# Investigating Guiding Information for Adaptive Collocation Point Sampling in PINNs

J Florido[1 [0000−1111−2222−3333]], H Wang[2 [1111−2222−3333−4444]], A Khan[1 [2222−−3333−4444−5555]], and P Jimack[1 [2222−−3333−4444−5555]]

[1] University of Leeds, Leeds LS2 9JT, UK
[2] University College London, London WC1E 6BT, UK

**Abstract.** This paper compares the use of different information sources in the guiding of collocation point resampling in PINNs. Previously seen resampling methods using PDE residuals are compared to the use of PDE residuals and solution estimates with respect to solution domain... These are tested on the Burgers' equation and modifications of it, as well as on the Allen-Cahn equation. We find ...

**Keywords:** First keyword · Second keyword · Another keyword.

## 1 Introduction

### 1.1 Context

This paper is concerned with the efficient implementation of Physics Informed Neural Networks (PINNs), first introduced by [2]. PINNs are a ML approach to the solution of systems of partial differential equations (PDEs), where there is limited or noisy ground truth data. This is thanks to the incorporation of physics by introducing PDE loss terms, evaluated through calculating PDE residuals at collocation points throughout the domain.

### 1.2 Motivation

The original implementation involves a fixed random distribution of collocation points. For certain problems, this can lead to slow or inefficient training. There, manually biasing the points towards feature of interest is necessary for practical reasons [5] to minimise the risk of the network getting trapped in local minima during training.

### 1.3 Prior Work

Different approaches and improvements have been suggested to handle this issue with collocation points. [6] indirectly tackles this by varying the weight of individual collocation points without changing their placement. However, a lot of approaches in the literature adjust the way collocation points are distributed.

Work by [7] notes the advantages of refinement, adding collocation points automatically based on where the PDE residuals are higher. [1] thoroughly evaluated different fixed sampling methods, where an initial pattern of points is selected and kept for the entire training process; as well as adaptive resampling and refining methods that move or add points based on residual information.

### 1.4 Signposting

[3] uses a cosine-annealing strategy, which changes the sampling method from uniform to adaptive throughout training. To guide adaptive sampling, points are selected according to the residual of the PDE and its gradient with respect to the domain. [8] looks at one different error indicator: a 'failure probability' also obtained from residual information. [2004, Wang - a high-order global spatially adaptive collocation method for 1d parabolic pdes]. [2022, Yu - Gradient enhanced physics informed neural networks for forward and inverse PDE problems] - gradients used already in PINNs - for optimisation as opposed to in collocation point sampling.

## 2 Problem Setup

This section will broadly cover the implementation of adaptive sampling methods, illustrating the effect of novel information sources vs the baseline; and justifying choices made in the process. First, the baseline fixed sampling and for adaptive sampling (based on [1]) is shown, as well as our alternative information sources, together with a justification for using a pseudo-random initial distribution even in adaptive sampling methods. A short note is then made on the hyper-parameters guiding how the information is used to move the points, with reference to alternative options to remove the dependence on these parameters. Lastly, in 2.3 all the metrics of the PINN itself and the training regime used are detailed.

### 2.1 Baseline, Methods & Initial Distribution

The original collocation point implementation by [2] used fixed random sampling, where the collocation points are not re-sampled. We will also be comparing to the adaptive re-sampling method introduced by [1], which they prove to be a significant improvement on a few different fixed sampling methods across a variety of benchmarks. Their adaptive re-sampling is based on using residual information obtained from the current training iteration - which as their BC and IC are satisfied by default, is exclusively the PDE loss.

We investigate the use of two other sources of information. First we investigate whether solution estimates produced by the network can be used to accurately resample the points towards areas of interest. We also consider the effect of taking derivatives of our information sources (both the PDE loss, and the velocity u) with respect to the spatial grid, which we refer to as $u_{xt}$ and $PDE_{xt}$.

Figure 1 shows how the resampling process, uses these values (middle) to re-sample the points from the initial distribution (left) to the next distribution (right).

