UNIVERSITY OF LEEDS

This is a repository copy of *Investigating Guiding Information for Adaptive Collocation Point Sampling in PINNs*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/211954/

Version: Accepted Version

Proceedings Paper:
Florido, J., Wang, H., Khan, A. et al. (1 more author) (2024) 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. International Conference on Computational Science (ICCS) 2024: 24th International Conference, 02-04 Jul 2024, Málaga. Springer , Berlin, Heidelberg , pp. 323-337. ISBN 978-3-031-63758-2

https://doi.org/10.1007/978-3-031-63759-9_36

© Author | ACM 2024. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Computational Science – ICCS 2024: 24th International Conference, Malaga, Spain, July 2–4, 2024, Proceedings, Part III, https://doi.org/10.1007/978-3-031-63759-9_36.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Investigating Guiding Information for Adaptive Collocation Point Sampling in PINNs

Jose Florido[1](✉)[0009−0001−7291−5029], He Wang[2][0000−0002−2281−5679], Amirul Khan[1][0000−0002−7521−5458], and Peter K. Jimack[1][0000−0001−9463−7595]

[1] University of Leeds, Leeds LS2 9JT, UK
{mn17jilf,A.Khan,P.K.Jimack}@leeds.ac.uk
[2] University College London, London WC1E 6BT, UK
he_wang@ucl.ac.uk

Abstract. Physics-informed neural networks (PINNs) provide a means of obtaining approximate solutions of partial differential equations and systems through the minimisation of an objective function which includes the evaluation of a residual function at a set of collocation points within the domain. The quality of a PINNs solution depends upon numerous parameters, including the number and distribution of these collocation points. In this paper we consider a number of strategies for selecting these points and investigate their impact on the overall accuracy of the method. In particular, we suggest that no single approach is likely to be "optimal" but we show how a number of important metrics can have an impact in improving the quality of the results obtained when using a fixed number of residual evaluations. We illustrate these approaches through the use of two benchmark test problems: Burgers' equation and the Allen-Cahn equation.

Keywords: Partial differential equations · Deep learning · Physics-informed neural networks · Adaptivity.

1 Introduction

1.1 Context

This paper is concerned with the mechanism by which Physics-informed neural networks (PINNs) [1, 2], allow us to introduce our prior knowledge of physics into a machine learning (ML) algorithm. PINNs are an ML approach to the solution of systems of partial differential equations (PDEs), where there is limited or noisy ground truth data. In this context, the PDEs provide a mathematical model of the physics, which is captured in the method through the inclusion of loss terms that are evaluated by calculating PDE residuals at collocation points throughout the domain. This point-based, meshless approach offers greater flexibility compared to meshed methods in traditional numerical models - but also poses an interesting problem regarding the number and distribution of said collocation points.

The original implementation of PINNs uniformly distributes a fixed number of collocation points randomly throughout the domain. However, in certain problems some areas are intrinsically harder to learn than others - and this purely random approach can lead to slow or inefficient training. Different collocation points contain different amounts of information for learning, which indicates there is an underlying distribution of locations for collocation points that can maximise the learning - which is mostly likely non-uniform and problem dependent. Such a proposition suggests that adjusting the distribution of points, by biasing towards features of interest, may be beneficial in terms of efficiency. It may also be necessary for practical reasons to minimise the risk of the network getting trapped in local minima during training [3]. However, manually choosing this distribution of collocation points can be an arduous task and requires *a priori* knowledge of the solution, or at least of which areas of the domain will be "most important". The alternative is to increase the number of points used globally which, whilst potentially effective, adds significantly to the computational cost.

To find the optimal or a good distribution of the collocation points, there are two common approaches to explicitly or implicitly approximate this distribution. The first approach is to have fixed collocation points but weight them differently, i.e. *adaptive weighting*. Examples of this are [4], [5]'s Self-Adaptive PINNs, a concept extended by [6]'s DASA point weighing, and more recently [7]'s Loss-Attentional PINNs. The residual-based attention scheme in [8] works similarly to the above, weighting the influence of specific collocation points in the domain to ensure key collocation points aren't overlooked.

The other approach is to refine the locations of collocation points, i.e. *adaptive resampling*. [9] first presents this type of adaptive refinement based on residual information. Other sampling approaches include [10–12]. [13]'s formulation for resampling is a more general version of some of the previous approaches, and systematically compares adaptive collocation resampling methods to fixed approaches. Also noteworthy is [14]'s implementation of a cosine-annealing strategy for restarting training from a uniform distribution when optimisation stalls. Whilst the above focus on the collocation points where the PDEs are evaluated, [15] also considers optimising the selection of experimental points (where the data is available for supervised learning problems) at the same time.

