout the domain, calculating the residual at each point individually. The sum of the squares of the residuals result in the overall physics loss term, which is minimsed through gradient-based optimisation as standard. Additional loss terms can be used for the boundary and initial conditions (soft constraints). However, for some problems these can also be resolved by imposing an output transformation on the output of the network, ensuring the solution is exactly satisfied at the boundaries (hard constraints).

A key challenge in training PINNs efficiently lies in the selection of these collocation points. Default implementations of PINNs distribute a fixed number of collocation points in a uniform random manner throughout the domain. However, not all collocation points contribute equally to the learning process, and certain areas of the domain may require more refined sampling than others to adequately learn the complex features of the solution [2]. Covering all the areas of complexity adequately thus requires either a large number of points to be used globally (increasing the computational cost proportionately), or manual placement of the points in areas known to be problematic for the training, thus motivating the research on methods to automatically find and select an appropriate distribution of points for training. [3] addresses this problem of distributions of points by adding new collocation points, and shows the impact on the network’s training efficiency and final accuracy. The distribution of the collocation points directly influences both the network’s training efficiency and final accuracy.

To optimize the selection of collocation points, two common approaches are used: adaptive weighting and adaptive resampling. Adaptive weighting methods maintain a fixed set of collocation points but assign different weights to them based on their relative importance for learning (See [4], [5], [6]). Adaptive resampling, like in [3], focuses on the method of distribution of the points within the domain. Studies have explored different strategies for optimizing collocation point selection. The simplest improvement is to employ quasi-random distributions, which generally outperforms the default uniform random distributions [7]. Another is to dynamically resample collocation points throughout the training process,

*Corresponding Author: J Florido, mn17jilf@leeds.ac.uk

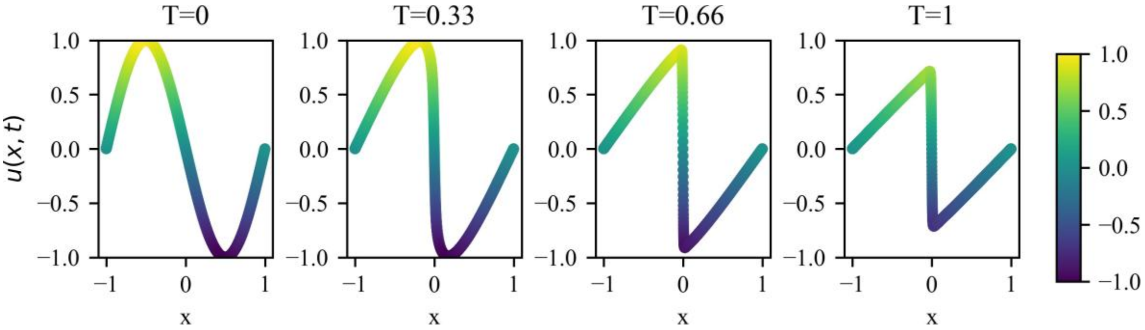

Figure 1: Representative solution of 1D Burgers' equation, evolving over time.

either randomly or with an adaptive bias towards areas where the residual is high ([3, 7, 8]). This resampling strategy adds minimal computational cost, as the number points can be kept consistent, while improving convergence.

Beyond residual-based sampling, additional heuristics have been investigated to guide collocation point selection. In particular, [9] explored alternative markers for resampling in one-dimensional PDEs, incorporating solution estimates and their derivatives alongside residual information. These findings suggest that adaptively refining the collocation point distribution based on geometric properties of the estimated solution can lead to more accurate PINN solutions with fewer collocation points, striking a balance between computational efficiency and accuracy.

The work presented here extends these ideas to a two-dimensional unsteady domain, applying them to nonlinear convection-diffusion problems. By leveraging adaptive point distribution strategies informed by solution geometry, we aim to further understand how different sampling methods interact with problem complexity and computational cost, offering insights into optimising PINN performance in more complex settings.

