FLOW-MATCHING SAMPLING IN PHYSICS-INFORMED NEURAL NETWORKS FOR PDES WITH SHARP SOURCE TERMS

Yana Khassan Nibal, Dmitry Efremenko, Fedor Buzaev & Denis Derkach
Faculty of Computer Science
Higher School of Economics
Moscow, Russia
fa.buzaev@hse.ru

ABSTRACT

Singularities in the source functions of partial differential equations (PDEs) pose significant challenges for physics-informed neural networks (PINNs), often leading to numerical instability and requiring a large number of sampling points to achieve accurate solutions, which increases computational costs. In this paper, we propose a novel sampling strategy that use diffusion models for generative sampling based on the distribution of PDE residuals. Using the optimal transport coupling flow-matching technique, our method adaptively generates additional sampling points in regions with high residuals, enhancing both solution accuracy and efficiency. Unlike existing approaches, which explicitly model probability densities proportional to residuals, our technique uses flow matching to directly sample from complex residual distributions, improving PINN performance for problems with sharply localized source terms. We validate our method on the Poisson equation with singular source functions and the linear elasticity equation in materials with complex geometries, achieving up to 10× lower MSE compared to baseline methods and outperforming normalizing flow-based sampling.

1 INTRODUCTION

Physics-informed neural networks (PINNs) provide a versatile framework for solving partial differential equations (PDEs) by embedding physics-based constraints directly into the training process of neural networks. PINNs have gained widespread popularity across a variety of real-world simulations Raissi et al. (2019); Farea et al. (2024). Compared to traditional numerical solvers, they offer several key advantages, including straightforward implementation for both forward and inverse problems, as well as the capability to handle high-dimensional systems effectively Miao & Chen (2021).

Despite their success in numerous PDE-related applications, PINNs face significant challenges when applied to systems with intricate characteristics. They often fail to converge to accurate solutions. As described by the F-principle Xu (2020), PINNs capture low-frequency solution components earlier in training, while resolving high-frequency features requires prolonged epochs. This inefficiency becomes especially problematic for PDEs with high-frequency solutions, as demonstrated in Chuprov et al. (2023); Buzaev et al. (2023); Chuprov et al. (2024). For PDEs with source functions exhibiting sharp peaks, accurate solutions often require sampling a large number of points in regions with steep gradients to adequately evaluate the loss function. This increased sampling burden significantly complicates training, both in terms of computational cost and optimization complexity. One promising approach to mitigate these challenges is the design of algorithms that optimize sampling distributions Gao et al. (2023), enabling more efficient training and improving PINN performance in these scenarios.

In this paper, we introduce a novel sampling strategy to address these challenges. Our approach employs diffusion models for generative sampling based on the distribution of loss residuals. By applying the optimal transport flow-matching technique, we dynamically generate additional sampling points in regions with high PDE residuals. This strategy enhances the accuracy of PINN solutions while improving their efficiency for problems with sharply peaked source functions, making them better suited for solving intricate systems.

2 PINN OVERVIEW

Consider a domain Ω, where we aim to solve a partial differential equation (PDE) of the form $\mathcal{D}u(\mathbf{x}) = s(\mathbf{x})$. Here, $s(\mathbf{x})$ represents the source function, $\mathcal{D}$ is the differential operator, and $\mathbf{x}$ is a d-dimensional vector within the domain Ω. The domain is bounded by $\partial\Omega$, where the solution is subject to boundary conditions of the form $\mathcal{B}u(\mathbf{x}) = g(\mathbf{x})$ for $\mathbf{x} \in \partial\Omega$. Here, $\mathcal{B}$ represents the boundary operator, such as Neumann or Dirichlet conditions. Together, the problem can be expressed as:

$$\begin{aligned} \mathcal{D}u(\mathbf{x}) &= s(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \\ \mathcal{B}u(\mathbf{x}) &= g(\mathbf{x}), \quad \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{1}$$

To approximate the solution of Equation equation 1, we consider a neural network $u(\mathbf{x}, \psi)$, where ψ denotes the trainable parameters of the network. The training process involves optimizing ψ to minimize the following total loss function:

$$\min_{\psi} L(\psi) = L_{PDE}(\psi) + L_{BC}(\psi), \tag{2}$$

where

$$L_{PDE}(\psi) = \sum_{\mathbf{x}_i \in \mathcal{S}_k} \left(\mathcal{D}u(\mathbf{x}_i, \psi) - s(\mathbf{x}_i)\right)^2 \tag{3}$$

is the PDE residual loss, and

$$L_{BC}(\psi) = \sum_{\mathbf{x}_i \in \partial\mathcal{S}_k} \left(\mathcal{B}u(\mathbf{x}_i, \psi) - g(\mathbf{x}_i)\right)^2 \tag{4}$$

is the boundary condition loss. Here, $\mathcal{S}_k$ and $\partial\mathcal{S}_k$ represent the sets of collocation points in the domain Ω and on the boundary $\partial\Omega$, respectively. The optimization process minimizes both the PDE residuals and the boundary condition errors by evaluating the loss function at the selected collocation points.

3 RELATED WORK

Adaptive sampling can improve the performance of PINNs by refining the selection of collocation points. Studies like Wu et al. (2023) demonstrate that uniform sampling often yields better results than mesh grids, avoiding trivial solutions. A prominent adaptive sampling approach is the Residual Aided Refinement (RAR) algorithm Lu et al. (2021), which iteratively adds points in regions with high residuals. Building on this, the Residual Adaptive Distribution (RAD) method Wu et al. (2023) improves accuracy by dynamically redistributing points based on PDE residuals, similar to importance sampling in Monte-Carlo methods Elvira et al. (2017). Other methods, such as Nabian et al. (2021), propose a distribution proportional to residual errors at selected seed points to prioritize regions with large residuals.