From the sampling point of view, existing methods essentially seek the ideal distribution of collocation points, via either explicit parameterisation [9], or implicit approximation [10]. The information exploited up to date is mostly focused on the loss function [10, 12, 13], e.g. information such as local PDE residual at collocation points, or gradients of the loss term. Deviating from existing literature, we investigate into a different category of information that proves to be useful in parametrising the ideal collocation points. This information is the geometric information of the estimated solution, e.g. their spatial and temporal derivatives. The intuition behind is the solution geometry reveals intrinsic information about the PDE. For instance, stiffness regions are harder to learn [7]. By introducing the solution geometry, our aim is to shed a deeper light into

the complexity behind seeking an "ideal" distribution, and into the interplay between the effectiveness of different point sampling methods, the complexity of the problem and the computational cost.

1.2 PINNs

The process of training a PINN is similar to that for a regular NN: once a topology and a set of weights is defined we seek to find values of these weights that minimise a prescribed loss function. As discussed below, we will consider a fully unsupervised learning task for which the loss function is obtained from the sum of all of the residuals of the PDEs at all of the selected collocation points. The significance of the number and location of the collocation points is substantial therefore, since they directly impact on the loss function which is to be minimised. The crux of adaptive resampling is that a fixed number of points are moved based upon some criteria (that we investigate), with a view to this modified loss function being a better objective for the overall minimisation . Previous investigations of this approach have redistributed the collocation points based upon a probability density function (PDF) that was formed using the values of the residuals at the current points or the derivatives of the loss function with respect to the location of the collocation points. In this work, we consider alternative criteria based upon mixed derivatives of the estimated solution and of the residual.

The loss function $\mathcal{L}(\theta)$ of a PINN can be typically characterised as the sum of the individual losses due to data fitting ($\mathcal{L}_{data}$), enforcing the governing PDE ($\mathcal{L}_{PDE}$), and weakly enforcing boundary conditions ($\mathcal{L}_{BCs}$) as follows:

$$\mathcal{L}(\theta) = w_{data}\mathcal{L}(\theta)_{data} + w_{PDE}\mathcal{L}(\theta)_{PDE} + w_{BC}\mathcal{L}(\theta)_{BCs}$$

Here, the relative impact of each loss term can be controlled through the weight, w, whilst θ represents the trainable network parameters. For the one-dimensionsal time-dependent problems that we look at, we will consider a spatial dimension x, a temporal dimension t and solve for the dependent variable $u(x,t)$. The PDE loss term is obtained by evaluating the PDE residual (f, say) for every collocation point in the set of selected collocation points $\mathrm{x} \in \mathcal{T}$:

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

It is important to distribute these points throughout the domain however evidence suggests that it may also be beneficial to identify and place a greater proportion of points in particular regions of importance, so that the loss term especially includes these areas. This can ensure that optimising the cost function improves how well the PDE is satisfied around these locations, and therefore hopefully improves the solution in those areas.

For forward problems, PINNs can be trained without the need for data, and boundary conditions can be enforced in a 'hard' manner. This is the approach

that we take here in order to focus exclusively on the PDE residual contribution to the loss. We strongly impose the boundary conditions by applying an output transformation to the results of the neural network and we treat the PINN as an unsupervised problem with no input data. As a result, $\mathcal{L}_{PDE}$ becomes the only term to optimise in our loss function.

Throughout this paper, for consistency with other studies in the literature [13], a simple feedforward NN is used consisting of 3 intermediate layers of 64 nodes each, with a tanh activation function after each layer. The training consists of 15,000 initial steps using the ADAM optimiser (with learning rate of 0.001), followed by 1000 steps using L-BFGS [13] before beginning the resampling process for adaptive methods. For these methods, the points are then redistributed at this stage and then the training continues with 1000 steps of ADAM and 1000 steps of L-BFGS, repeating until the number of resamples specified are completed.

2 Adaptive resampling

2.1 Probability density functions

As mentioned in the introduction, for many problems PINNs will give more accurate results when the collocation points are distributed following an appropriate, problem-dependent distribution. The process of rearranging the collocation points can be automated by using information gathered during the training process. This approach has been shown to work well in the literature using the local residual as a guiding metric [9, 13]; but this paper also proposes alternatives based on the spatial and temporal derivatives of both the residuals and estimates of the solution. These proposed alternatives are assessed against both uniform, fixed distributions and existing adaptive redistribution methods, and shown to be beneficial in many cases.

