

FMS PINN: FLOW-MATCHING SAMPLING FOR EFFICIENT SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS WITH SOURCE SINGULARITIES

Anonymous authors

Paper under double-blind review

ABSTRACT

Singularities in the source functions of partial differential equations (PDEs) can pose significant challenges for physics-informed neural networks (PINNs), often leading to numerical instability and necessitating a large number of sampling points thereby increasing the computational time. In this paper, we introduce a novel sampling point selection method to address these challenges. Our approach is based on diffusion models capable of generative sampling from the distribution of PDE residuals. Specifically, we apply the optimal transport coupling flow-matching technique to generate more sampling points in regions where the PDE residuals are higher, enhancing the accuracy and efficiency of the solution. In contrast to existing approaches in the literature, our method avoids explicit modeling of the probability density proportional to residuals, instead using the benefits of flow matching to generate novel and probable samples from more complex distributions, thereby enhancing PINN solutions for problems with singularities. We demonstrate that this method, in certain scenarios, outperforms existing techniques such as normalizing flow-based sampling PINN. Especially, our approach demonstrates effectiveness in improving the solution quality for the linear elasticity equation in the case of material with complex geometry of inclusion. A detailed comparison of the flow matching sampling method with other approaches is also provided.

1 INTRODUCTION

Physics-Informed Neural Networks (PINNs) are used to solve Partial Differential Equations (PDEs) using neural networks. With the rapid development of computing resources and machine learning algorithms, PINNs have become popular for a wide range of realistic simulations Raissi et al. (2019). PINNs utilize automatic differentiation mechanisms to encode PDEs into loss functions, incorporating PDE residuals and boundary conditions. PINNs may be preferred over classical numerical solvers due to their easy coding algorithms for both forward and inverse problems, and their ability to handle high-dimensional problems. Despite the widespread success of PINNs in various PDE-related problems they often struggle with complex PDEs, leading to "failure modes" Wang et al. (2022). Specifically, PINN loss function is very non-convex making it challenging to find a global minimum using conventional optimization algorithms for neural network training, such as Adam. Moreover, according to the F-principle Xu (2020), in PINNs the low frequency features of the solution are captured emerge first, while it will take several training epochs to reproduce high frequency features. In this regard, PINNs may be not efficient for solving PDEs with high-frequency solutions, as shown in Chuprov et al. (2023). For simple PDEs (single-scale, single-mode), conventional PINNs can quickly achieve satisfactory solutions Buzaev et al. (2023). However, for more complex PDEs, conventional PINNs often fall short as the low-frequency global solution deviates from the exact solution. To address these issues, recent years have seen the emergence of efficient implementations of the PINNs method. For instance, loss re-weighting methods McClenny & Braga-Neto (2023) and adaptive sampling strategies Gao et al. (2023) have been developed to find a balance between loss and probability distribution on weight or sampling, enhancing the performance of PINNs in complex scenarios.

The goal of this paper is to design a novel method of point sampling to address these challenges. This approach uses the idea of diffusion models which are capable of generative sampling from the distribution of loss residuals. The optimal transport flow-matching technique is applied to generate more sampling points in regions where the PDE residuals have large values, enhancing the accuracy and efficiency of the solution.

2 PINN OVERVIEW

Consider a domain Ω on which we want to solve a partial differential equation $\mathcal{D}u(\mathbf{x}) = s(\mathbf{x})$, where $s(\mathbf{x})$ is the source function, \mathcal{D} is a differential operator, while \mathbf{x} is a d -dimensional vector from Ω . The domain is bounded by $\partial\Omega$ on which solution is subject to the following boundary condition: $\mathcal{B}u(\mathbf{x}) = g(\mathbf{x})$ for $\mathbf{x} \in \partial\Omega$, where \mathcal{B} is the boundary operator (e.g. Neumann or Dirichlet conditions). Thus, we have

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

We consider a neural network $u_\psi(\mathbf{x})$ that approximates the solution of Equation 1, where ψ represents the parameters of the neural network that will be optimized during the training process. The training of the neural network is based on the minimization of the following function:

$$\min_{\psi} L(\psi) = L_{PDE}(\psi) + L_{BC}(\psi) = \|r(\mathbf{x}_i, \psi)\|_{2,\Omega} + \|\mathcal{B}u(\mathbf{x}) - g(\mathbf{x})\|_{2,\partial\Omega},$$

where

$$L_{PDE,N}(\psi) = \sum_{x_i \in \mathbb{S}_k} (\mathcal{D}u(x_i, \psi) - s(x_i))^2 \tag{2}$$

and

$$L_{BC,N}(\psi) = \sum_{x_i \in \partial\mathbb{S}_k} (\mathcal{B}u(x_i, \psi) - g(x_i))^2. \tag{3}$$

The partial derivatives of PINN with respect to the vector \mathbf{x} can be computed on the basis of automatic differentiation libraries.

3 RELATED WORK

3.1 ADAPTIVE SAMPLING AND ITS RELATION TO GENERATIVE LEARNING

The easiest method for selecting points is a mesh grid (regular grid), that is often used in finite difference schemes. However, in Wu et al. (2023) it was shown that this approach can potentially yield trivial solutions and that the PINN solution derived on a uniform grid is more accurate than that obtained on a mesh grid. In addition, sampling based on pseudo-random series (e.g. Sobol sequences, Latin hypercube etc) can be utilized.

Adaptive sampling methods are based on the principle to select points based on their influence on the loss function. One of the algorithms that pioneered this approach is the so called Residual Aided Refinement (RAR) algorithm Lu et al. (2021). The RAR algorithm aims to improve the distribution of residual points during training by introducing additional points in areas where the PDE loss values are large after a certain number of iterations. An advanced version of RAR is called residual adaptive distribution algorithm: RAD Wu et al. (2023). The PINN training begins with uniformly distributed points. After a few iterations, residual values are evaluated, and new points are added in areas with high residuals. The PINN model is retrained with the updated set of points, and this process is repeated to improve accuracy. This algorithm is similar to the classical importance sampling method, an extension of Monte-Carlo methods.

The algorithm that apply importance sampling idea to estimation of loss and sampling points for PINN is Nabian et al. (2021), where they propose to a special proposal distribution that is based on calculation of residual error at nearest to a point specially selected seed points. For instance, the proposal is a distribution that is a PDE residual loss at nearest seed point divided by sum of all

108 residual values. It means that points with larger values of PDE residual are more likely to be added
109 to a batch. A similar methods are presented in Nabian et al. (2021).