The first question regarding re-sampling methods is how to resample the points initially, before training begins. From [1]'s work looking at fixed sampling methods, the final accuracy can be impacted by that single selection of points. Some pseudorandom methods are generally more accurate, such as the Hammersley sequence. As this latter sees the best results for most of the benchmarks looked at, it was chosen as the initial distribution for the methods investigated. We verified the effect of the initial distribution in adaptive sampling methods by comparing different cases 1, evaluating the error via the $L^2$ relative error (see section 3.1 for details).

Whilst the effect on mean error is not very significant, looking at individual runs it seems that it prevented outliers where network got stuck and produced errors higher order of magnitude higher than the mean. For this reason, it was decided to use the Hammersley initial distribution method for runs (denoted by R), except for the case of comparing to the baseline of [1]'s RAD, which we simply label PDE, R.

### 2.2 Hyper-Parameters

A second important set of parameters relate to how the information was transformed into the probability distribution function that selected new points. This is done via equations 1 and 2, which turn information $Y$ into a normalised probabilty distribution function $\hat{P}(x)$ for a large number of randomly selected points $x$. The prescribed $N$ collocation points are then selected according to $\hat{P}(x)$.

This introduces two hyper-parameters for us to tune; $k$ and $c$. $c$ adds a base probability to all points, effectively driving towards a uniform random distribution; whereas k adjusts the sensitivity to changes in information.

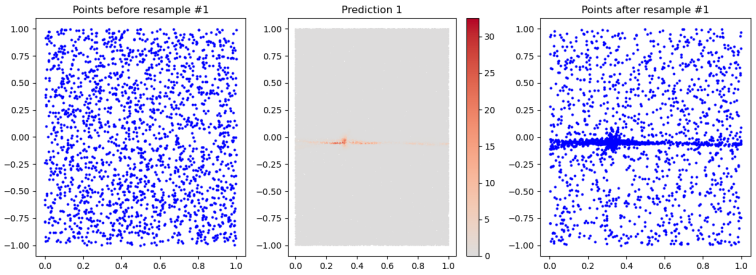
$$P(x) = \frac{|Y(x)|^k}{\overline{Y(x)^k}} + c \tag{1}$$

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

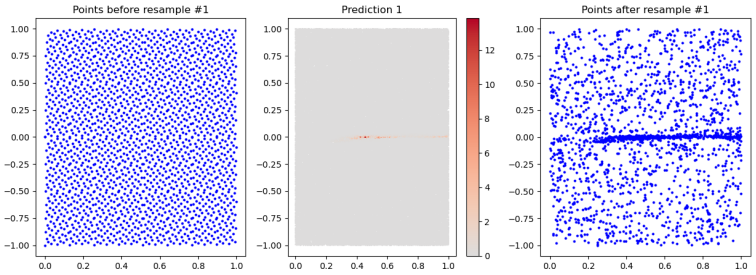
For the baseline method PDE, R from [1], they find values $k = 1, c = 1$ to be a good default. We quickly verify this by looking at the effect of varying k, comparing values for PDE, H as the method. This can be viewed in table 1.

Using second derivatives of the solution estimate however we find that reducing $k$ yields better results. Moving on, the best hyper-parameters as read from table 1 are utilised (k=1 for PDE, k=0.5 for other methods involving curvature).

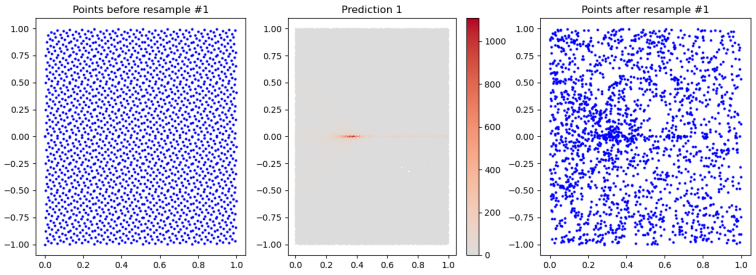
As the effect of the hyper-parameter $c$ is essentially to add some noise, increasing it opposes the adaptive resampling. However, having some level of random distribution can ensure that collocation points are distributed throughout the domain even in cases with extreme residuals or solution gradients. As one could