2. Methodology

2.1. Physics-Informed Neural Networks

Our approach builds on the adaptive collocation framework developed in [9] and applies it to solve 2D Burgers' equations without using training data (an unsupervised learning approach). The boundary conditions are enforced via hard constraints, and thus the loss function L(θ) (equation 1) that the neural network minimises consists only of the physics loss. This loss is computed at the set of collocation points $\mathcal{Y}$ located in the interior of the domain.

$$\mathrm{L} = \frac{1}{|\mathcal{Y}|}\sum_{x\in\mathcal{Y}} |f(x; u; u_x, u_t, u_{xx}; \theta)|^2 \tag{1}$$

For the 1D case (equation 2, representative solution shown in figure 1), an approximate solution for the dependent variable $u(x, t)$ dependent on the trainable network parameters (represented by θ) is obtained at every point x in $\mathcal{Y}$; the required gradients (u_x, u_t, u_{xx}) at each point are computed via automatic differentiation. For an inaccurate prediction, the terms on the left hand side of equation 2 may not equal zero; this is what we will refer to as 'residual', and is the physics loss at a given point that we hope to minimise(denoted by f (x; u; u_x, u_t, u_{xx}; θ) in equation 1). By training the network to approximately satisfy the PDE everywhere, the PINN obtains an accurate solution of $u(x, t)$ for our set domain and viscosity υ.

$$uu_x + u_t - \upsilon\, u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1] \tag{2}$$

Following [7, 9], the network we train is a fully connected NN with 3 intermediate layers with tanh activation functions and 64 nodes each. ADAM and L-BFGS optimisers are used in conjunction, with 15,000 steps of Adam with a learning rate of 0.001 used to initialise training for all approaches, followed by L-BFGS steps. In adaptive resampling, training is reinitialised after every new set of points is selected by using 1000 steps of ADAM (learning rate 0.001) followed by L-BFGS.

2.2. Adaptive Resampling

For the initial training steps with Adam, a quasi-random [10] grid of collocation points is generated over the domain to provide an initial sampling distribution, establishing a baseline solution from which to base the first resampling of points. In the residual-based resampling methods in the literature, the PDE residual losses are computed across the domain at this stage; in the approach by [7], this is done on a dense grid of candidate points X ($\approx$ 100, 000 points in a grid covering the entire space-time domain), for which a probability density function (see equation 3) is employed to assign a probability of being chosen to every point. This aims to make points in regions where the solution is less accurate or underrepresented likely to be chosen, according to some hyperparameters k and c which are manually selected. In addition to using the PDE residual as the Y (X), [9] utilised solution estimates and derivatives of the residual to identify underrepresented regions.

$$P(\mathrm{X}) = \frac{Y(\mathrm{X})^{k}}{\overline{Y(\mathrm{X})^{k}}} + c, \qquad \hat{P}(\mathrm{X}) = \frac{P(\mathrm{X})}{\|P(\mathrm{X})\|_1} \tag{3}$$

After each resampling step, the updated set of collocation points is used to continue training the PINN model for a fixed number of steps before repeating the resampling process, such that the next high-error regions are identified and prioritised. This process is repeated iteratively until the number of total resamples indicated has been carried out. To allow for direct comparison, the baseline nonadaptive sampling strategies were allowed to run for a number of optimisation steps equal to the maximum number of resamples investigated (∼116 000 optimisation steps).

The error metric used is L^2 error, given by (4), where $u(i)$ is the PINN prediction for a given point i, and u_{gt} represents the ground truth solution. This ground truth solution is obtained via a very highly resolved finite differences solver [11].

$$L^2 = \frac{\sqrt{\sum (u(i) - u_{gt}(i))^2}}{\sqrt{\sum u_{gt}(i)^2}} \tag{4}$$