110
111 One of the first methods introduced for sampling for variance reduction purposes was normalizing
112 flows. In image generation such methods as GANs Goodfellow et al. (2020), VAEs and diffusion
113 models can be used to digest data distribution and get a new sample from the population. In Bond-
114 Taylor et al. (2021) it was shown that GANs generation can achieve higher quality for high resolution
115 images than VAEs. However, GANs can be unstable due to mode covering problem while VAEs are
116 able to cover all modes of distribution.

117 Diffusion models Song et al. (2020) do not suffer from mode collapse and can beat GAN models
118 in image quality. They are based on the idea of iterative refinement of an input noise signal until
119 it converges to a specific data distribution, such as an image. Diffusion models are trained by the
120 forward process of incremental noise injection into data image and sampling represents the inverse
121 process of image generation from noise. However, they require a substantial time for training Xiao
122 et al. (2021) as compared to GANs and VAEs.

123 To improve the inference time of diffusion models with minimal depreciation in quality flow match-
124 ing for image generation, a method Lipman et al. (2022) was introduced that is based on the refine-
125 ment of loss function of the continuous normalizing flows Mathieu & Nickel (2020). This model
126 do not implicitly approximate the probability distribution but can produce high quality samples at
127 reasonable time. Flow matching is more stable during training because of its loss function, making
128 it a preferable choice for generation tasks compared to score-based diffusion methods, especially for
129 low-dimensional data such as points of collocation for training PINNs. This is the reason why this
130 model was used as a tool to add samples that move in the direction of large residual regions.

131 3.2 ADAPTIVE SAMPLING STRATEGIES FOR PINNS

132
133 In Tang et al. (2023a), the so called DAS PINN was proposed, in which a normalizing flow was
134 applied for adaptive PDE residual sampling for solution of Poisson equation with singular source
135 peaks, while in Wang et al. (2024a) a similar approach was used for a cavity flow problem. In
136 Tang et al. (2023b) the Wassertein GAN-like model (WGAN) was proposed to solve the Poisson
137 equation with narrow peaks in the source function. It generalizes a sampling approach for any
138 normalized residual distribution p_α but also uses the KR-net architecture same, as in Tang et al.
139 (2023a). The main difference of AAS-PINN Tang et al. (2023b) from DAS-PINN is that AAS PINN
140 learns the distribution of residual with WGAN like a loss function that also applies regularization to
141 the gradient of p_α : ∇p_α , while Tang et al. (2023a) uses KL divergence loss.

142 Due to the fact that KR net model for probability density function (PDF) approximation is invertible,
143 it implicitly models PDF. However let us note that it may face difficulties to approximate more
144 complex probability density functions of points with large residual. Moreover, as it is articulated in
145 Wang et al. (2024b) due to the fact that normalizing flows preserve the topology of the input space
146 through continuous transformations, they face difficulties in representing certain simple classes of
147 function Dupont et al. (2019).

148 This property leads to limited representation capabilities, high computational costs, and training
149 problems in practical implementations Ho et al. (2019). That is why in this paper, we decided to
150 use an architecture different from KR-net and other normalizing flow architectures that represent
151 invertible transformations for our proposed method.

152 3.3 NORMALIZING FLOW PINN

153
154 In Tang et al. (2023a), a model was developed, which integrates the PINN training with sampling
155 from a normalizing flow. Subsequently, the flow model is refined through the minimization of
156 the cross entropy loss between the residual and the flow’s output logarithmic density. The flow
157 model, implemented as KR-net, is constructed with affine coupling layers and the Knothe-Rosenblatt
158 rearrangement. This architecture is uniquely designed to calculate both the forward and inverse
159 probability density.
160

161 The optimization problem for the flow model is articulated as minimizing the Kullback-Leibler (KL)
divergence between the residual function and the probability density generated by the flow. This

162 approach was specifically applied to solve the Poisson equation with a single peak source function
 163 and two peaks.

164 In the following section, we compare the normalizing flow PINN with our approach, providing a
 165 comprehensive analysis of the performance and efficiency of sampling techniques.
 166

167 3.4 FLOW MATCHING PINN

168 3.1 Flow matching

169 Flow matching Lipman et al. (2022) is the generative algorithm that can deal with high complexity
 170 of data. Unlike the normalizing flow method, it does not require the neural network transformation
 171 to be invertible. Instead of implicitly modeling the probability density function $p_1(x)$, the flow
 172 matching model enables sampling from this probability density function $p_1(x)$ by modeling vector
 173 field flow dynamics restoration. As sampling is based on sampling prior of Gaussian distribution
 174 $p_0(x)$ that through flow field dynamics $f_\theta(x, t)$ is transformed into more complex distribution $p_1(x)$,
 175 where time t varies from 0 to 1. In this regard, the sample from unconditional probability density
 176 function can be generated with the solution of the ODE:
 177

$$178 \begin{cases} dX_t = f(X_t, t)dt & , \\ X_0 \sim p_0 & . \end{cases} \quad (4)$$

179 This fact is proved in Lipman et al. (2022) (see Theorem 1 therein) that relates the conditional prob-
 180 ability distribution law dynamics $p_t(x|x_1)$ with vector field $f_\theta(x, t)$ through the continuity equation
 181 :

$$182 \frac{d}{dt} p_t(x) = -\operatorname{div}(p_t(x)f_t(x)) \quad (5)$$

$$183 \int_0^1 \mathbb{E}_{p_t(x)} \|f_\theta(x, t) - f(x, t)\| \quad (6)$$

184 Let us note that $f(x, t)$ is unknown and this functional is not feasible to evaluate. It turns out that
 185 it can be reduced to the conditional flow dynamics $f_\theta(x, t|z)$, where z can be a latent variable that
 186 is sampled from a prior distribution. That is why according to Theorem 2 Lipman et al. (2022) the
 187 intractable integral in 6 can be reduced to the following optimization problem that is tractable to
 188 solve:
 189

$$190 \int_0^1 \mathbb{E}_{q(z)p_t(x|z)} \|f_\theta(x, t) - f(x, t|z)\|^2 dt \rightarrow \min_{\theta} \quad (7)$$