The method that we use to resample the points is consistent with [13] for comparison, which chooses the next set of collocation points $\mathcal{T}$ from a fine grid of random points $\mathcal{X}$. All points in $\mathcal{X}$ are assigned a probability ($P(\mathcal{X})$), and the prescribed number of collocation points are chosen according to the normalised PDF $\hat{P}(\mathcal{X})$. These are obtained from equations 1 and 2:

$$P(\mathcal{X}) = \frac{Y(\mathcal{X})^k}{\overline{Y(\mathcal{X})^k}} + c \tag{1}$$

$$\hat{P}(\mathcal{X}) = \frac{P(\mathcal{X})}{\|P(\mathcal{X})\|_1} \tag{2}$$

Here $Y(\mathcal{X})$ is the vector containing the magnitude of the information source chosen (e.g. the current residual or its derivatives, or some derivatives of the current estimate of the solution), $\overline{Y(\mathcal{X})^k}$ is the mean value across all points in $\mathcal{X}$ and k, c are constant hyper-parameters that affect the resampling behaviour. A visualisation of the distribution of points obtained using the PDE residual as $Y(\mathcal{X})$ for solving the Burgers' Equation is shown in figure 1

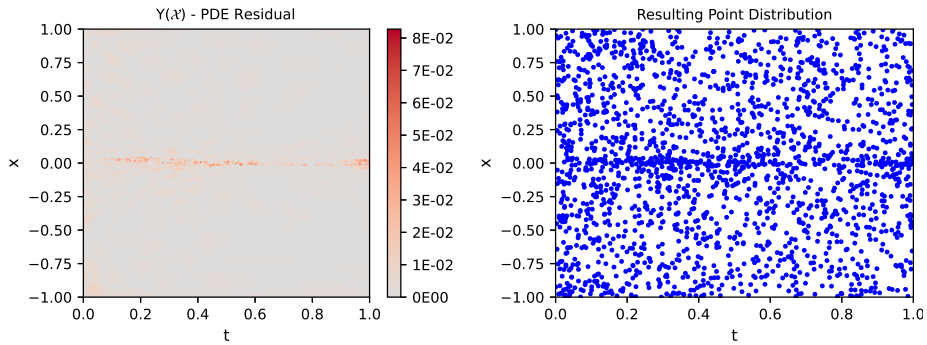

Fig. 1. $Y(\mathcal{X})$ = PDE Residual shown at every point in $\mathcal{X}$ (left); and the corresponding points selected for $\mathcal{T}$ (right).

As a baseline, we also compare the performance of the adaptive methods that we consider against two fixed distributions: a uniform random distribution and an alternative uniform, pseudo-random, distribution made with a Hammersley sampling algorithm [16]. For these fixed distributions, instead of resampling after a given number of steps, the training is continued until an equivalent amount of training steps are carried out.

We consider two cases for the adaptive method for which Y is based upon the current local residuals: one where the initial distribution of points before any resamples is randomly generated (PDE, R), and the other where the initial distribution follows Hammersley sampling (PDE, H). We also consider the use of the current solution estimates as an information source, defining Y based upon the mixed (spatial and temporal) second derivative of the current solution. We will refer to this as the local geometric curvature of u, and denote this case as "U_{xt}" in the subsequent text. We also consider a version of Y that is based upon the mixed second derivative of the local residual values, which will subsequently be denoted as "PDE_{xt}".

2.2 Problem Definition

The main benchmark that we initially use in order to assess the different sampling strategies that we consider is the 1D Burgers' Equation, given by:

$$uu_x + u_t = \nu\, u_{xx}, \quad x \in [-1, 1], \quad t \in [0, 1], \tag{3}$$

where the magnitude of ν determines the relative effect of diffusion. Dirichlet boundary conditions are assigned:

$$u(-1, t) = u(1, t) = 0,$$

as is the following initial condition:

$$u(x, 0) = -\sin(\pi x).$$

Note that, since we solve this problem on a space-time domain (see figure 2) the initial condition and the boundary conditions are treated identically by the PINN. To illustrate the main features of the analytical solution to this problem a colour map of u is plotted in figure 2. Note that the smooth initial solution (the left boundary of the domain shown) steepens to a very sharp front by $t = 0.25$ and this remains an important feature of the solution for all subsequent times.

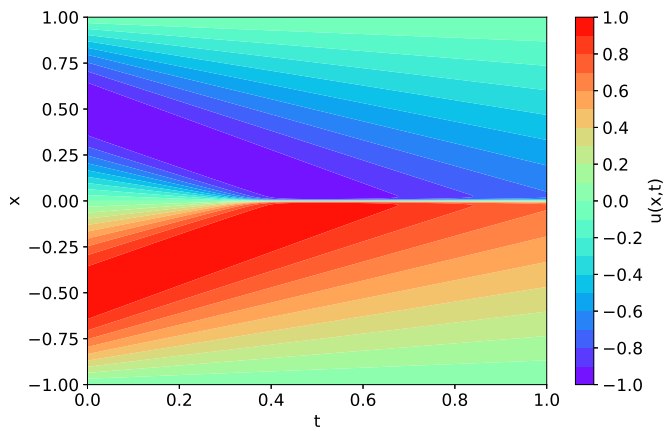

Fig. 2. Burgers' Equation contour plot of example solution with $\nu = \frac{\pi}{100}$