2.3. Adjustments for two-dimensional Burgers' equations

The system of equations for Burgers' equation in 2D is as follows (with representative solutions shown in Figure 2):

$$u_t + uu_x + vu_y - \nu\,(u_{xx} + u_{yy}) = 0, \quad x, y \in [-1, 1], \quad t \in [0, 1], \tag{5}$$

$$v_t + uv_x + vv_y - \nu\,(v_{xx} + v_{yy}) = 0, \quad x, y \in [-1, 1], \quad t \in [0, 1]. \tag{6}$$

With $\nu = \frac{1}{100\pi}$, Dirichlet $u = v = 0$ boundary conditions at every wall and the initial conditions:

$$u(x, y, 0) = \sin(2\pi x)\sin(2\pi y); \quad v(x, y, 0) = \sin(\pi x)\sin(\pi y). \tag{7}$$

The architecture remains a fully connected network, albeit with two output nodes for the two different components of velocity u and v. However, we permit the option to increase the network depth, D, to ensure there is enough capacity to learn the higher-dimensional solutions. Due to the two outputs we now calculate two distinct physics losses for the individual PDEs. Both are calculated at every collocation point, and θ is optimised to minimise a global loss function that is a simple sum of both physics losses with no weighting applied. For the purpose of residual-based resampling, the PDE losses are also added at each point and inserted into (3) as before. For final accuracy measurements separate L^2 error is computed for u and v, and we display the average of the two as our results.

Increasing network depth generally increases training time, as more optimisation steps may be required to minimise a given loss function. For the experiments with varying numbers of fixed collocation points we observed that approximately 100,000 L-BFGS epochs saw the loss plateau for both PDEs. For the adaptive resampling experiments, each new set of points was trained on for 1,000 Adam steps, followed by 1,000 L-BFGS epochs. However, at higher network depths we found utilising longer steps of L-BFGS per resample was necessary to increase a network's capacity to learn from each set. To keep the total number of optimisation steps comparable, the number of resamples was decreased in proportion, down to 5 resamples with 20,000 L-BFGS steps each for the approach with results shown in figure 4.

3. Results

3.1. 1D Burgers' Equation

Figure 3 summarises results in 1D, showing errors achieved for a fixed number of points against number of resamples. Three adaptive strategies (solid lines) are compared against two fixed distribution

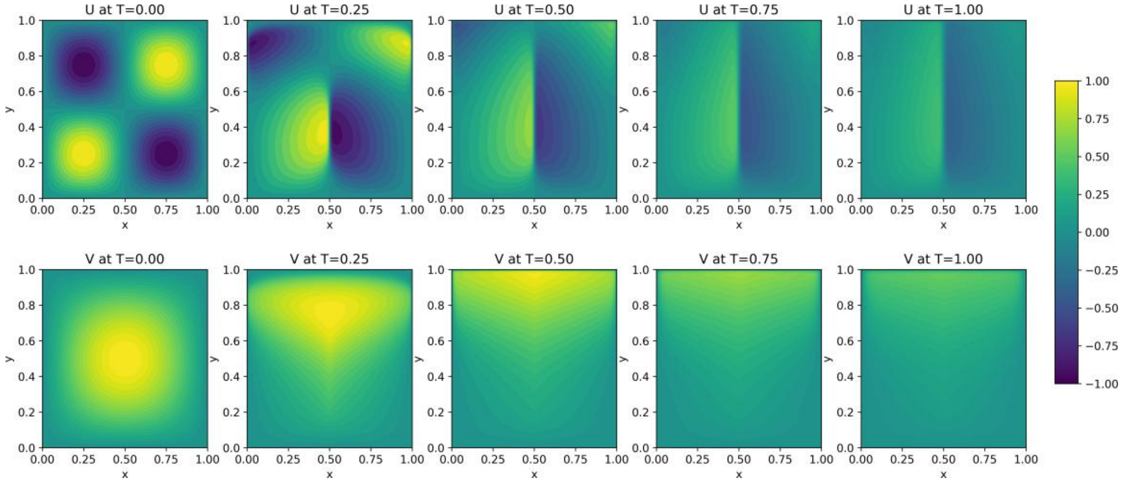

Figure 2: Ground truth for Burgers' equations in 2D, obtained via finite differences [11], showing horizontal speed u in the top row, vertical speed v in the bottom column, evolving over time.