191 The minimum that is a solution to this optimization problem is attained on the real vector field
 192 $f(x, t)$.

193 In order to find the minimum of such optimization problem the gradient can be calculated as an
 194 expectation that is found using the Monte Carlo approximation, namely,
 195

$$196 \nabla_{\theta} \int_0^1 \mathbb{E}_{q(z)p_t(x|z)} \|f_\theta(x, t) - f(x, t|z)\|^2 dt \quad (8)$$

197 where $z \sim q(z)$, $x \sim p_t(x|z)$. The equivalence of this fact was proved in Appendix that is based
 198 Theorem 2 from Lipman et al. (2022). Here we consider special case of optimal transport conditional
 199 vector field: $f(x, t|z) = (1 - (1 - \sigma_{\min})t)x + tz$ for this we can rewrite the flow matching loss as:

$$200 \mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t, q(x_1), p(x_0)} \|f_\theta(x, t) - (x_1 - (1 - \sigma_{\min})x_0)\|^2$$

201 where x_1 belong to data sample. According Theorem 1 in Lipman et al. (2022) this marginal vector
 202 field produces the probability path that is justified through continuity equation 5.
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215

3.4.1 THE SAMPLING ALGORITHM FOR PINN WITH BOOTSTRAP REWEIGHING

In order to tackle complex singularities in the domain (e.g. narrow peaks in the source function), we propose the following adaptive sampling algorithm for PINN. We apply flow matching paradigm as a vector field approximation of residual distribution approximation to sample points according to this vector field to add new points to the collocation point set for PINN training. The idea of the algorithm is based on refinement of PINN though adding points from region where high residual is concentrated. Instead of approximating the probability density function proportional to residual distribution we use this residual distribution in our algorithm to construct a probable sample from this distribution. For this case the residual of $r(x_i, \psi^k)$ is calculated from PINN:

$$r(x_i, \psi^k) = \mathcal{D}u(x_i, \psi^k) - s(x_i). \quad (9)$$

We construct a sub-sample \mathbb{A}_i of size M from previous sample \mathbb{S}_{i-1} proportional to residual values using weighted bootstrap procedure. We train the vector field neural $f_\theta(x, t)$ net on this sub-sample \mathbb{A}_i by minimizing flow matching objective 7 reformulated for tractability as in 8.

This vector field $f_\theta(x, t)$ governs the dynamics of point from prior distribution to the closest point in residual distribution. In this way by application of flow matching sampling we generate points according to this $f_\theta(x, t)$ by ODE dynamics to construct a new sample \mathbb{V}_i :

$$\begin{cases} dX_t = f_\theta(X_t, t)dt & , \\ X_0 \sim p_0 & . \end{cases} \quad (10)$$

This ODE can be solved numerically by using the Euler-Maruyama discretization scheme.

Moreover, for efficient sampling of points we propose the following algorithm of solving PINN with generation point proposal based on flow matching form:

Algorithm 1: FMS PINN: PINN with matching flow

Input : number of points in initial sample N , number of points for training sample for vector field M , number of stages K

Sample N points uniformly from the domain Ω denote this set as $\mathbb{S}_0 = \{\mathbf{x}_i\}_{i=1}^N$;
Train PINN model on sample \mathbb{S}_0 by optimizing empirical loss

$$\min_{\psi} L(\psi, N) = L_{PDE, N}(\psi) + L_{BC, N}(\psi) \quad (11)$$

where $L_{BC, N}(\psi)$ defined in 3 and $L_{PDE, N}(\psi)$ defined in 2. ;

for k from 1 to K **do**

 Calculate $r(\mathbf{x}_i, \psi^k)$ at each point of \mathbb{A}_k , i.e., and get values $\{r(\mathbf{x}_i, \psi^{k-1})\}_{i=1}^N$;

 Based on these weights, perform a weighted bootstrap resampling of points to form the root points A_k for the flow, denoted as $\{\mathbf{x}_i^*\}_{i=1}^N$;

 Train the vector field $f_\theta(x, t)$ on this root points sample A_k by optimizing flow matching objective as in 7;

 Sample new points \mathbb{V}_k according to the vector field $f_\theta(x, t)$ that corresponds to $p_1(x)$ using the Euler method for solving ODE 10;

 Construct a sample for PINN as $\mathbb{S}_{k+1} = \mathbb{V}_k \cup \mathbb{S}_k$;

 Train PINN model $u(x, \psi^k)$ on \mathbb{S}_{k+1} by optimizing loss 11 ;

return $u(x, \psi^k)$;

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

Algorithm 2: Flow-matching Sampling

Input: Trained network f_θ , Sample-access to base distribution q , Step-size Δt

Output: Sample from target distribution p

$x_1 \leftarrow \text{Sample}(q)$;

for $t = 1, (1 - \Delta t), (1 - 2\Delta t), \dots, \Delta t$ **do**

$x_{t-\Delta t} \leftarrow x_t + f_\theta(x_t, t)\Delta t$;

return x_0 ;

The idea of Algorithm 1 can be represented by this diagram shown in Figure.

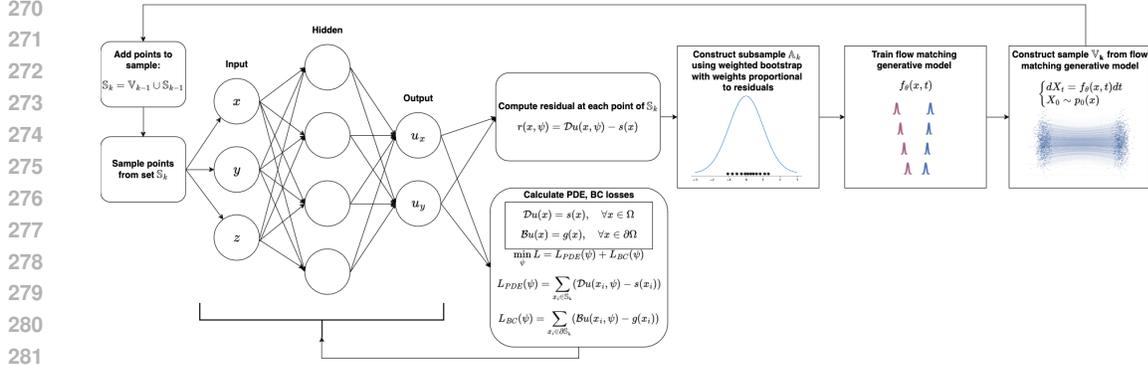


Figure 1: FMS PINN algorithm

4 NUMERICAL RESULTS

4.1 9 PEAKS PROBLEM

285
286
287
288
289
290
291
292
293
294

The issue of singular domains characterized by multi-modality in terms of peaks poses a significant challenge for simple generative models. These models, such as GANs, often struggle with the problem of mode collapse in terms of mode covering. This is why the Poisson equation with a source function consisting of 9 peaks is an important example to demonstrate the effectiveness of our method in enhancing PINN to address the problem of mode covering. By doing so, our method enables the production of accurate solutions that capture the complex structure of the domain.