As discussed in the previous section the boundary and initial conditions may be imposed either weakly or strongly when using PINNs. Here we opt for the latter by applying an output transformation to the u that is output by the network:

$$output = -\sin(\pi * x) + (1 - x^2)ut$$

For this case; the PDE residual when used as an information source is therefore given by:

$$PDE = uu_x + u_t - \nu\, u_{xx} \tag{4}$$

Similarly,

$$U_{xt} = \frac{\partial^2}{\partial x \partial t} u(x,t) \tag{5}$$

and

$$PDE_{xt} = \frac{\partial^2}{\partial x \partial t}(uu_x + u_t - \nu\, u_{xx}) \tag{6}$$

To measure and compare the accuracy of the different methods considered an L^2 relative error metric is used at the end of training. This compares the prediction u to ground truth u_{gt} as follows:

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

Note that the sums in this last expression are over every point i in a very fine background grid which contains many more points than the number of collocation points used to evaluate u. Furthermore, due to the stochastic nature of ML training, a minimum of 20 repeats with different seeds were carried out for each case investigated throughout this paper; with the quoted L^2 errors always representing the average of these.

2.3 Results: default case

For this first test problem we have applied six different solution strategies using $N = 2000$ collocation points. In each case we follow the training regime of [13], described in section 1.2 (15,000 initial steps of ADAM followed by 1000 steps of L-BFGS and then resampling every 2000 steps (1000 ADAM/1000 L-BFGS). Figure 3 plots the error against the number of resamples taken for each of the approaches considered, including the two constant point sets based upon a uniform random distribution and a distribution based upon the Hammersley sampling algorithm.

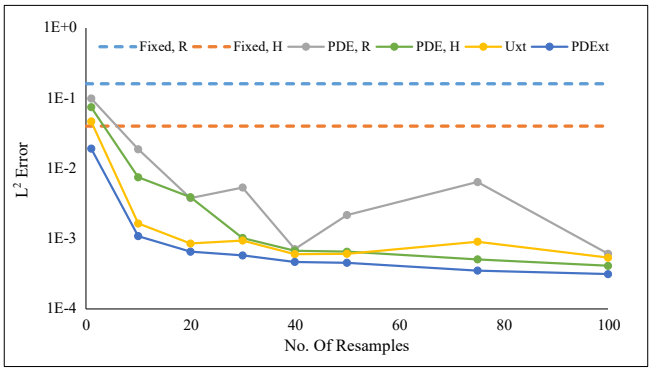

Fig. 3. Error against number of resamples solving Burgers' Equation with described default parameters $v = \frac{\pi}{100}$; $N = 2000$

The initial distribution is seen to have an effect in both the fixed and residual-based methods, with the use of the Hammersley sampling algorithm resulting in consistently lower error. The local residual method with random initialisation (PDE, R) exhibited the least robustness and greatest variability, which can be seen from the fact increasing the number of resamples did not consistently lower the average error between runs. Generally, more resamples does result in improving accuracy, with error descending below 0.1% for above 30 resamples for the most of the adaptive resampling methods.

For the same case we also look at the error achieved using different numbers of collocation points. This was obtained for 100 rounds of resampling and is plotted in figure 4. This figure suggests that the threshold for the number of points

required to obtain results of a given accuracy varies from method to method. Even for the most accurate method (PDE_{xt}), the error seems to plateau however between 0.01% and 0.1%, which the other adaptive methods eventually reach at the default 2000 points. Nevertheless, this clearly suggests that the PDE_{xt} approach could allow similar accuracy to other techniques but at a significantly lower cost.

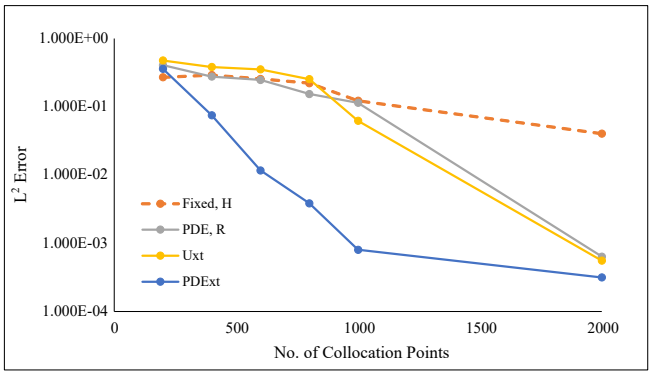

Fig. 4. Error against number of collocation points for different resampling methods solving Burgers' Equation with described default parameters $v = \frac{\pi}{100}$; 100 resamples

In conclusion, based on this single test case, we observe that the achieved error of a PINN is heavily influenced by the chosen collocation point distribution, echoing what has been seen previously (see Introduction). Furthermore, the adaptive resampling methods easily outperform the fixed distributions, with both of the derivative-based redistribution methods performing quite well (especially when considering the mixed second derivative of the residual). In the next section we explore the extent to which this behaviour generalises by changing multiple aspects of the problem under consideration, including the initial conditions and the magnitude of the diffusion term, and by considering a second benchmark PDE: the Allen-Cahn Equation.