More advanced sampling strategies are based on generative models. For example, flow-matching methods Lipman et al. (2022) improve efficiency in diffusion models by refining the loss function of continuous normalizing flows. These models provide high-quality samples and stable training, making them suitable for generating collocation points in PINNs. In Tang et al. (2023a), a normalizing flow was used for adaptive sampling to solve Poisson equations with highly peaked source functions, while in Wang et al. (2024) this approach was extended to cavity flow problems. Another related approach, namely AAS-PINN Tang et al. (2023b), used the Wasserstein generative adversarial network (WGAN) to generalize sampling for normalized residual distributions. Although normalizing flows such as KR-net offer invertible transformations and efficient density estimation, they struggle to approximate more complex residual distributions and may suffer from computational inefficiencies Wang et al. (2025); Ho et al. (2019).

To address these limitations, we propose a PINN with novel flow-matching sampling (FMS) strategy (FMS PINN), which avoids the drawbacks of normalizing flows and focuses on efficiently generating points in regions with large residuals.

4 FMS PINN

Let us briefly recall the concept of flow matching. Flow matching Lipman et al. (2022) is a generative algorithm designed to handle data distributions with high complexity. Unlike normalizing flow methods, it does not require the neural network transformation to be invertible. Instead of explicitly modeling the probability density function $p_1(x)$, flow matching enables sampling from some target distribution $p_1(x)$ by modeling the dynamics of a vector field flow. This process begins with a simple Gaussian distribution $p_0(x)$, which is transformed into a more complex distribution $p_1(x)$ through a flow field $f(x, t)$, where time t varies from 0 to 1. Sampling from the target distribution is governed by the solution of the following ordinary differential equation (ODE):

$$\begin{cases} dX_t = f(X_t, t)\, dt, \\ X_0 \sim p_0. \end{cases} \tag{5}$$

The vector field $f(x, t)$, responsible for transporting samples from a simple base distribution $p(x_0)$ to a more complex target distribution $q(x_1)$, is parameterized by a neural network, which is trained by minimizing the following loss function:

$$\mathcal{L} = \mathbb{E}_{t,\, x_0 \sim p(x_0),\, x_1 \sim q(x_1)} \left\| f(x, t) - (x_1 - x_0) \right\|^2 , \tag{6}$$

where x_0 and x_1 denote samples from the base and target distributions, respectively. This neural network approximates the time-dependent vector field that guides the evolution of samples from x_0 to x_1.

The idea of flow matching can be applied to design an adaptive sampling algorithm that addresses the challenge of non-uniform error distribution in PINN training. By interpreting the residual distribution as a target probability density, flow matching provides a principled framework to construct a neural vector field that transports samples from a simple prior toward regions of high residuals. This approach enables the generation of new collocation points that are adaptively concentrated in the most informative regions of the domain.

At the first stage, the algorithm calculates the residual $r(x_i, \psi^k)$ at each point, defined as:

$$r(x_i, \psi^k) = \|\mathcal{D}u(x_i, \psi^k) - s(x_i)\|, \tag{7}$$

where ψ^k represents the PINN training parameters at iteration k. A weighted bootstrap procedure is then used to construct a sub-sample $\mathcal{A}_i$ of size M, proportional to the residual values. This sample is constructed using a multinomial distribution with probabilities proportional to the residuals based on the training points of the PINN, $\mathcal{S}_k$, where the initial set of points for training the PINN is drawn from a uniform distribution over the domain Ω.

The vector field neural network $f(x, t)$ is trained on this sub-sample $\mathcal{A}_i$ by minimizing the flow matching loss. The trained vector field $f(x, t)$ governs the dynamics of transforming prior samples from $p_0(x)$ to match the residual distribution.

At each step, the target distribution $p_1(x)$ of the vector field is the normalized residual distribution $p_r(x, \psi^k)$, that reads as follows:

$$p_r(x, \psi^k) = \frac{r(x, \psi^k)}{\int_{x \in \Omega} r(x, \psi^k) dx}. \tag{8}$$

In this context, it is assumed that the prior distribution $p_0(x)$ follows a normal distribution. Points are generated by solving Eq.equation 19 numerically using the Euler-Maruyama discretization scheme, generating a new sample $\mathcal{V}_i$ of points. The points $\mathcal{V}_i$ are combined with the set of training points from the previous step, $\mathcal{S}_k$, and are used in the next stage to fine-tune the PINN neural network $u(x, \psi^K)$ on this combined set. In this way, $\mathcal{V}_i$ enriches the training set with new points sampled according to the distribution of residuals $p_r(x, \theta^k)$.

After the PINN is fine-tuned on the new sample $\mathcal{S}_{k+1}$, the updated parameters of the PINN, ψ^k, result from this retraining procedure. The algorithm then calculates the new residuals $r(x, \psi^k)$, and the procedure continues.

The application of this algorithm in the PINN framework is summarized below and illustrated in Figure 1: [H]