295
296

The Poisson equation with 9 peaks looks as follows:

$$297 \quad -\Delta u(\mathbf{x}) = s(\mathbf{x}) \quad \text{in } \mathbb{D},$$

$$298 \quad u(\mathbf{x}) = g(\mathbf{x}) \quad \text{on } \partial\mathbb{D}, \quad (12)$$

299
300
301

where $\mathbf{x} = [x_1, x_2]^T$ and $\mathbb{D} = [-1, 1]^2$. Here $s(\mathbf{x})$ has centers in $(x_0^i, y_0^i) = (-0.5, 0.5) + (\frac{\text{mod}(i,3)}{2}, 0) + (0, \frac{i}{3})$, $i = 0, \dots, 8$ and is represented by $s(\mathbf{x}) = \sum_{i=0}^8 s_i(\mathbf{x})$, where

$$302 \quad s_i(\mathbf{x}) = -e^{-1000((x-c_{i,0})^2+(y-c_{i,1})^2)} \left((-2 \cdot 1000(x-c_{i,0})^2 - 2 \cdot 1000) - \right.$$

$$303 \quad \left. -e^{-1000((x-c_{i,0})^2+(y-c_{i,1})^2)} \left((-2 \cdot 1000(y-c_{i,1})^2 - 2 \cdot 1000) \right), i = 0, \dots, 8 \right. \quad (13)$$

306
307
308
309
310
311

We also evaluate the absolute difference profile at the validation dataset for a network approximating the PDE solution. For the network, we used a fully connected network (FCN) with 6 blocks and a layer width of 64. For the flow matching model, we used an optimal transport coupling based on a FCN network. We trained the flow vector field model for 2000 iterations, each time resampling points and repeating the resampling every 5000 iterations, adding 28000 points each time. For

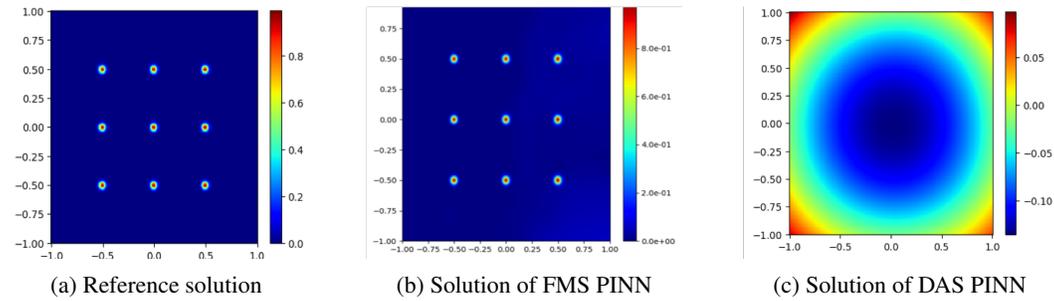


Figure 2: Comparison of solution for 9 peaks problem

the normalizing flow comparison, we used the KR-net implementation from Tang et al. (2023a) in

TensorFlow with the same number of points, epochs, and resampling stages. We added the 9 peaks loss functional and reference in TensorFlow to make the comparison. In Figure 12, we observe that the normalizing flow method fails to capture the main solution compared to the matching flow approach. The flow matching PINN solution accurately depicts all nine peaks, keeping the solution outside the peaks close to zero. Figure 3 show that, in most of the domain, the solution of the flow matching PINN is quite accurate. The mean square difference (MSE) comparison between normalizing flow PINN and matching flow PINN during training for the 9 peak Poisson equation is illustrated by the MSE metrics calculated for different epochs and depicted in Figure 4b. The MSE of the Flow matching PINN decreases and converges to an order of 10^{-3} . Our method samples points from peaks center as depicted by Figure 4a.



Figure 3: Comparison of residual profiles for 9 peaks problem

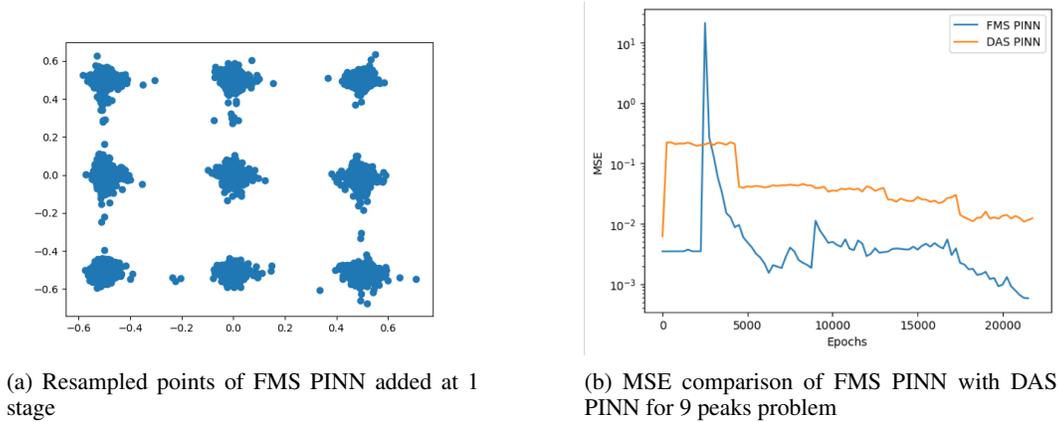


Figure 4: MSE plot and samples from FMS PINN

4.1.1 FIVE-DIMENSIONAL TWO PEAKS PROBLEM

For five dimensional problem the two centers of peaks are placed in $(x_1, x_2, x_3, x_4, x_5) = (0.5, 0.5, 0, 0, 0)$ and at $(x_1, x_2, x_3, x_4, x_5) = (-0.5, -0.5, 0, 0, 0)$. As in Zhang et al. (2024) the reference solution of this problem is:

$$u^*(x, y) = \sum_{i=1}^c \sum_{j=1}^d \exp \left[-K \left((x_j - x_j^i)^2 \right) \right], (x_1, x_2, \dots, x_5) \in \Omega \quad (14)$$

where $K = 100$. In order to make an inference of the model and compute numerical errors efficiently we follow the methodology as in Zhang et al. (2024) where the two-stage sampling strategy for inference where proposed, where firstly 100k points are sampled uniformly across the domain. Then these points are combined with 15k points drawn from Gaussian distributions, whose mean and covariance are determined by each part of the solution led by one of the centers. These points

Table 1: Comparison of linear elasticity equation PINN with Normalizing flow PINN in terms of MSE

Method	2 peaks problem 16	2 peaks in 5D	9 peaks problem
FMS PINN	7.7e-5	6.1e-3	4.2e-4
DAS PINN	5.2e-4	2.3	1e-1

with 10k points on the boundary are subsequently used to compute numerical errors for this five-dimensional two peaks problem.