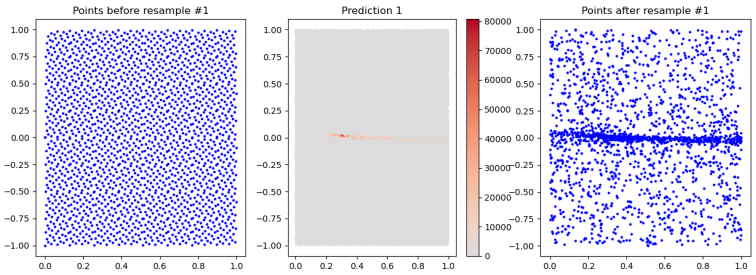

(a) PDE




(b) PDE, H

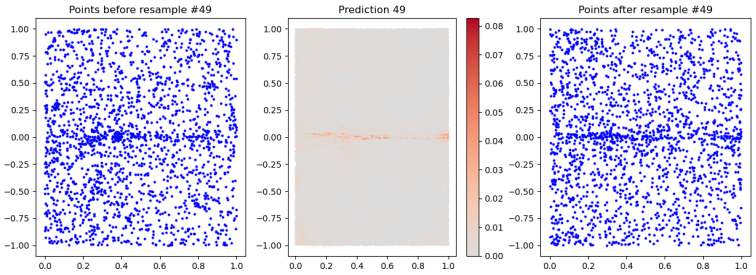



(c) U Curvature

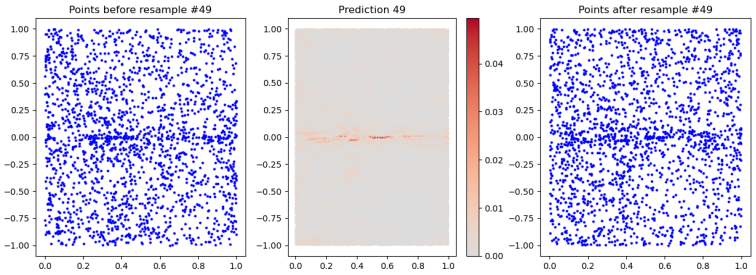



(d) PDE Curvature

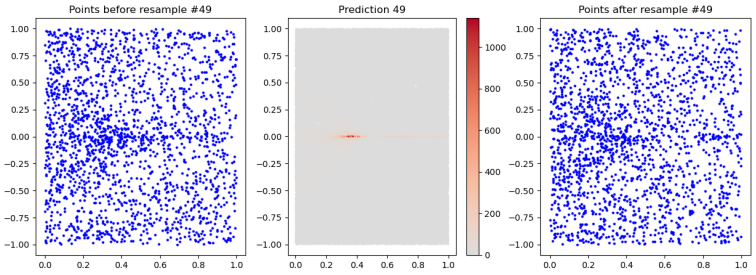
**Fig. 1.** First resample using different methods