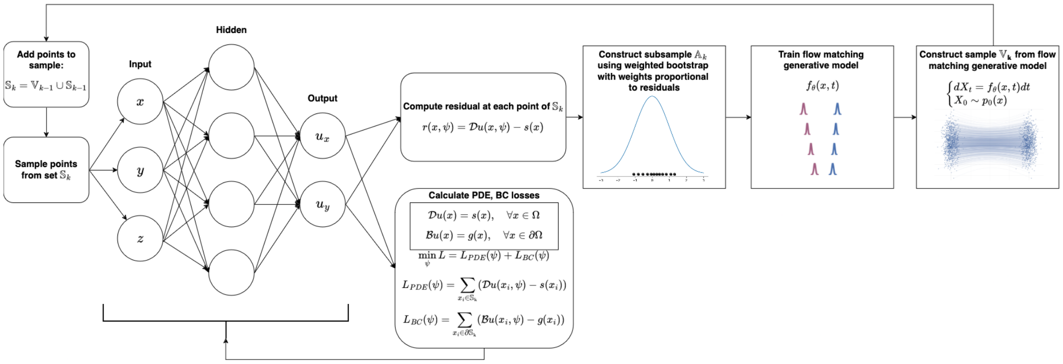

Add points to sample:
$\mathbb{S}_k = \mathbb{V}_{k-1} \cup \mathbb{S}_{k-1}$
Sample points from set $\mathbb{S}_k$
Input
x
y
z
Hidden
Output
u_x
u_y
Compute residual at each point of $\mathbb{S}_k$
$r(x, \psi) = \mathcal{D}u(x, \psi) - s(x)$
Calculate PDE, BC losses
$\mathcal{D}u(x) = s(x), \quad \forall x \in \Omega$
$\mathcal{B}u(x) = g(x), \quad \forall x \in \partial\Omega$
$\min_{\psi} L = L_{PDE}(\psi) + L_{BC}(\psi)$
$L_{PDE}(\psi) = \sum_{x_i \in \mathbb{S}_k} (\mathcal{D}u(x_i, \psi) - s(x_i))$
$L_{BC}(\psi) = \sum_{x_i \in \partial\mathbb{S}_k} (\mathcal{B}u(x_i, \psi) - g(x_i))$
Construct subsample $\mathbb{A}_k$ using weighted bootstrap with weights proportional to residuals
−3
−2
−1
0
1
2
3
Train flow matching generative model
$f_\theta(x, t)$
Construct sample $\mathbb{V}_{\mathbf{k}}$ from flow matching generative model
$\begin{cases} dX_t = f_\theta(x, t)dt \\ X_0 \sim p_0(x) \end{cases}$

Figure 1: Overview of the FMS PINN algorithm. The diagram illustrates the iterative process of residual evaluation, flow-matching model training, generative sampling of new collocation points, and PINN retraining, highlighting the integration of flow-matching within the adaptive sampling loop.

1. Sample N points uniformly from the domain Ω to form the initial set $\mathcal{S}_0$.
2. Train the PINN model on $\mathcal{S}_0$ by minimizing the loss:
$$L(\psi^0, N) = L_{PDE,N}(\psi^0) + L_{BC,N}(\psi^0), \tag{9}$$
where L_{PDE} and L_{BC} are defined in Eq. (3) and Eq. (4), respectively.
3. For each stage $k = 1, \ldots, K$:
   (a) Calculate residuals $r(x_i, \psi^{k-1})$ at each point x_i in $\mathcal{S}_k$.
   (b) Resample m points from $\mathcal{S}_k$ proportionally to corresponding residuals in order to make the set $\mathcal{A}_k$ (i.e. using the weighted bootstrap technique).
   (c) Train the vector field neural network $f(x, t)$ on $\mathcal{A}_k$ by optimizing the flow matching loss (6).
   (d) Using found $f(x, t)$ and Eq. (19) we find new points $\mathcal{V}_k$.
   (e) Update the collocation set: $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \mathcal{V}_k$.
   (f) Retrain the PINN model on points $\mathcal{S}_{k+1}$.
4. Return the final trained PINN model $u(x, \psi^K)$.

5 Numerical results

In this section we compare the proposed FMS PINN algorithm with DAS PINN, AAS PINN uniform sampling, RAD and RAR. PINNs are trained on a single GPU V-100.

5.1 Poisson equation

5.1.1 Two-dimensional Poisson equation with nine peaks

Let us consider the Poisson equation in the two-dimensional case:
$$\begin{aligned} -\Delta u(\mathbf{x}) &= s(\mathbf{x}), \quad \mathbf{x} \in \mathbb{D}, \\ u(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \partial\mathbb{D}, \end{aligned} \tag{10}$$
where $\mathbf{x} = (x, y)$ and $\mathbb{D} = [-1, 1] \times [-1, 1]$. To rigorously evaluate the performance of our PINN approach, we start by assuming an exact solution $u(\mathbf{x})$ for the Poisson equation. This assumed solution is a superposition of nine highly peaked trial functions:
$$u(\mathbf{x}) = \sum_{i=0}^{8} q_i(\mathbf{x}, c_{i0}, c_{i1}), \tag{11}$$

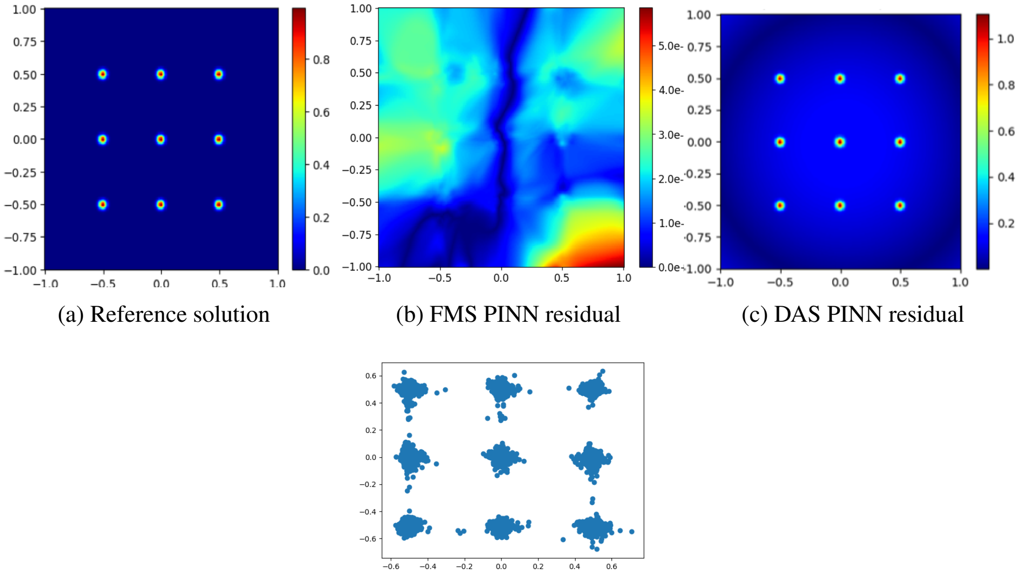
where

$$q_i(\mathbf{x}, c_{i0}, c_{i1}) = -e^{-1000\left((x-c_{i0})^2+(y-c_{i1})^2\right)}, \tag{12}$$

and the coordinates of the peak centers are defined as:

$$(c_{i0}, c_{i1}) = (-0.5, 0.5) + \left(\frac{\mathrm{mod}(i,3)}{2}, 0\right) + \left(0, \frac{\lfloor i/3 \rfloor}{2}\right). \tag{13}$$

From this assumed solution, we compute the source function $s(\mathbf{x})$ by applying the Laplace operator. Similarly, the boundary conditions $g(\mathbf{x})$ are defined by evaluating $u(\mathbf{x})$ on the boundary $\partial\mathbb{D}$. This ensures that the problem is fully specified, with an analytically known exact solution. The reference solution is shown in Figure 2a. PINN is taken as a fully connected network (FCN) with 6 layers, each containing 64 neurons with ReLU activation function. Figures 2b and 2c show the residuals of solutions obtained with the FMS PINN and the DAS PINN, respectively. While the DAS PINN struggles to accurately capture all nine peaks, the FMS PINN effectively resolves the peaks, maintains a solution close to zero outside these regions, and exhibits smaller residuals across most of the domain.