For training purpose at initial step of training we draw 100k points from uniform distribution and 60k points from Gaussian centers. For optimizer algorithm we used Adam with learning rate 0.001.

We trained FMS PINN with sampling 40000 additional points at every resampling stage from the vector field trained via optimal transport flow matching objective on weighted bootstrap sub-sample from PINN training set S_{k-1} and its residual distribution as weights.

We compare our algorithm with DAS PINN approach on the same number of training points equal to 100k and 60k points from center and for comparison use normalizing flow architecture of KR-net. We see that our method successfully captures all features of the solution, while method based on normalizing flow DAS PINN fails to produce the solution for same number of points and resampling stages.

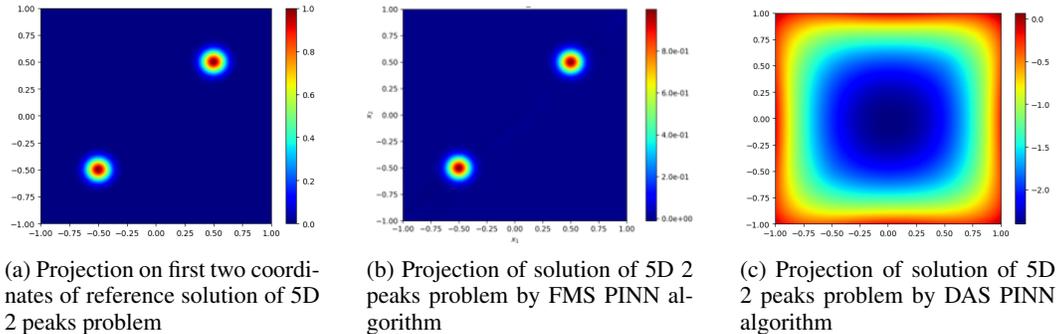


Figure 5: 5D 2 peaks problem

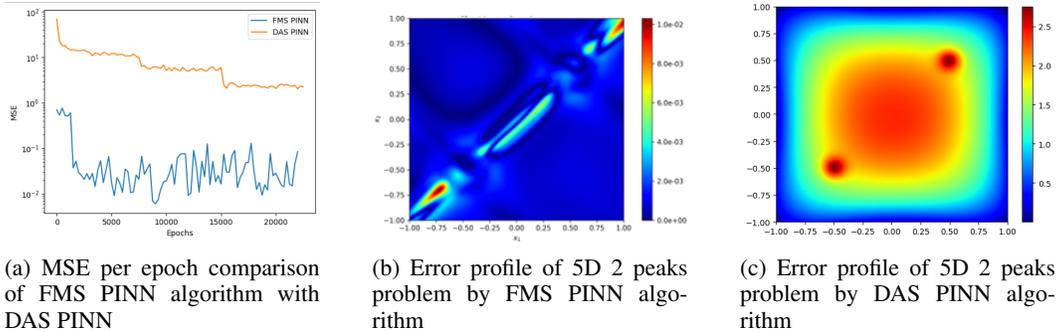


Figure 6: MSE comparison of FMS PINN algorithm with DAS PINN and comparison with reference

Finally, Table 1 summarizes comparison results for the normalizing flow PINN and the flow matching PINN for Poisson problems with peaks in source function, revealing comparable efficiency of the proposed method comparing to the normalizing flow.

4.2 LINEAR ELASTICITY EQUATION

In this section we consider solving a special instance of the mechanical equilibrium equation for a rectangular plate with a unique geometric inclusion made of a second material that we call linear elasticity equation. The primary equation that governs the mechanism of stress under deformation is

$$\nabla \cdot \sigma = 0,$$

where σ is the stress tensor - the 2-nd order tensor describing the internal pressure state of the object. This equation is then can be represented:

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

where E and ν are the Young modulus and Poisson ratio-constants, describing the material properties, while u_x and u_y represent horizontal and vertical displacement respectively.

The detailed derivation of this equation can be found in the subsection A.2.

where $C = \frac{E}{(1+\nu)(1-2\nu)}$ - constant.

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

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

We consider a specific kind of plate, that consists of one base material and second material in complex geometry inclusion, that is characterised with different Young modulus (= material property of stiffness) E . The geometric configurations are diamond and 2 circles.

Structure of the neural network is represented by 5 separate fully connected neural nets.

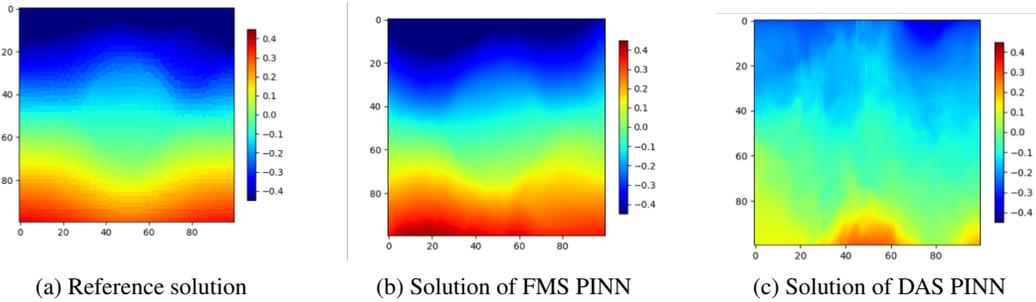


Figure 7: Comparison of solution for 2 circles u_x problem

For 30000 epochs of training we see that flow matching PINN outperforms DAS PINN. The results of MSE comparison for the flow matching PINN with the normalizing flow PINN is shown in Table 2. For two circles problem the flow matching method helps to improve quality, while for the diamond configuration it provides the solution of the same quality as the normalizing flow PINN.