strategies (dashed lines). The fixed distributions are a uniform random (suffix *R*), and a quasirandom Hammersley (suffix *H*). The different adaptive strategies use the method described in section 2.2, but using different markers as Y (X) in (3). These are the current PDE residual, based upon the approach proposed in [7]; the second derivative of the solution u with respect to the spatial and temporal dimensions (label U_{xt}), and the second derivative of the PDE residual (label PDE_{xt}). We can observe significant differences between adaptive resampling strategies and fixed point distributions in PINN training. Among the fixed distributions, purely random sampling (suffix *R*) yields the worst results, whereas quasi-random sampling (suffix *H*) performs slightly better. As expected, adaptive resampling — when initialized with an initial quasi-random distribution — outperforms the fixed quasi-random approach. It typically achieved accurate solutions (error < 0.1%) in as few as 20 resamples (equating to ~30% the optimisation steps).

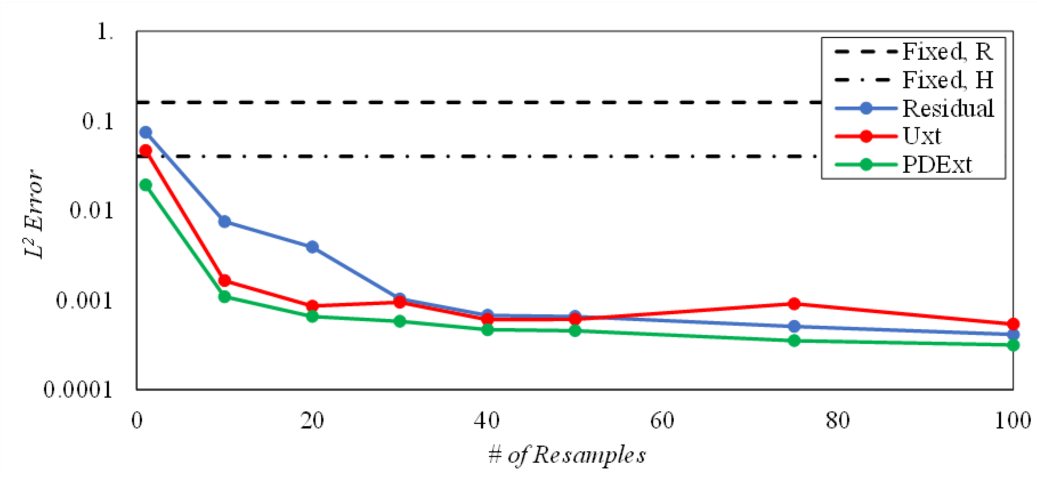

Figure 3: Error against number of resamples for the case of 1D Burgers' equation with 2,000 collocation points, trained for up to 115,000 epochs at 100 resamples.

In [9], it is shown that the choice of method became less impactful at very high number of points, with accuracy converging across cases. For lower values of N like the one shown above, performance depended heavily on method used, with adaptive methods able to reach good accuracy with significantly less training steps. However, for more complex initial conditions when the PINN struggled to learn effectively, neither the number nor the distribution of collocation points was sufficient to compensate for fundamental training deficiencies. The findings highlight the potential of adaptive resampling for maintaining low computational costs. However, it is unclear whether the strategies used for point redistribution scale efficiently with increasing domain dimensionality and, consequently, whether they contribute to sustaining the overall effectiveness of adaptive resampling methods.