(a) Reference solution

(b) FMS PINN residual

(c) DAS PINN residual

(d) Resampled points by FMS PINN

Figure 2: Comparison of the reference solution and residual error profiles for the Poisson equation with a nine-peak source function. The FMS PINN effectively resolves all peaks and maintains low residuals, while the DAS PINN fails to fully capture all features of the solution.

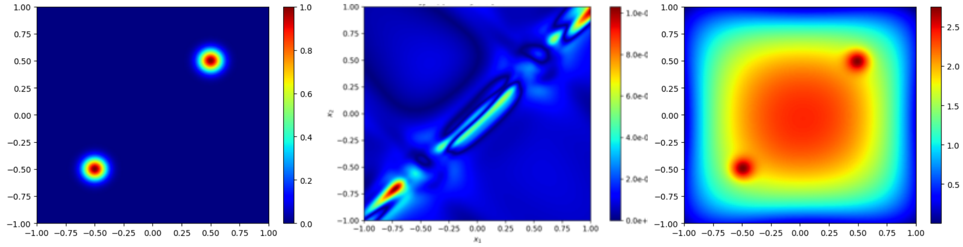
Figure 2d illustrates the resampled points added by the flow matching method, demonstrating its ability to prioritize high-residual regions effectively. Let us note that PINN with uniform sampling fails to solve the equation when using the same neural network architecture, whereas the training times for DAS PINN and FMS PINN are nearly identical.

5.2 Five-dimensional Poisson equation with two peaks

We consider a five-dimensional Poisson equation problem. Following Zhang et al. (2024), the reference solution for this problem is given by:

$$u(\mathbf{x}) = \sum_{i=1}^{2}\sum_{j=1}^{5} \exp\left[-K\left(\left(x_j - x_j^i\right)^2\right)\right], \tag{14}$$

where $K = 100$. The two peaks are centered at $(0.5, 0.5, 0, 0, 0)$ and $(-0.5, -0.5, 0, 0, 0)$. A two-dimensional projection of the reference solution, along with residual profiles for the FMS PINN and DAS PINN, is shown in Figure 3.



(a) Reference solution (b) FMS PINN residual (c) DAS PINN residual

Figure 3: Two-dimensional projections for the five-dimensional Poisson problem with two peaks: (a) reference solution, (b) residuals from FMS PINN, and (c) residuals from DAS PINN. FMS PINN captures the peak features more accurately, exhibiting lower residuals across the domain.

To compute numerical errors efficiently and perform inference, we adopt the two-stage sampling strategy described in Zhang et al. (2024).

For training, we initially draw 100,000 points from a uniform distribution and 60,000 points from Gaussian distributions centered at the peaks. Additionally, 10,000 boundary points are included, resulting in a set used to compute numerical errors for the five-dimensional problem. We use the Adam optimizer with a learning rate of 0.001. The flow matching PINN (FMS PINN) is trained with an additional 40,000 points sampled at each resampling stage. These points are generated from the vector field trained via the optimal transport flow matching objective, using a weighted bootstrap procedure based on the residual distribution from the previous stage.

We compare the FMS PINN with the DAS PINN approach, AAS PINN, residual sampling methods, and uniform sampling using the same number of training points (100,000 uniform and 60,000 Gaussian). The DAS PINN and AAS PINN employ the KR-net architecture for normalizing flows, while the main difference is that the DAS PINN is based on cross-entropy loss, whereas the AAS PINN incorporates the Wasserstein distance between the residual-induced distribution and the uniform distribution into its loss. As illustrated in Figure 3, the FMS PINN captures the features of the solution with much smaller residuals compared to the DAS PINN and the AAS PINN, which struggle to represent the solution accurately under the same conditions. Other methods, including DAS PINN, AAS PINN, uniform sampling, RAR, and RAD approaches, failed to converge to the solution for the same sample size, unlike the FMS PINN method.

Table 2 shows the mean squared error (MSE) comparison between the DAS PINN, the FMS PINN, uniform sampling, RAR, RAD and AAS. The results reveal that the FMS PINN achieves significantly lower MSE values, demonstrating its superior accuracy and efficiency.

Table 1: MSE comparison for Poisson equation solutions

Method	2 peaks in 5D	9 peaks in 2D	2 peaks in 2D	Klein-Gordon
FMS PINN	**6.1e-3**	**1.8e-4**	**7.7e-5**	3.9e-2
DAS PINN	2.3	1.1e-1	5.2e-4	7.1e-1
Uniform	0.2	5.5e-4	2.7e-4	3.8e-2
RAR	0.4	6.1e-4	5.7e-4	5.1e-2
RAD	0.5	8.5e-4	7.1e-3	5.6e-2
AAS	120	2.2e-4	1.1e-4	8.1e-2

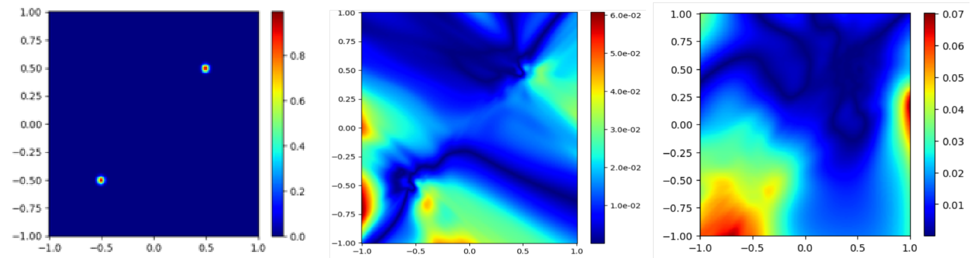
5.3 Two-dimensional Poisson equation with two peaks