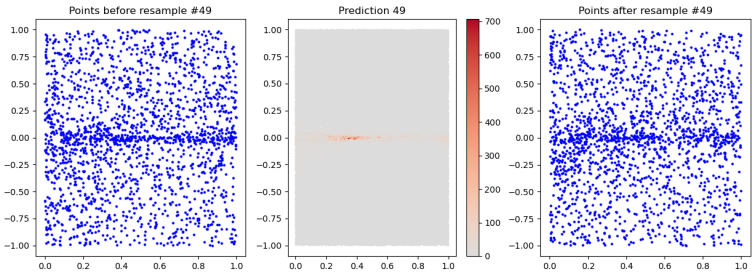

(a) PDE




(b) PDE, H




(c) U Curvature




(d) PDE Curvature

**Fig. 2.** Middle resample using different methods

**Table 1.** Associated Errors of different initial distributions and hyper-parameter choices

| Method | Hyperparameters | Error | Standard Deviation |
|---|---|---|---|
| Fixed, Random | N/A | 1.61E-01 | 1.11E-01 |
| Fixed, Uniform Grid | N/A | 1.33E-01 | 4.90E-02 |
| Fixed, Hammersley | N/A | 4.02E-02 | 3.48E-02 |
| PDE, Random Initialisation | $k = 1, \quad c = 1$ | 4.92E-04 | 2.95E-04 |
| PDE, Hammersley Initialisation | $k = 1, \quad c = 1$ | 5.15E-04 | 2.38E-04 |
| | $k = 0.5, c = 1$ | 5.41E-04 | 5.17E-04 |
| | $k = 2, \quad c = 1$ | 8.37E-04 | 3.37E-04 |
| Uxt, Random Initialisation | $k = 1, \quad c = 1$ | 1.01E-03 | 4.28E-04 |
| | $k = 0.5, c = 1$ | 4.80E-04 | 2.16E-04 |
| | $k = 2, \quad c = 1$ | | |
| Uxt, Hammersley Initialisation | $k = 1, \quad c = 1$ | 1.10E-03 | 6.11E-04 |
| | $k = 0.5, c = 1$ | 5.39E-04 | 3.63E-04 |
| | $k = 2, \quad c = 1$ | 1.56E-03 | 8.19E-04 |

assume the necessity for this might decrease as training advances and the placement of points is more stable in the case of solution estimate as an information source. To check this, we also investigate the effect of linearly decreasing c from 1 to 0 as training progresses, which has the additional benefit of removing an additional hyper-parameter to tune. The accuracy of this is compared to using the default $c = 1$ in table 2

**Table 2.** Effect of Linearly Reducing Noise Factor

| | Method | Error | Standard Deviation |
|---|---|---|---|
| Fixed $c = 1$ | PDE, Random | $6.12 \times 10^{-4}$ | $7.16 \times 10^{-4}$ |
| | PDE, Hammersley | $4.13 \times 10^{-4}$ | $2.66 \times 10^{-4}$ |
| | Uxt, Hammersley | $5.39 \times 10^{-4}$ | $3.63 \times 10^{-4}$ |
| Linearly Reducing c | PDE, Hammersley | $4.75 \times 10^{-4}$ | $2.12 \times 10^{-4}$ |
| | $u_{xt}$, Hammersley | $4.46 \times 10^{-4}$ | $2.05 \times 10^{-4}$ |

This could serve a similar role to cosine annealing by [3], where on detecting plateau in loss their PINN reverts to uniform sampling with the aim of avoiding local training minima.

### 2.3 Metrics

NN architecture, metrics, training/resampling regimes. A simple feedforward NN was used, for consistency with other studies in literature, consisting of 3 intermediate layers of 64 nodes each.

The boundary conditions were enforced in a hard way by applying an output transformation to the results of the neural network; example given in 3.1.

The training consisted of 15,000 initial steps using ADAM optimiser (with learning rate of 0.001), followed by 1000 steps using L-BFGS before beginning the resampling process for adaptive methods. For these, the points were redistributed at this point and then the training continued with 1000 steps of ADAM and 1000 steps of L-BFGS, repeating until the number of resamples specified are done.

## 3 Burgers' Equation

In this section, more details are given on Burgers' equation problem and on the error metrics considered (3.1). The results of the investigation for the default initial condition are then displayed in **??**, showing the performance of different methods with decreasing resources (reducing number of steps/resamples and number of training points). In **??** this is extended to more challenging initial conditions; and in **??** to altering the PDE itself via adjusting $\nu$.

### 3.1 Equation, ICs, Error metrics

Burgers equation is given by:

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

where the magnitude of $\nu$ determines the relative effect of diffusion. with Dirichlet boundary condition:

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

and the default initial condition:

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

Alternate initial conditions

$$u(x, 0) = -1.5 \sin(\pi x), \quad u(x, 0) = \sin(2\pi x)$$

Plus two more complex initial conditions that were combination of random sin curves satisfying $f(-1, t) = f(1, t) = 0$ Boundary conditions were therefore satifies in a hard way by applying an output transformation. As an example, for the default case.