3 Results: other cases

3.1 Alternate Initial Conditions

In this example we contrast the same methods as in the previous section to once again solve Burgers' Equation, but this time we vary the initial conditions, and therefore the entire evolution of the solution of the PDE. For each of the four cases considered we compute a high resolution numerical solution to represent the ground truth and use this to assess the error against an increasing number of adaptive resamples of the collocation points. The corresponding results are plotted in figure 5.

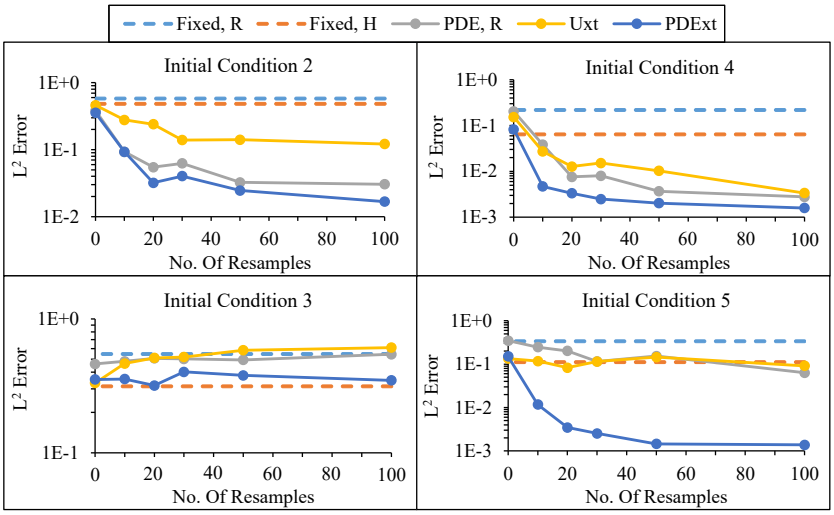

Fig. 5. Error versus number of resamples. Initial conditions 2 and 3 are combinations of randomly weighted sin curves of different frequency; with 2 being made up of three low frequency curves and 3 being a higher frequency combination of four curves. Initial conditions 4 and 5 are $sin(2\pi)$ and $1.5sin(\pi)$ respectively.

Inspection of figure 5 suggests that for three of the four initial conditions (2, 4 and 5), PDE_{xt} clearly outperforms the other methods (as it did for initial condition 1 in the previous section). Not only does it deliver a smaller error after the last resample (the maximum amount of training investigated) but the error curve lies beneath the others at each stage in the adaptive process. This gap is most noticeable in the 5th case, which corresponds to an initial condition with a larger amplitude, thus leading to a faster shock formation and a sharper shock. In this case it far outperforms the next best method. Furthermore, in each case other than initial condition 3, the adaptive collocation point algorithms all performed significantly better than using fixed collocation points. Unsurprisingly, of the two fixed point approaches Hammersley always outperforms the uniform random distribution.

The obvious outlier in figure 5 is initial condition 3. Here all of the methods to quite poorly in terms of reducing the error and least worst result is obtained using fixed collocation points based upon a Hammersley sampling. Further investigation is required to better understand this result, which shows that the PINN performance (or, at least, the ease with which the PINN can be trained) is highly dependent upon the problem being solved.

3.2 Adjusting PDE parameters

For the following example, we investigate adjusting the value of the ν parameter in the PDE; which affects the magnitude of the diffusion term $\nu\ u_{xx}$. Lower values

increase the shock sharpness, increasing the complexity of the problem without fundamentally changing the shape of the solution. Increasing ν sees the opposite effect. We use values of ν of 0.01 and 0.001, approximately multiplying and dividing by 3 from the default $\nu = \frac{\pi}{100}$ used in the previous section. The ground truth is again computed using high-resolution numerical simulations and the error when using different information sources to control the adaptive resampling is again compared against the fixed uniform methods: see figures 6 and 7.