Figure 4 illustrates the performance of different PINN sampling strategies for solving the Poisson equation with two sharply peaked source terms. Figure 7a shows the ground-truth solution, highlighting the localized structure of the source. Residual of FMS PINN (see Figure 7b) is smaller by an order of magnitude than that of DAS PINN (Figure 7c). Compared to residual sampling methods and generative sampling, the FMS PINN MSE is more than 2 times smaller than the MSE corre-

Table 2: Elapsed time of the sampling stage for the Poisson equation in seconds

Method	2 peaks in 5D	9 peaks in 2D	2 peaks in 2D
FMS PINN	840	72	80
DAS PINN	6675	782	960
Uniform	6	2	1
RAR	15	5	3
RAD	16	5	3
AAS	3582	943	1384

sponding to RAD, RAR, and AAS PINN. It is slightly better than uniform sampling, and the FMS PINN has more stable quality compared to uniform sampling. The corresponding MSE values are given in Table 2. (see more technical details about the experiments in the supplementary material technical appendix)



(a) Reference solution (b) FMS PINN residual (c) DAS PINN residual

Figure 4: The Poisson equation with two sharp source peaks: (a) reference solution, (b) residuals from FMS PINN, and (c) residuals from DAS PINN.

5.4 Linear elasticity equation

In this section, we solve a specific instance of the mechanical equilibrium equation for a rectangular plate with a geometric inclusion made of a second material, commonly referred to as the linear elasticity equations. The system is given as:

$$\begin{aligned}
&(1-\nu)\frac{\partial^2 u_x}{\partial x^2} + \nu\frac{\partial^2 u_y}{\partial x \partial y} + \frac{1}{2}(1-2\nu)\left(\frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_y}{\partial x \partial y}\right) = 0,\\
&\frac{1}{2}(1-2\nu)\left(\frac{\partial^2 u_x}{\partial x \partial y} + \frac{\partial^2 u_y}{\partial x^2}\right) + \nu\frac{\partial^2 u_x}{\partial x \partial y} + (1-\nu)\frac{\partial^2 u_y}{\partial y^2} = 0,
\end{aligned} \tag{15}$$

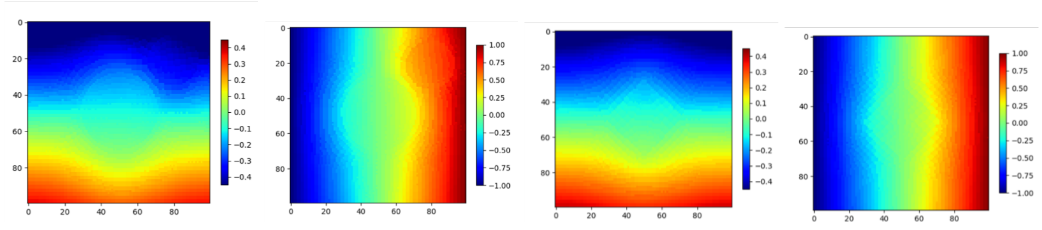
where E and ν are the Young's modulus and Poisson's ratio, describing the material properties, while u_x and u_y represent horizontal and vertical displacements, respectively.

We consider a square plate with $(x, y) \in [x_{\text{min}}, x_{\text{max}}] \times [y_{\text{min}}, y_{\text{max}}]$. Dirichlet boundary conditions are applied for the horizontal displacement:

$$\begin{cases}
u_x(x,y) = -0.01, & x = x_{\text{min}},\ \forall y \in [y_{\text{min}}, y_{\text{max}}],\\
u_x(x,y) = 0.01, & x = x_{\text{max}},\ \forall y \in [y_{\text{min}}, y_{\text{max}}],\\
u_y(x,y) = 0.0, & \text{on the boundary.}
\end{cases}$$

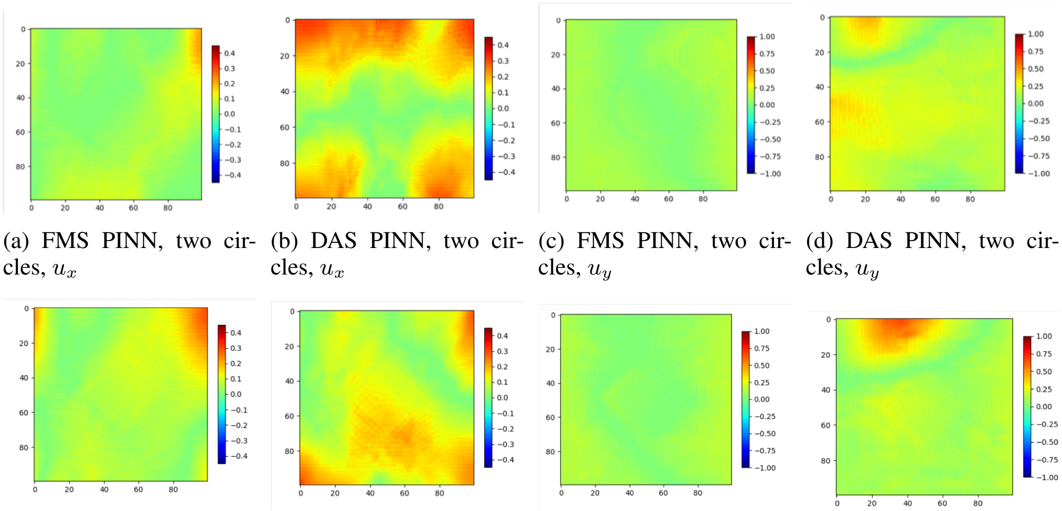
The plate consists of a base material with a geometric inclusion of a second material, characterized by a different value of E. We consider two inclusion geometries: (1) two circular inclusions and (2) a diamond-shaped inclusion. The reference solutions for both cases are shown in Figure 5.

The PINN architecture consists of five separate fully connected networks, each with 5 layers and 40 neurons per layer, as suggested in Rezaei et al. (2022). The model was trained for 30,000 epochs using the Adam optimizer with the ‘ReduceLROnPlateau‘ learning rate scheduler.