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

To measure the accuracy of the different methods, at the end of training the error metric of $L^2$ relative error is used, which compares the prediction $u$ to ground truth $u_{gt}$

$$L^2 = \frac{||\hat{u} - u||_2}{||u||_2} = \frac{\sqrt{\sum (u(i) - u_{gt}(i))^2}}{\sqrt{\sum u_{gt}(i)^2}}$$

### 3.2 Default Condition vs Resamples and vs Points

Looking then at the performance then of the main methods, this was done by looking at the accuracy after running these for different number of resamples (figure 3 - as this was the main contributor to computational cost and therefore sampling accurate to the required degree with smallest number of resamples could be considered better.

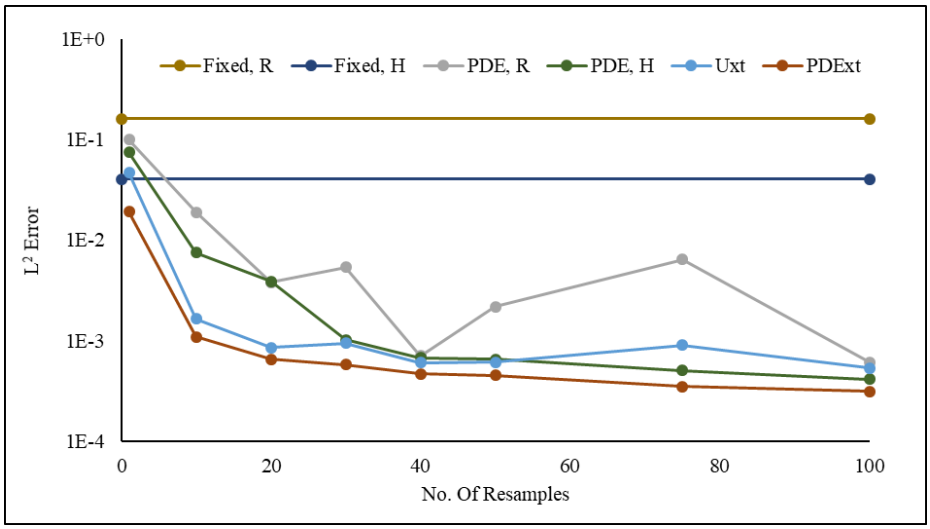



**Fig. 3.** Error vs Number of Resamples

Exploring the application of PDExt with initial values of 1,1 yielded suboptimal results. However, when adjusted to 0.5,1, it showed slight improvements over all previous configurations across various resamples. The Wu (random init) method exhibited the least robustness, likely attributed to random initialization. Despite conducting 50 repeats, the uncertainty in results persisted. As expected more resamples improved accuracy, and error rose above 1E-3 with less than 30 resamples.

The case of no-resample for comparison was allowed to run 400k steps to allow the loss to plateau. It's worth noting how this leaves low numbers of resamples at quite a disadvantage, running only 15k + 2k per resample versus 400k; yet still having a higher accuracy.

A second way of quantifying cost could be looking at the number of points required to achieve a certain accuracy. This was done by looking at error versus number of points (figure 4), using 100 rounds of resampling.

From this graph, the threshold for number of points required to start gaining accuracy varies from method to method. even for the most accurate method