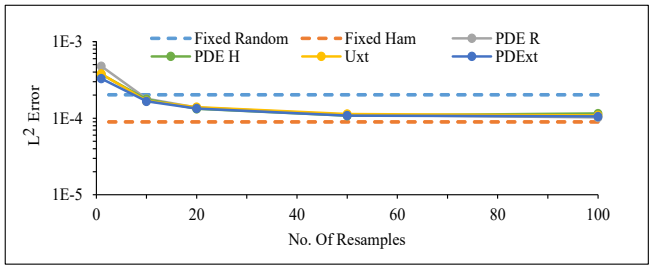

Fig. 6. Error against number of resamples, for Burgers' Equation with $\nu = 0.01$

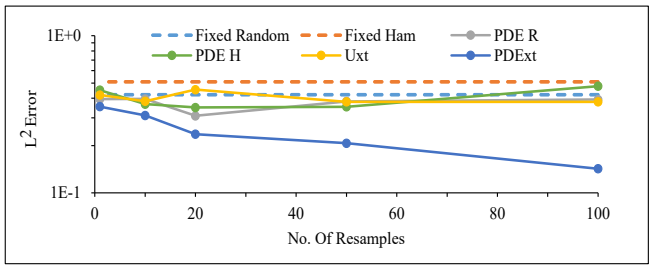

Fig. 7. Error against number of resamples, for Burgers' Equation with $\nu = 0.001$

For the simpler case, $\nu = 0.01$ (figure 6), all adaptive methods behave similarly, delivering a small error after relatively few resamples. Comparing the fixed methods, the Hammerlsey distribution is again consistently better than a random fixed distribution. Furthermore, in this specific example, it again outperforms the adaptive sampling methods. Unlike case 3 of figure 5 however, the accuracy of all of these PINNs solutions is high. One possible explanation for the Hammersley approach being the best in this case is that, unlike the resampling methods, it uses the L-BFGS optimisation algorithm for the majority of training steps (as opposed to an equal split between ADAM and L-BFGS for the adaptive methods). Overall, It is clear that by making the solution of the problem smoother (through increasing ν) all of the methods do well and therefore use of

the simplest approach, based upon a fixed distribution of collocation points, can be appropriate.

For the more challenging case of $\nu = 0.001$ (figure 7), all methods struggle to converge to an accurate solution using $N = 2000$ collocation points. The least worst approach, with an L^2 error of 14%, is again the adaptive method based upon PDE_{xt}. Based upon this set of results we can make the opposite proposition: that for complex problems, the choice of adaptive sampling method may not be the limiting factor to accuracy, and an increase to either the number of collocation points or the network complexity may be needed to increase accuracy.

We test this proposition by undertaking a study to look at the impact that changing the number of collocation points has on the solutions obtained in the $\nu = 0.01$ and $\nu = 0.001$ cases. In table 1 we consider the simpler $\nu = 0.01$ case, observing the change in performance when the number of collocation points is halved to 1000. Conversely, in table 2, the more complex case with the sharper shock ($\nu = 0.001$), we double the number of collocation points to $N = 4000$ and observe whether there are any significant increases in accuracy.

Table 1. Effect of changing N; L^2 Error at 100 resamples for $\nu = 0.01$

N	Fixed, R	Fixed, H	PDE, R	PDE, H	U_{xt}	PDE_{xt}
2000	2.019E-04	8.95E-05	1.024E-04	1.152E-04	1.089E-04	1.055E-04
1000	3.089E-03	3.925E-04	1.503E-04	1.867E-04	1.458E-04	1.258E-04
Error Change	1429.9%	338.4%	46.8%	62.1%	33.9%	19.3%

In table 1, decreasing the number of collocation points from 2000 to 1000 increases the error as expected. However, there is a big discrepancy in how much the different methods are affected. A positive % signifying increase in error, the fixed methods are substantially less effective with fewer points. The opposite is true for adaptive methods, especially for PDE_{xt} which only sees an increase in error of 19.3% compared to the 3-fold increase of the fixed Hammersley distribution of points.

Table 2. Effect of changing N; L^2 Error at 100 resamples for $\nu = 0.001$

N	Fixed, R	Fixed, H	PDE, R	PDE, H	U_{xt}	PDE_{xt}
2000	4.209E-01	5.093E-01	3.925E-01	4.780E-01	3.797E-01	1.425E-01
4000	4.049E-01	3.836E-01	1.858E-01	1.996E-01	2.591E-01	3.660E-02
Error Change	-3.9%	-32.8%	-111.2%	-139.5%	-46.6%	-289.3%

In table 2, we are this time looking at the decrease in error as a result of doubling the number of collocation points in the case of $\nu = 0.001$. As anticipated, the error does decrease in each column (signified by a negative increase)

however this improvement is relatively small in most cases, meaning that the overall errors are still not competitive, even with this increase in resources. The biggest decreases are seen for the PDE, H and especially PDE_{xt}, which is the only method to achieve an error under 10%.