(a) Two circles, u_x (b) Two circles, u_y (c) Diamond, u_x (d) Diamond, u_y

Figure 5: Reference solutions for linear elasticity with two inclusion geometries.



(a) FMS PINN, two circles, u_x (b) DAS PINN, two circles, u_x (c) FMS PINN, two circles, u_y (d) DAS PINN, two circles, u_y

(e) FMS PINN, diamond, u_x (f) DAS PINN, diamond, u_x (g) FMS PINN, diamond, u_y (h) DAS PINN, diamond, u_y

Figure 6: Error profiles: linear elasticity with circular and diamond inclusions. FMS PINN outperforms DAS PINN in resolving fine details, especially circular inclusions.

Figure 6 presents the error profiles for the FMS PINN and DAS PINN for both the two-circle and diamond inclusion cases. The flow-matching approach yields significantly lower error in the two-circle case. For the diamond inclusion case, it achieves performance comparable to DAS PINN, while demonstrating improved stability and better localization of features. For the elasticity equation with a two-circle setup, one stage of flow matching took 65 seconds on a single GPU, while each normalizing flow training session required 13 minutes.

Table 3 presents MSE comparison for different sampling strategies applied to the linear elasticity equation with two types of inclusions. FMS PINN consistently achieves the lowest MSE in three out of four displacement components across both test cases. For the two-circle inclusion geometry, FMS PINN significantly outperforms all other methods, especially in the u_x component, where the error is an order of magnitude smaller than that of DAS PINN and AAS PINN. In the diamond-shaped inclusion case, FMS PINN remains competitive and achieves the best result for u_x, while AAS PINN and DAS PINN slightly outperform it in the u_y component.

5.5 Klein–Gordon equation

The Klein–Gordon equation is a nonlinear hyperbolic partial differential equation that describes the evolution of a relativistic scalar field. In two spatial dimensions, it is given by:

$$\begin{aligned}\frac{\partial^2 u(t,x,y)}{\partial t^2} - \frac{\partial^2 u(t,x,y)}{\partial x^2} - \frac{\partial^2 u(t,x,y)}{\partial y^2} &+ u(t,x,y)^2 \\ &= q(t,x,y) \end{aligned} \tag{16}$$

Table 3: MSE for the linear elasticity equation solutions

Method	Two Circles		Diamond	
	u_x	u_y	u_x	u_y
FMS PINN	**1.5e-3**	**7.1e-3**	4.6e-3	**6.2e-3**
DAS PINN	1.7e-2	1.2e-2	7.1e-3	8.6e-3
Uniform	4.5e-3	7.4e-3	**4.4e-3**	7.1e-3
RAD	8.3e-3	8.3e-3	8.2e-3	8.4e-3
RAR	1.1e-2	7.2e-3	7.5e-3	8.1e-3
AAS PINN	1.3e-2	2.5e-2	4.8e-3	8.5e-3

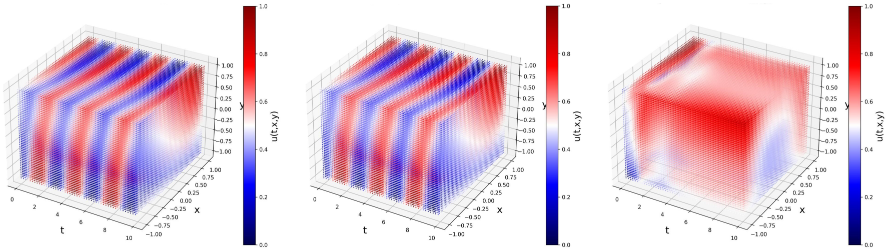
where $(x, y) \in [-1, 1]^2$, $t \in [0, 10]$, and $q(t, x, y)$ is the source term taken as

$$q(t, x, y) = u(t, x, y)^2 - 4\,u(t, x, y). \tag{17}$$

The exact solution for this problem is known and given by:

$$u_{\text{exact}}(t, x, y) = (x + y)\cos(2t) + xy\sin(2t). \tag{18}$$

A comparison of the DAS PINN and FMS PINN solutions with the reference solution is presented in Figure 7. The corresponding averaged MSE values, including those for other sampling strategies, are summarized in Table 2. As expected, since the Klein–Gordon problem does not contain sharply localized source features, the FMS PINN performs comparably to uniform sampling in terms of accuracy. Nonetheless, it achieves consistently better results than both DAS PINN and AAS PINN, indicating its robustness even in smooth-source scenarios.



(a) Reference solution

(b) FMS PINN solution

(c) DAS PINN solution

Figure 7: Klein-Gordon equation: (a) reference solution, (b) solution of FMS PINN, and (c) solution of DAS PINN.

6 Conclusion

In this paper, we proposed a novel flow-matching sampling strategy for training physics-informed neural networks. The method adaptively generates collocation points in high-residual regions using a generative diffusion model guided by optimal transport. Numerical experiments show that FMS PINN significantly improves accuracy in problems with sharp source terms. For instance, in the 2D Poisson problem with two peaks, it achieves an MSE of 7.7e-5, compared to 5.2e-4 for DAS PINN and 2.7e-4 for uniform sampling. In the 5D Poisson case, FMS reduces error to 6.1e-3 versus 2.3 for DAS and 0.2 for uniform. For the linear elasticity problem, it yields the lowest error in 3 of 4 cases, with up to 10× improvement. Even for the Klein–Gordon equation with smooth sources, FMS PINN still outperforms DAS PINN (MSE 3.9e-2 vs. 7.1e-1) and matches uniform sampling. These results confirm the robustness and efficiency of the proposed method across diverse PDE settings. Future work will focus on applying FMS PINN to a broader range of PDEs to realize its potential.

Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement

000000C313925P4E0002 and the agreement with HSE University 139-15-2025-009.
This research is supported in part through computational resources of HPC facilities at HSE University.