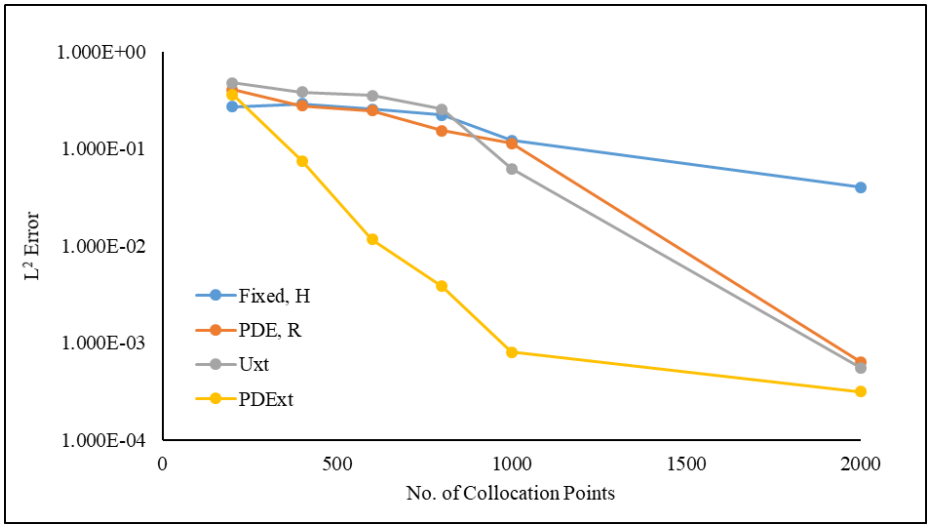



**Fig. 4.** Error vs Number of Collocation Points

(PDExt), the error seems to plateau however between 1E-4 and 1E-3, which the other methods eventually reach at the default 2000 points.

### 3.3 Alternate Initial Conditions

Now we will look at how the different information sources perform with the different initial conditions described in section 3.1

For initial condition 2 and especially for the simpler intial conditions (4 and 5), PDExt performed slightly better than the other methods, becoming more accurate with less resamples and remaining the most accurate at the longest training case (L = 100).

Overall accuracy for all methods was significantly lower in the complex conditions (2 and 3). In these, even using PDExt did not drive the error below 1E-2 and 1E-1 respectively . This could be due to the constrained depth of the network, and fixed number of collocation points used. Interestingly, in these cases the increased number of steps in the fixed distribution methods allowed them to stay a competitive alternative. However, it is worth noting the associated computational costs are therefore higher.

### 3.4 Adjusting $\nu$

Altering the term $\nu$ changes the magnitude of the diffusion term $\nu \cdot du_{xx}$. Lower values increase the shock sharpness, increasing the complexity of the problem whilst using the same PDE. We attempted using a $\nu$ of 0.01 and 0.001, roughly multiplying and dividing by 3 from the default $\nu = \frac{0.01}{\pi}$ used in the previous

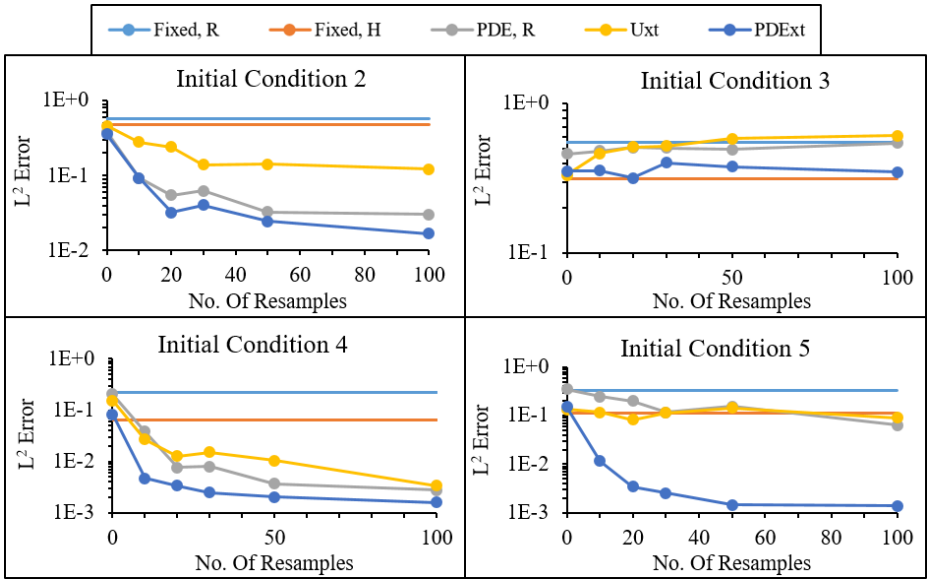



**Fig. 5.** Error versus Number of Resamples. IC. 2 and 3 are randomly generated combinations of sin curves. 4 and 5 are $sin(2\pi)$ and 1.5*amplitude respectively.

sections. The error of using different information sources in adaptive resampling is again compared to fixed uniform methods in figures 6, 7.