3.3 An alternative PDE: Allen-Cahn

Having presented several variations of the Burgers' Equation problem, our next step is to consider a different PDE entirely and observe the results. The Allen-Cahn equation is chosen for this due to its complex transient behaviours which are known to be challenging to capture computationally. The problem is defined as follows:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + 5(u - u^3), \quad x \in [-1, 1], \quad t \in [0, 1] \tag{7}$$

We take D = 0.001 and use the initial condition

$$u(x, 0) = x^2 \cos(\pi x) \ ,$$

and boundary conditions

$$u(-1, t) = u(1, t) = -1 \ .$$

For this problem we again compute a high-resolution numerical solution for the ground truth and consider the L^2 error as the number of resamples is increased. Based on the observations of the previous subsections, we undertake these tests for three distinct numbers of collocation points: $N = 500, 1000, 2000$. The same resampling methods as in section 2 are used, using a minimum of 20 repeats per case. The results for $N = 500$ are shown below in figure 3.3. For this case, the adaptive methods are again consistently superior to using fixed collocation points. Similar to the Burger's equation solutions with a large $\nu = 0.01$, there are no clear advantages between PDE_{xt}, and the approaches based upon the local residual (though using U_{xt} is not so good).

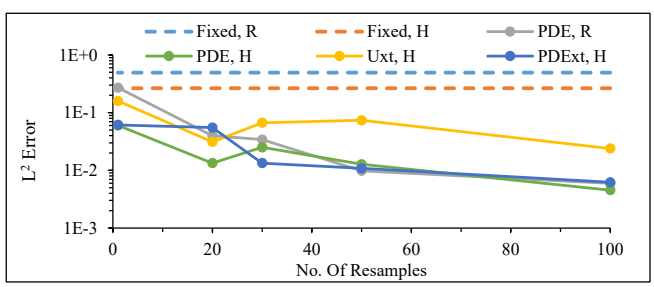

Fig. 8. Error against number of resamples solving the Allen-Cahn Equation with $N = 500$

For $N \geq 1000$, a similar pattern is observed to that shown in figure 6, where there is little difference between resampling and fixed methods. In this case, the increased number of points again makes the problem simple enough to solve with fixed collocation strategies, so that the benefits previously observed from using adaptive resampling methods disappear. In fact a fixed pseudo-random Hammersley distribution performs better than the other curves: perhaps again benefiting from more L-BFGS optimiser steps.

For the Allen-Cahn Equation case, the behaviour and importance of sampling in these methods varies significantly depending on the number of points used. The less sharp shocks in this solution could also be influencing the effectiveness of using derivatives of the solution as redistribution criteria. Nevertheless, the adaptive methods clearly outperform the fixed distribution of collocation points for the case $N = 500$, again suggesting that results of a prescribed accuracy can be obtained with fewer collocation points when adaptive sampling is permitted.

4 Discussion and Conclusion

In this paper, to better explore the problem of collocation point sampling in PINNs we have examined the two benchmark problems of Burgers' Equation and the Allen-Cahn Equation. We have used novel information sources based on derivatives of the local residual and the solution estimate to guide the resampling process, which has been kept consistent with other approaches in literature, and compared the results to other resampling methods and fixed random and pseudo-random distributions.

Overall, we have observed that using adaptive point distribution typically allows more accurate PINNs solutions to be obtained than using fixed point distributions for small numbers of collocation points. Using fewer collocation points is potentially important since the training cost of a PINN grows in proportion to this number (all other factors being equal). Furthermore, we have observed that defining the probability distribution for the collocation point locations based upon the mixed second derivative of the residual with respect to the independent variables typically outperforms, often significantly, using the residual alone, as in [13].

For Burgers' equation with a smooth initial condition that leads to the formation of a single "shock", and a diffusion parameter such that this features is sufficiently sharp, we obtained the smallest errors using this approach (Fig. 3). Equally importantly, we observed that we could reach such errors with significantly fewer collocation points than for any of the other techniques that we compared against (Fig. 4). When considering other initial conditions and other diffusion parameters the relative performance of the adaptive approaches was seen to depend upon the complexity of the specific solution in each case. Typically, there appears to be a value of N above which the use of a uniform (Hammersley) distribution of points is preferred - however this value can be relatively large for the more complex problems (and is relatively small when the solution is smoothly varying everywhere). When considering numbers of points below

this critical value of N we invariably find that using the mixed derivative of the residual is the best strategy for defining the PDF for drawing the collocation point locations.

Consideration of a different PDE, the Allen-Cahn equation in this case, demonstrates similar behaviour. Specifically, for a given problem there is a value of N below which the adaptive sampling approach is preferable (and above which a uniform Hammersley distribution suffices). In the particular examples considered here, defining the PDF based upon the mixed derivative of the residual is still reliable and robust, however its improvement relative to using the residual alone is less clear. Future work will extend the exploration of these resampling methods to other PDEs beyond Allen-Cahn, helping assess the generalizability of our findings. As our approach does not explicitly constrain the type of domain we can study, we should also be able to consider domains with higher spatial dimensions and more complex shapes.