REFERENCES

Fedor Buzaev, Jiexing Gao, Ivan Chuprov, and Evgeniy Kazakov. Hybrid acceleration techniques for the physics-informed neural networks: a comparative analysis. *Machine Learning*, 113(6): 3675–3692, December 2023. ISSN 1573-0565. doi: 10.1007/s10994-023-06442-6. URL `http://dx.doi.org/10.1007/s10994-023-06442-6`.

I. Chuprov, J. Gao, D. Efremenko, E. Kazakov, F. Buzaev, and V. Zemlyakov. Optimization of physics-informed neural networks for solving the nolinear Schrödinger equation. *Doklady Mathematics*, 108(S2):S186–S195, December 2023. ISSN 1531-8362. doi: 10.1134/s1064562423701120. URL `http://dx.doi.org/10.1134/S1064562423701120`.

I. A. Chuprov, J. Gao, D. S. Efremenko, F. A. Buzaev, and V. V. Zemlyakov. Solution of the multimode nonlinear schrödinger equation using physics-informed neural networks. *Doklady Mathematics*, 110(S1):S15–S24, December 2024. ISSN 1531-8362. doi: 10.1134/s1064562424602105. URL `http://dx.doi.org/10.1134/S1064562424602105`.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F. Bugallo. Improving population monte carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131:77–91, 2017. ISSN 0165-1684. doi: 10.1016/j.sigpro.2016.07.012. URL `http://dx.doi.org/10.1016/j.sigpro.2016.07.012`.

Amer Farea, Olli Yli-Harja, and Frank Emmert-Streib. Understanding physics-informed neural networks: Techniques, applications, trends, and challenges. *AI*, 5(3):1534–1557, 2024. ISSN 2673-2688. doi: 10.3390/ai5030074. URL `http://dx.doi.org/10.3390/ai5030074`.

Zhiwei Gao, Liang Yan, and Tao Zhou. Failure-informed adaptive sampling for pinns. *SIAM Journal on Scientific Computing*, 45(4):A1971–A1994, July 2023. ISSN 1095-7197. doi: 10.1137/22m1527763.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International conference on machine learning*, pp. 2722–2730. PMLR, 2019.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1):208–228, 2021.

Zheng Wu Miao and Yong Chen. Physics-informed neural networks method in high-dimensional integrable systems. *Modern Physics Letters B*, 36(01), December 2021. ISSN 1793-6640. doi: 10.1142/s021798492150531x. URL `http://dx.doi.org/10.1142/S021798492150531X`.

Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021.

Maziar Raissi, Paris Perdikaris, and George 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*, 378:686–707, 2019.

Shahed Rezaei, Ali Harandi, Ahmad Moeineddin, Bai-Xiang Xu, and Stefanie Reese. A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method. *Computer Methods in Applied Mechanics and Engineering*, 401:115616, 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115616. URL `http://dx.doi.org/10.1016/j.cma.2022.115616`.

Kejun Tang, Xiaoliang Wan, and Chao Yang. Das-pinns: A deep adaptive sampling method for solving high-dimensional partial differential equations. *Journal of Computational Physics*, 476: 111868, 2023a.

Kejun Tang, Jiayu Zhai, Xiaoliang Wan, and Chao Yang. Adversarial adaptive sampling: Unify pinn and optimal transport for the approximation of pdes. *arXiv preprint arXiv:2305.18702*, 2023b.

Xili Wang, Kejun Tang, Jiayu Zhai, Xiaoliang Wan, and Chao Yang. Deep adaptive sampling for surrogate modeling without labeled data. *arXiv preprint arXiv:2402.11283*, 2024.

Yanjie Wang, Feng Liu, Faguo Wu, and Xiao Zhang. Navigating pinns via maximum residual-based continuous distribution. *Communications in Nonlinear Science and Numerical Simulation*, 141: 108460, 2025. ISSN 1007-5704. doi: 10.1016/j.cnsns.2024.108460. URL http://dx.doi.org/10.1016/j.cnsns.2024.108460.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. 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.

Zhi-Qin John Xu. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, June 2020. ISSN 1991-7120. doi: 10.4208/cicp.oa-2020-0085. URL http://dx.doi.org/10.4208/cicp.OA-2020-0085.

Zhengqi Zhang, Jing Li, and Bin Liu. Annealed adaptive importance sampling method in pinns for solving high dimensional partial differential equations. *arXiv preprint arXiv:2405.03433*, 2024.

A Appendix: Flow matching derivation

Sampling from the target distribution is governed by the solution of the following ordinary differential equation (ODE):

$$\begin{cases} dX_t = f(X_t, t)\, dt, \\ X_0 \sim p_0. \end{cases} \tag{19}$$

The relationship between the conditional probability distribution $p_t(x \mid x_1)$ and the vector field $f(x,t)$ is governed by the continuity equation:

$$\frac{d}{dt} p_t(x) = -\operatorname{div}\left(p_t(x) f(x,t)\right), \tag{20}$$

as shown in Lipman et al. (2022) (see Theorem 1 therein).

In this study, we consider a specific instance of the optimal transport vector field, conditioned on a Gaussian noise vector z, defined as:

$$f(x, t \mid z) = x_1 - x_0 \tag{21}$$

$$x_t = (1-t)\, x_0 + tz. \tag{22}$$

Let us assume that $f(x,t)$ is unknown and the functional below is not feasible to evaluate.

$$\int_0^1 E_{p_t(x)} ||f_\theta(x,t) - f(x,t)|| \tag{23}$$

It turns out that it can be reduced to the conditional flow dynamics $f_\theta(x,t|z)$, where z can be a latent variable that is sampled from a prior distribution. That is why according to Theorem 2 Lipman et al. (2022) the intractable integral in 23 can be reduced to the following optimization problem that is tractable to solve:

$$\int\limits_0^1 \mathbb{E}_{q(z)p_t(x|z)} \|f_\theta(x,t) - f(x,t|z)\|^2 \mathrm{d}t \to \min_\theta, \tag{24}$$

The minimum that is a solution to this optimization problem is attained on the real vector field $f(x, t)$.