Results of our method compared to the reference solution and DAS PINN for 2 circles case is illustrated in Figure 7 and Figure 8. PINN neural net architecture for our methods consists of 5 separate neural networks that have 5 fully connected layers with 40 neurons in each layer. As an optimizer, we use Adam with the scheduler ReduceLROnPlateau. As it is shown in Figure 9, our method captured all main patterns of the reference solution as we see our algorithm FMS PINN outperforms DAS PINN.

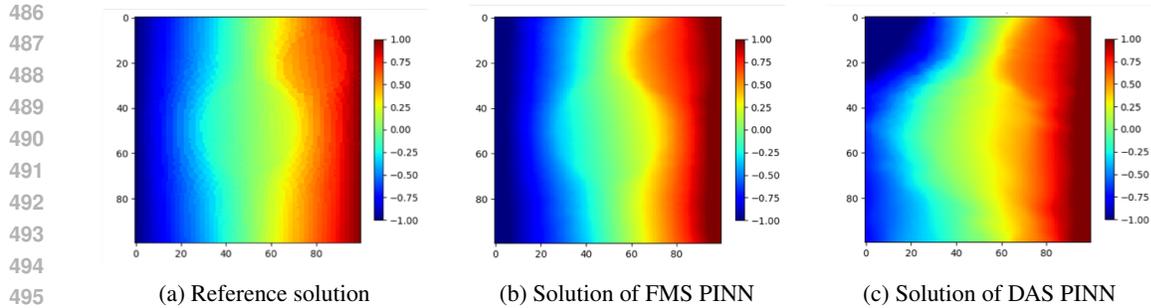
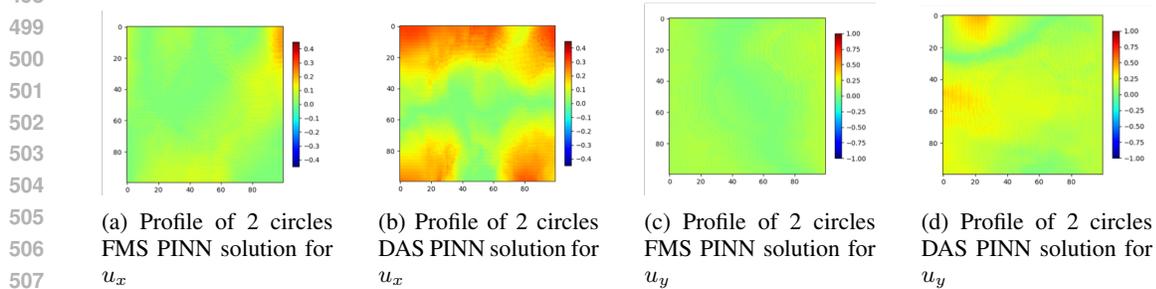
Figure 8: Comparison of solution for 2 circles u_y problem

Figure 9: Comparison of error profiles for DAS PINN and FMS PINN for 2 circles problem

Table 2: Comparison of Elasticity PINN with Normalizing flow PINN in terms of MSE

Method	2 circles u_x	2 circles u_y	diamond u_x	diamonds u_y
FMS PINN	1.5e-3	7.9e-3	4.6e-3	9.2e-3
DAS PINN	1.7e-2	1.2e-2	7.1e-3	8.6e-3

518 5 CONCLUSION

519 In this paper a novel approach referred to as flow-matching sampling is proposed. It allows to select
520 points for PINNs training, at which the evaluation of the PDE residual is performed. The idea of the
521 method is based on the generative matching flows and adaptive sampling.

522 The numerical experiments show that our approach helps to solve singular problems and enhance
523 the solution. We have examined an efficiency of the proposed method for the Poisson equation and
524 linear elasticity equation system. It has been shown that the proposed method in several cases allow
525 to achieve more accurate solution than the normalization flow approach. The latter can be considered
526 as the closest competitor of the flow-matching method. It has been shown that the flow-matching
527 method is efficient in the case of singularities in the solution. In our future work we will examine
528 this method on larger number of epochs.

532 6 REPRODUCIBILITY STATEMENT

533 All of our experimental results are fully reproducible, and we have documented all settings and param-
534 eters used in our experiments. Upon request from the reviewers, we are prepared to provide the
535 code and detailed instructions to help to replicate our findings. For the comparison with the DAS
536 PINN method, we utilized the publicly available repository at <https://github.com/MJfadeaway/DAS>.
537 By employing this repository, we ensured that our comparative analysis was conducted under consist-
538 ent conditions, thereby guaranteeing a fair and accurate assessment between our proposed approach
539 and the DAS PINN algorithm.

REFERENCES

- 540
541
542 Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling:
543 A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models.
544 *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7327–7347, 2021.
- 545 Fedor Buzaev, Jiexing Gao, Ivan Chuprov, and Evgeniy Kazakov. Hybrid acceleration techniques
546 for the physics-informed neural networks: a comparative analysis. *Machine Learning*, 113(6):
547 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>.
- 548
549
550 I. Chuprov, Jiexing Gao, D. Efremenko, E. Kazakov, F. Buzaev, and V. Zemlyakov. Optimiza-
551 tion of physics-informed neural networks for solving the nolinear schrödinger equation. *Dok-*
552 *lady Mathematics*, 108(S2):S186–S195, December 2023. ISSN 1531-8362. doi: 10.1134/
553 s1064562423701120. URL <http://dx.doi.org/10.1134/S1064562423701120>.
- 554 Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In H. Wal-
555 lach, H. Larochelle, A. Beygelzimer, F. d'Alch'e-Buc, E. Fox, and R. Garnett (eds.),
556 *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.,
557 2019. URL [https://proceedings.neurips.cc/paper_files/paper/2019/](https://proceedings.neurips.cc/paper_files/paper/2019/file/21be9a4bd4f81549a9d1d241981cec3c-Paper.pdf)
558 [file/21be9a4bd4f81549a9d1d241981cec3c-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/21be9a4bd4f81549a9d1d241981cec3c-Paper.pdf).
- 559
560 Zhiwei Gao, Liang Yan, and Tao Zhou. Failure-informed adaptive sampling for pinns. *SIAM*
561 *Journal on Scientific Computing*, 45(4):A1971–A1994, July 2023. ISSN 1095-7197. doi:
562 10.1137/22m1527763.
- 563 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,
564 Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the*
565 *ACM*, 63(11):139–144, 2020.
- 566
567 Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-
568 based generative models with variational dequantization and architecture design. In *International*
569 *conference on machine learning*, pp. 2722–2730. PMLR, 2019.
- 570 Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching
571 for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- 572
573 Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library
574 for solving differential equations. *SIAM review*, 63(1):208–228, 2021.
- 575
576 Emile Mathieu and Maximilian Nickel. Riemannian continuous normalizing flows. *Advances in*
577 *Neural Information Processing Systems*, 33:2503–2515, 2020.
- 578
579 Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks.
580 *Journal of Computational Physics*, 474:111722, February 2023. ISSN 0021-9991. doi: 10.1016/
581 j.jcp.2022.111722.
- 582
583 Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. Efficient training of physics-
584 informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure*
585 *Engineering*, 36(8):962–977, 2021.
- 586
587 Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A
588 deep learning framework for solving forward and inverse problems involving nonlinear partial
589 differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- 590
591 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben
592 Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint*
593 *arXiv:2011.13456*, 2020.
- 594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