In terms of computational cost, the PINNs network we have considered is not competitive with standard discretization methods for solving individual forward problems. This is primarily due to the high computational cost of training with full gradient descent methods like L_BFGS. As for the cost of the resampling strategies we have discussed, these only contribute to a small overhead (estimated at 3% in [13]) to the overall computational cost of PINNs training.

Considering other common problems in ML, the biggest challenge is determining what constitutes an appropriate training regime. We did not observe overfitting to occur, even with fixed sampling methods. However, the choice of optimisation algorithm and training regime was still impactful. The use of faster stochastic gradient descent methods like ADAM helped initialise training but had a limited ceiling to accuracy, hence the need for the more expensive L_BFGS. Whilst we were not able to fully explore the impact of different training regimes within the scope of this paper, efficiency could feasibly be improved by further optimising this via methods such as cosine annealing [14]; or by optimising the architecture employed for solving problems of varying complexity.

In summary, we have shown that a number of different strategies are viable for selecting the location of PINNs collocation points with resampling methods. In particular, we have shown that resampling can be guided not only by the local residuals, but also by the spatial and temporal derivatives of the residuals and even of the solution estimates themselves. Furthermore, we have observed that a number of different factors impact the accuracy of the PINN method, making it difficult to analyse any single aspect (such as collocation point selection) in isolation. Whilst we believe that our contribution begins to shed some light on the interplay between the location of the collocation points and the ability of a PINN to learn the PDE solution it is clear that further research is desirable in order to be able to propose robust implementations that support reliable training or provide accuracy guarantees.

Acknowledgments. The first author gratefully acknowledges financial support via the EPSRC Centre for Doctoral Training in Fluid Dynamics at Leeds (EP/S022732/1).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Lagaris, I., Likas. A., Fotiadis, D.: Artificial neural Networks for Solving Ordinary and Partial Differential Equations. IEEE Transactions on Neural Networks **9**(5), 987-1000 (1998)
2. Raissi, M., Perdikaris, P., Karniadakis, G.: 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)
3. Wang, Y., Lai, C., Gomez-Serrano, J., Buckmaster, T.:Asymptotic Self-Similar Blow-Up Profile for Three-Dimensional Axisymmetric Euler Equations Using Neural Networks. Physical Review Letters **130**, 244002 (2023)
4. Li, W., Zhang C., Wang, C., et al.: Revisiting PINNs: a Generative Adversarial Physics-Informed Neural Networks and Point-Weighting Method, arXiv Preprint, arXiv:2205.08754 (2022)
5. McClenny, L. Braga-Neto, U.: Self-Adaptive Physics-Informed Neural Networks using a Soft Attention Mechanism. Journal of Computational Physics **474**, 111722 (2022)
6. Zhang, G., Yang, H., Zhu, F., et al.: Dasa-pinns: Differentiable adversarial self-adaptive pointwise weighting scheme for physicsinformed neural networks, SSRN (2023)
7. Song, Y., Wang, H., Yang, H.: Loss-Attentional Physics-Informed Neural Networks. Journal of Computational Physics **501**, 112781 (2024)
8. Anagnostopoulos, S., Toscano, J., Karniadakis, G., et al.: Residual-based attention and connection to information bottleneck theory in PINNs, arXiv Preprint, arXiv:2307.00379 (2023)
9. Lu, L., Meng, X., Karniadakis, G., et al.: DeepXDE: A Deep Learning Library for Solving Differential Equations. SIAM Review **63**, 208-228 (2021)
10. Nabian, M., Gladstone, R., Meidani, H.: Efficient training of physics-informed neural networks via importance sampling. Computer-Aided Civil and Infrastructure Engineering **36**(8), 962-977 (2021)
11. Hanna, J., Aguado, J., Borzacchiello, D., et al.: Residual-based adaptivity for two-phase flow simulation in porous media using Physics-informed Neural Networks. Computer Methods in Applied Mechanics and Engineering **396**, 115100 (2022)
12. Gao, Z., Yan, L., Zhou, T.: Failure-Informed Adaptive Sampling for PINNs. SIAM Journal on Scientific Computing **45**, A1971-A1994 (2023)
13. Wu, C., Zhu, M., Lu, L., et al.: A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering **403**, 115671 (2023)
14. Subramanian, S., Kirby, R., Gholami, A., et al/: Adaptive Self-Supervision Algorithms for Physics-Informed Neural Networks. ECAI 2023 **372**, (2023)
15. Lau, G., Hemachandra, A., Low, B., et al.: PINNACLE: PINN Adaptive ColLocation and Experimental points selection. In: The Twelfth International Conference on Learning Representations (2024)
16. Hammersley, J. M., and D. C. Handscomb. Monte Carlo methods, methuen & co. Ltd., London 40 (1964): 32.