3.2. 2D Burgers' Equations

Extending to the 2D problem as described in 2.3, we computed results of error against number of collocation points (N). Initially, we obtained results for a network depth of $D = 3$ using fixed point distributions, using the same training strategies as in 1D for different values of N. This was extended to deeper networks with $D = 5$ and $D = 8$. The residual resampling approach was carried out on these depths with both the original training strategy, and with the adapted strategy with longer optimisation regimes per resample. The results are presented in figure 4, where the black and red lines represent the fixed point methods using random and quasirandom distributions for all depths; the blue lines represent the previous residual resampling method; and the green lines represent the 'adjusted' strategy with longer training stages between resamples. Unlike in 1D, the quasi-random distribution exhibits similar performances to the uniform random distribution for the fixed cases throughout the range of N. Increasing the depth to $D = 5$ saw an increase in accuracy at $N = 100,000$, whereas using the highest depth of 8 actually decreased accuracy as $N > 20,000$. The residual-based adaptive resampling method with the previous training strategy shows the highest error independent of depth and number of collocation points. The adjusted method of residual-based resampling achieved lower errors at the lower values of N considered, and the lowest error overall for the case of $N = 50,000$, after which the performance deteriorated as the number of collocation points increased.

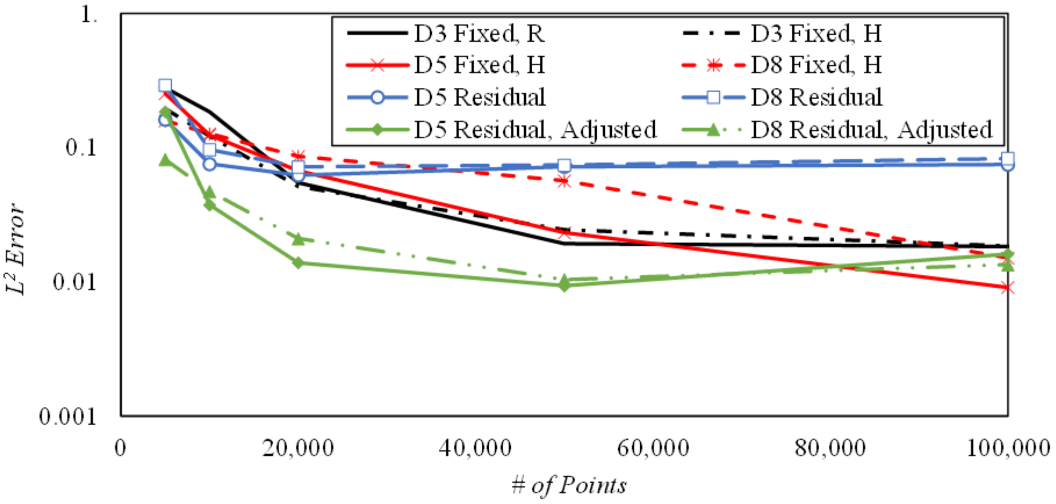

Figure 4: Error versus number of collocation points across network depths, comparing fixed distributions and residual-based adaptive resampling. Each model was trained for 115,000 epochs in total.

Beyond $N > 100,000$ (not pictured), error increases for all cases observed, suggesting that beyond the 1D case the increase in the number of points is not only computationally costly but also necessitates additional training steps (and/or a more refined strategy) for the additional points to have a positive impact. It can thus be observed that the Number of collocation points in 2D requires appropriate selection. In the 1D problem, the selection of the hyper-parameters in equation 3 also influenced the accuracy of the resampling method and represents another possibly significant dependency.

3.3. Discussion

A possible limitation of the current resampling method is the nature of the probability distribution function itself. While straightforward in design, it may unintentionally remove critical points at each resampling step, shifting them away from high-importance regions. This continuous redistribution of points introduces high variance in point selection, which could potentially hinder the Adam optimiser's ability to converge on stable network parameters. This may explain the observed lower performance compared to the fixed distribution method. Further investigation is required to determine whether alternative optimisation strategies could provide more robust convergence in this setting.

Beyond the specific resampling methodology, broader architectural constraints could also contribute to learning capacity limitations. The issue may not stem solely from network depth but rather from the restrictions imposed by using a fully connected network. Such architectures may lack the necessary representational flexibility to adapt efficiently to the changing set of collocation points. It is worth noting that the most accurate solutions hover around 1×10^{-3}, in contrast to 1×10^{-5} for the 1D problem. However, this could be partly attributed due to the increased complexity of the solution.