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

596
597 Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent
598 kernel perspective. *Journal of Computational Physics*, 449:110768, January 2022. ISSN 0021-
599 9991. doi: 10.1016/j.jcp.2021.110768.

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

602
603 Yanjie Wang, Feng Liu, Faguo Wu, and Xiao Zhang. Navigating pinns via maximum residual-based
604 continuous distribution. 2024b.

605 Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-
606 adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer
607 Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

608 Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with
609 denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

610
611 Zhi-Qin John Xu. Frequency principle: Fourier analysis sheds light on deep neural networks. *Com-
612 munications in Computational Physics*, 28(5):1746–1767, June 2020. ISSN 1991-7120. doi: 10.
613 4208/cicp.oa-2020-0085. URL <http://dx.doi.org/10.4208/cicp.OA-2020-0085>.

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

617 A APPENDIX

618
619 In this section, we apply the flow matching PINN and normalizing flow PINN to several PDEs. We
620 begin our analysis with the Poisson equation that contains a singular source with peaks, representing
621 the complex structure of the domain. As uniform sampling may not adequately cover the peak region
622 within the domain, it can result in significant deviations of the PINN solution from the reference
623 solution
624

625 A.1 2 PEAKS PROBLEM

626
627 Next we consider an equation that has part that correspond to Laplace operator of the function
628 that resembles Poisson equation combined with the divergence of the vector field of the function
629 $u(\mathbf{x})\nabla v(\mathbf{x})$:
630

$$631 \begin{cases} -\nabla \cdot [u(\mathbf{x})\nabla v(\mathbf{x})] + \nabla^2 u(\mathbf{x}) = s(\mathbf{x}) & \text{in } \mathbb{D}, \\ u(\mathbf{x}) = g(\mathbf{x}) & \text{on } \partial\mathbb{D}, \\ s(\mathbf{x}) = s_1(\mathbf{x}) + s_2(\mathbf{x}), \end{cases} \quad (16)$$

$$632 \quad s_1(\mathbf{x}) = \left(e^{-1000(x_1-0.5)^2+(x_2-0.5)^2} \right) (4000(1000(x_1)^2-1000x_1+1000(x_2)^2-1000x_2+499)+$$

$$633 \quad \quad \quad + 4(1000(x_1)^2 - 500x_1 + 1000(x_2)^2 - 500x_2 - 1))$$

$$634 \quad s_2(\mathbf{x}) = e^{500(2x_1^2+2x_1+2x_2^2+2x_2+1)} (4000(1000x_1^2 + 1000x_1 + 1000x_2^2 + 1000x_2 + 499) +$$

$$635 \quad \quad \quad + 4(1000x_1^2 + 500x_1 + 1000x_2^2 + 500x_2 - 1))$$

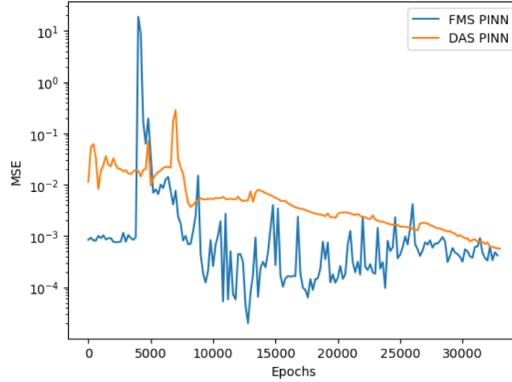
636
637
638
639
640
641 where $\mathbf{x} = [x_1, x_2]^T$, $v(\mathbf{x}) = (x_1)^2 + (x_2)^2$, and the domain is $\mathbb{D} = [-1, 1]^2$. According to Tang
642 et al. (2023a), the exact solution of 16 reads as follows:
643
644

$$645 \quad u(x_1, x_2) = e^{-1000[(x_1-0.5)^2+(x_2-0.5)^2]} + e^{-1000[(x_1+0.5)^2+(x_2+0.5)^2]}, \quad (17)$$

648 which has two peaks at the points $(0.5, 0.5)$ and $(-0.5, -0.5)$. Here, the Dirichlet boundary condi-
 649 tion on $\partial\Omega$ is given by the exact solution.

650
 651 We see that our method succeeds in capturing the solution for the two-peaks problem and achieves
 652 the same order of error as compared to the normalizing flow PINN. Nevertheless, the MSE score is
 653 1.5 times lower for the matching flow PINN as compared to the normalizing flow PINN.

654 To improve the flow matching model, we trained it for 2,000 iterations each time we resampled
 655 points and repeated the resampling process every 10,000 iterations, adding 28,000 points during
 656 each resampling stage. In the initial stage to prevent overfitting, we trained the initial PINN model
 657 for 10,000 epochs. For comparison of our flow matching with the normalizing flow we use a KR-
 658 net implementation from Tang et al. (2023a) code implementation in Tensorflow with same number
 659 of points, epochs and re-sampling stages. Figure 10 indicates that after 6000 epochs the matching
 660 flow algorithm achieves better quality than normalizing flow PINN. For this problems flow matching
 661 PINN after 25000 is better than solution of the normalizing flow PINN, while Figure 11 shows that
 662 both normalizing flow PINN and flow matching PINN approximate the solution in a descend way.