For the simpler case 0.01, all adaptive methods behaved very similarly. Considering the fixed methods, the Hammerlsey initial distribution again made a significant difference, with the additional training steps allowing it to outperform the adaptive sampling methods.

For the more complex case of 0.001, all methods struggled to converge to an accurate solution. The difficulty lies in the shock being much sharper.

## 4 Allen-Cahn

## 5 Discussion

In this section...

- Why initial distribution matters. Why hyper-parameters are matter but mentioning having one more thing to tune runs counter to having adaptive sampling... $k$ based on variance, cosine annealing in place of $c$?
- The main analysis of which information sources work best and why, tackling the research question. The ways of looking at performance - max accuracy vs sufficient accuracy under certain computational cost. Under these lenses, how did these methods do. . Mentioning here as well that looking just at # of points or # of resamples is not biggest picture.

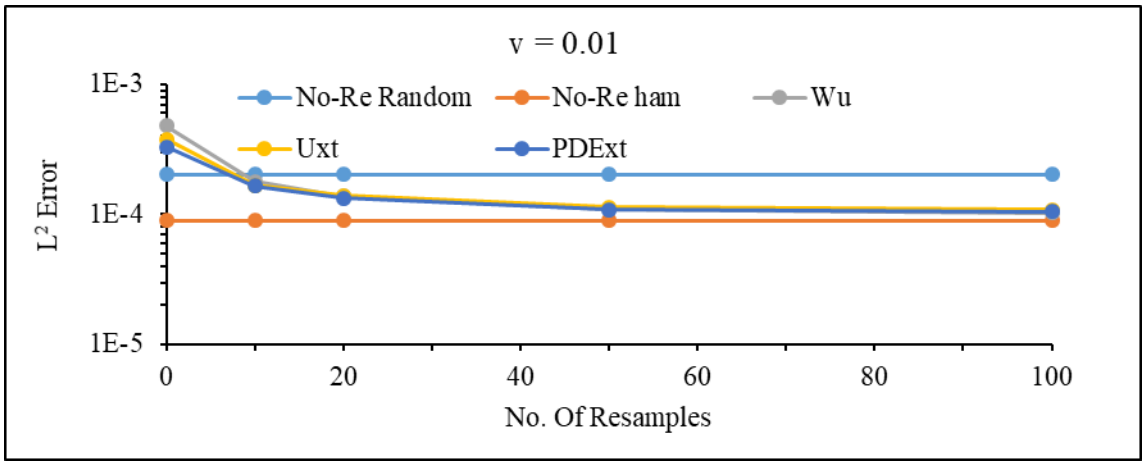



**Fig. 6.** Error vs Resamples, $\nu = 0.01$

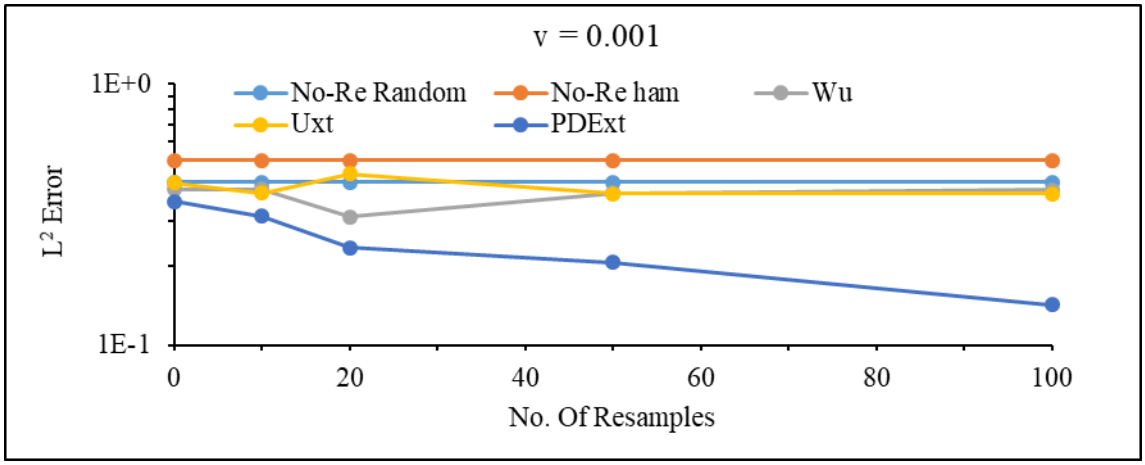



**Fig. 7.** Error vs Resamples, $\nu = 0.001$

- Possible best applications for different information sources looked at.
- Lack of memory in samplings + is there a “good" point distribution we're moving towards and if so methods of optimising to get to that optimal distribution earlier.
- Future work
- Questions out of this scope
  - Does approach work for / how to implement into semi-supervised methods, where collocation point distribution has to factor in for known data locations to avoid redundancy.

### 5.1 Problem setup - initial distribution, hyperparmeters etc

Some methods the average error is higher because some runs get stuck on local minima presumably and have very high error around 1 or 2 orders of magnitude higher. This is somewhat alleviated by using a smarter initial distribution over a random one. Being able to detect these outliers and discard them could improve a lot of methods. For that same reason though it can be determined that the more accurate methods are also more robust, as they minimise getting stuck on local minima. This could be an advantage of residual information?

A comment can be made that the addition of k,c adds complexity; contrary to the philosophy of adaptive sampling. We've quickly tested damping which eliminates the need for c. Simple linear damping not positive or negative, but does remove the need for manual selection fo c.