The use of derivatives to guide the adaptive resampling methods was the most successful method for the 1D Burgers' equation, but has not yet been explored in the 2D case. Consideration must be given that as dimensionality increases, taking a high order derivative of the solution with respect to all the dimensions quickly becomes prohibitively expensive. An alternative could be using functions of the first derivatives only, but the number of possible permutations also increases considerably. An important issue with scalability comes from the current resampling method, based on the approach in [7], which involves predicting derivatives on a very dense grid (defined previously as X). As the size of this grid also needs to be increased in proportion to the domain size, and exponentially with the number of dimensions, the approach does not scale to higher dimension PDEs.

4. Conclusion

This study has explored and contrasted the performance of different collocation point sampling strategies for the training of physics-informed neural networks on a 2D problem. It builds upon previous findings in 1D benchmarks that show resampling methods can produce more accurate solutions with fewer collocation points and optimisation steps than the standard fixed, uniformly random grid of collocation points. If using a fixed distribution of points, a quasirandom Hammersley distribution represents a robust alternative that performs as well as or better than uniform random sampling in both the 1D and 2D benchmarks observed. This paper has presented observations that the advantages of adaptive resampling could extend to 2D PDEs, but highlights that the current methods have limitations that can cause adaptive sampling to underperform relative to the baseline. A discussion on these issues is presented, identifying some of the underlying problems that would need to be addressed regarding the scalability of the method of resampling and the optimisation strategies employed.

References

[1] 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.

[2] Y. Wang, C.-Y. Lai, J. Go´mez-Serrano, and T. Buckmaster, "Asymptotic self-similar blow-up profile for three-dimensional axisymmetric euler equations using neural networks," *Phys. Rev. Lett.*, vol. 130, p. 244002, Jun 2023.

[3] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM Review*, vol. 63, pp. 208–228, 2021.

[4] G. Zhang, H. Yang, F. Zhu, Y. Chen, and x. zheng, "Dasa-pinns: Differentiable adversarial self-adaptive pointwise weighting scheme for physics-informed neural networks," *SSRN*, 2023.

[5] L. D. McClenny and U. M. Braga-Neto, "Self-adaptive physics-informed neural networks," *Journal of Computational Physics*, vol. 474, p. 111722, 2023.

[6] Y. Song, H. Wang, H. Yang, M. L. Taccari, and X. Chen, "Loss-attentional physics-informed neural networks," *Journal of Computational Physics*, vol. 501, p. 112781, 2024.

[7] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 403, 1 2023.

[8] M. A. Nabian, R. J. Gladstone, and H. Meidani, "Efficient training of physics-informed neural networks via importance sampling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, p. 962–977, July 2021.

[9] J. Florido, H. Wang, A. Khan, and P. K. Jimack, "Investigating guiding information for adaptive collocation point sampling in pinns," in *Computational Science – ICCS 2024: 24th International Conference, Malaga, Spain, July 2–4, 2024, Proceedings, Part III*, (Berlin, Heidelberg), p. 323–337, Springer-Verlag, 2024.

[10] T.-T. Wong, W.-S. Luk, and P.-A. Heng, "Sampling with hammersley and halton points," *J. Graph. Tools*, vol. 2, p. 9–24, Nov. 1997.

[11] M. S. Mathias, W. P. de Almeida, J. F. Coelho, L. P. de Freitas, F. M. Moreno, C. F. D. Netto, F. G. Cozman, A. H. Reali Costa, E. A. Tannuri, E. S. Gomi, and M. Dottori, "Augmenting a physics-informed neural network for the 2d burgers equation by addition of solution data points," in *Intelligent Systems* (J. C. Xavier-Junior and R. A. Rios, eds.), (Cham), pp. 388–401, Springer International Publishing, 2022.