In order to find the minimum of such optimization problem the gradient can be calculated as an expectation that is found using the Monte-Carlo approximation, namely,

$$\nabla_\theta \int\limits_0^1 \mathbb{E}_{q(z)p_t(x|z)} \|f_\theta(x,t) - f(x,t|z)\|^2 \mathrm{d}t, \tag{25}$$

where $z \sim q(z), x \sim p_t(x|z)$. The equivalence of this fact was proved in Appendix that is based Theorem 2 from Lipman et al. (2022).

The sampling step of flow matching is performed following the steps listed below:

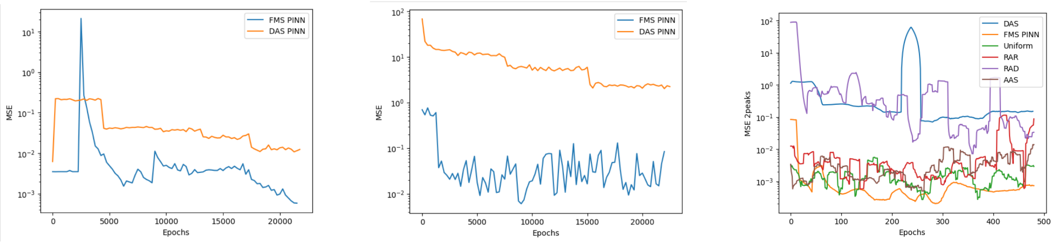
[H] Trained network f_θ, Sample-access to base distribution q, Step-size Δt Sample from target distribution p

$x_1 \leftarrow$ Sample(q) $t = 1, (1 - \Delta t), (1 - 2\Delta t), ..., \Delta t$ $x_{t-\Delta t} \leftarrow x_t + f_\theta(x_t, t)\Delta t$ x_0

Appendix: More technical details on experiments

The mean square error (MSE) comparison between the normalizing flow PINN Tang et al. (2023a) and our method based on matching flow PINN during training for Poisson equation with 9 peaks is illustrated by the MSE metrics calculated for different epochs, as depicted in Figure 8a. The MSE of the flow matching PINN decreases and converges to an order of 10^{-3}.

Moreover, for the five-dimensional Poisson equation with two peaks, the MSE per epoch for our method is lower than that of DAS PINN.Tang et al. (2023a), which did not work in higher dimension.



(a) 9 peaks

(b) 2 peaks 5D

(c) 2 peaks 2D

Figure 8: MSE plot for Poisson equation for 2 peaks problem in 5D and 2D and 9 peaks problem in 2D

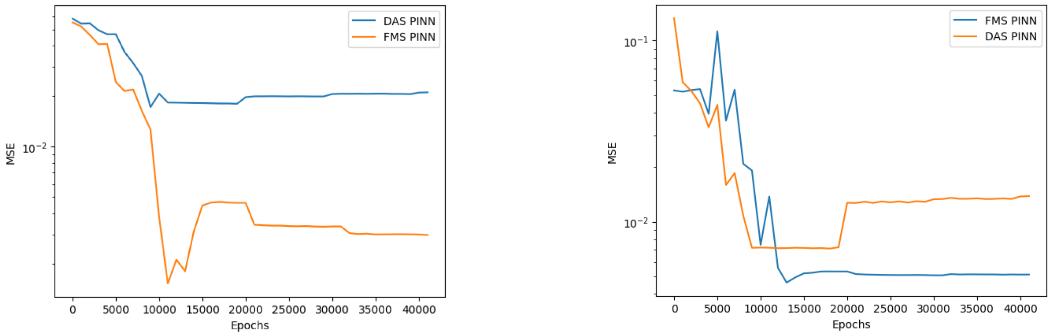
We also see that our method outperforms DAS PINN in terms of sum MSE for u_x and u_y for both diamond and 2 circles setups as depicted by Figure 9:

A.1 2 peaks Problem

Next we consider an equation that has part that corresponds to Laplace operator of the function that resembles Poisson equation combined with the divergence of the vector field of the function $u(\boldsymbol{x})\nabla v(\boldsymbol{x})$:

$$\begin{cases} -\nabla \cdot [u(\boldsymbol{x})\nabla v(\boldsymbol{x})] + \nabla^2 u(\boldsymbol{x}) = s(\boldsymbol{x}) \quad \text{in } \mathbb{D}, \\ u(\boldsymbol{x}) = g(\mathrm{x}) \quad \text{on } \partial\mathbb{D}, \end{cases} \tag{26}$$

where $\boldsymbol{x} = [x_1, x_2]^\mathsf{T}$, $v(\boldsymbol{x}) = (x_1)^2 + (x_2)^2$, and the domain is $\mathbb{D} = [-1, 1]^2$.



(a) MSE per epoch for 2 circles setup of linear elasticity equation

(b) MSE per epoch for diamond setup of linear elasticity equation

Figure 9: Comparison of MSE for diamond and 2 circles setup for FMS PINN and DAS PINN

According to Tang et al. (2023a), the exact solution of 26 reads as follows:

$$\begin{aligned} u(x_1, x_2) = \mathrm{e}^{-1000[(x_1-0.5)^2+(x_2-0.5)^2]} + \\ + \mathrm{e}^{-1000[(x_1+0.5)^2+(x_2+0.5)^2]}, \end{aligned}$$

which has two peaks at the points $(0.5, 0.5)$ and $(-0.5, -0.5)$. From this solution, we compute the source function $s(\boldsymbol{x})$ by applying Laplace operator and combining it with divergence of the vector field of the function $u(\boldsymbol{x})\nabla v(\boldsymbol{x})$. Here, the Dirichlet boundary condition on $\partial\Omega$ is given by the exact solution.

We see that our method succeeds in capturing the solution for the two-peaks problem and achieves the same order of error as compared to the normalizing flow PINN. Nevertheless, the MSE score is 1.5 times lower for the matching flow PINN as compared to the normalizing flow PINN.

To improve the flow matching model, we trained it for 2,000 iterations each time we resampled points and repeated the resampling process every 10,000 iterations, adding 28,000 points during each resampling stage. In the initial stage to prevent overfitting, we trained the initial PINN model for 10,000 epochs. For comparison of our flow matching with the normalizing flow we use a KR-net implementation from Tang et al. (2023a) code implementation in Tensorflow with same number of points, epochs and re-sampling stages.