A more sophisticated alternative is cosine-annealing, which reverts to “uniform" distribution when loss plateaus, and could be grounds for further work. One could similarly come up with ways of eliminating need to manually select k.

### 5.2 Main Analysis of Methods

Curvature absolutely better than Wu, Fixed sampling methods. Maybe not as good as PDExt with simple implementation, but certainly there is some merit that solution estimates can be used as a valid information source. Might be better for certain types of problems? More universal than loss. Could also think of the approach

PDExt vs PDE, is going to focus on same regions, but the probability will be much higher. Perhaps PDE with certain hyperparameters could have same effect.

why was lower k better when info source was second derivative? I think magnitude goes high and the biasing is either too narrowly focused (possibly redundant points?), points aren't well distributed through the area of interest. lowering k accounts for that. The magnitude of values in info vectors could be quantified to automatically select k, lowering it if there is large disparity between areas of interest and the rest of the domain.

### 5.3 'Recommendations'

Is PDExt always better? Why? In what cases would other methods have merit? At a certain number of points, most methods seems to converge to E-3/E-4 and plateau. PDExt getting there earlier could signify it scaling better to bigger problems where perhaps number of points is the main contributor to high computational overhead. However, this could impact the absolute lowest error, or require more rounds of resampling for the solution to converge to a minimum degree of accuracy. This would need some more looking into, perhaps recreating figure 3 for decreasing point number.

### 5.4 Bigger picture commentary - thoughts beyond information source, about method

Lack of memory in resamplings. Wanting to test if network remembers old points after a few resamples or whether it loses accuracy.

**Acknowledgments.**

## References

1. C, Wu.: A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. Comput. Methods Appl. Mech. Engrg. **403** (2023) 115671
2. Raissi, ...
3. 
4. Wang, Y.:Asymptotic Self-Similar Blow-Up Profile for Three-Dimensional Axisymmetric Euler Equations Using Neural Networks. Phys. Rev. Lett **130** (2023) 244002
5. Lai, C.:
6. McClenny, L.: Self-Adaptive Physics-Informed Neural Networks using a Soft Attention Mechanism
7. 
8. Gao, Z.: Failure-Informed Adaptive Sampling for PINNs. SIAM Journal on Scientific Computing. **45** (2023) A1971-A1994