663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 Figure 10: MSE comparison for 2 peaks problem



679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 (a) Residual profile of FMS PINN

(b) Residual profile of DAS PINN

690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 Figure 11: Comparison of residual profiles for 2 peaks problem

700
 701
 A.2 LINEAR ELASTICITY EQUATION: DETAILS

Here is more detailed derivation of linear elasticity equation. The primary equation that governs the mechanism of stress under deformation is

$$\nabla \cdot \sigma = 0,$$

where σ is the stress tensor - the 2-nd order tensor describing the internal pressure state of the object. In matrix notation, equation for the 2D case takes the form:

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix} = 0, \tag{18}$$

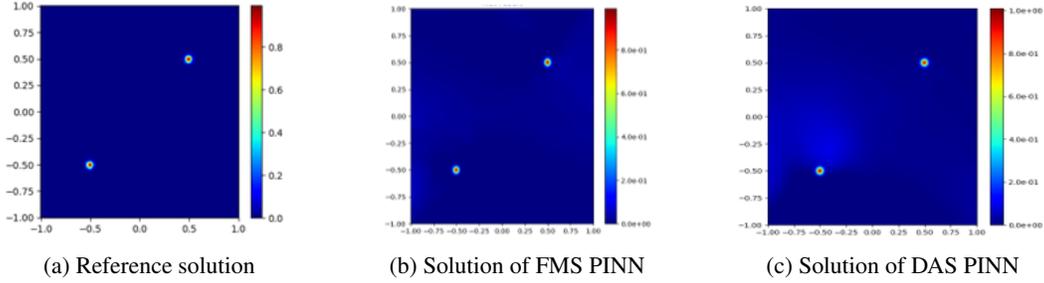


Figure 12: Comparison of solution for 2 peaks problem

We can express the stress tensor σ in terms of the 4th order elasticity tensor C and the 2nd order strain tensor ϵ :

$$\sigma_{ij} = C_{ijkl} \cdot \epsilon_{lk}$$

For the case of linear isotropic materials in 2D, this reduces to 2-dimensional Hooke's law for plane strain:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{(1+\nu) \cdot (1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix} \cdot \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix}, \quad (19)$$

where E and ν are the Young modulus and Poisson ratio constants, describing the material properties, E and ν are individual for each material and can vary a lot from one material to another

The strain tensor C describes the deformation of the solid body in each elementary volume. The ϵ_{xx} and ϵ_{yy} components are describing the relative elongation of the elementary volume in x and y directions respectively, and the ϵ_{xy} , component describes the shift deformations.

In classical case the equation is solved with respect to displacements, and the strain tensor can be expressed:

$$\begin{aligned} \epsilon_{xx} &= \frac{\partial u_x}{\partial x}, \\ \epsilon_{yy} &= \frac{\partial u_y}{\partial y}, \\ \epsilon_{xy} &= 0.5 \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right). \end{aligned} \quad (20)$$

where u_x and u_y are the displacements of the points of solid body in directions x and y respectively

By substituting this idea 18 to initial equation 20 we can finally obtain:

$$\begin{aligned} C(1-\nu) \frac{\partial^2 u_x}{\partial x^2} + C\nu \frac{\partial^2 u_y}{\partial x \partial y} + \frac{1}{2}C(1-2\nu) \left(\frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_y}{\partial x \partial y} \right) &= 0 \quad (\text{x-axis}) \\ \frac{1}{2}C(1-2\nu) \left(\frac{\partial^2 u_x}{\partial x \partial y} + \frac{\partial^2 u_y}{\partial x^2} \right) + C\nu \frac{\partial^2 u_x}{\partial x \partial y} + C(1-\nu) \frac{\partial^2 u_y}{\partial y^2} &= 0 \quad (\text{y-axis}), \end{aligned} \quad (21)$$

where $C = \frac{E}{(1+2\nu)(1-2\nu)}$ - constant

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

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

We consider a specific kind of plate, that consists of one base material and second material in complex geometry inclusion, that is characterised with different Young modulus E . The geometric configurations are diamond and 2 circles.

Here is the solution for diamond configuration for FMS PINN compared with DAS PINN:

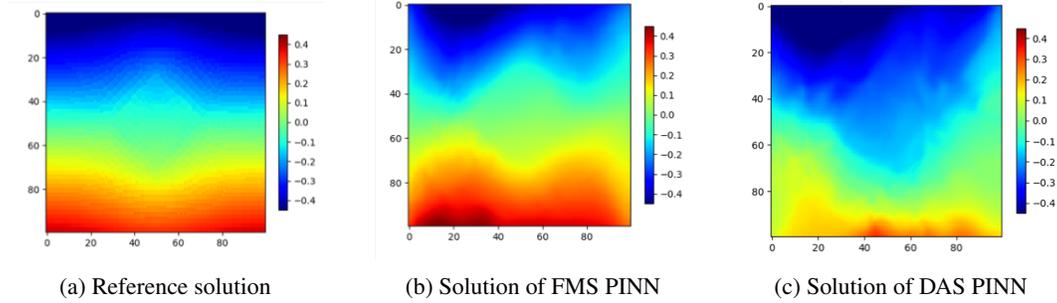


Figure 13: Comparison of solution for diamond u_x problem

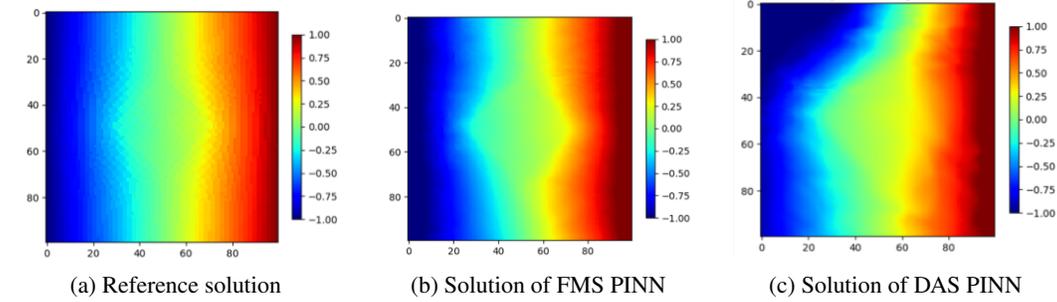


Figure 14: Comparison of solution for diamond u_y problem

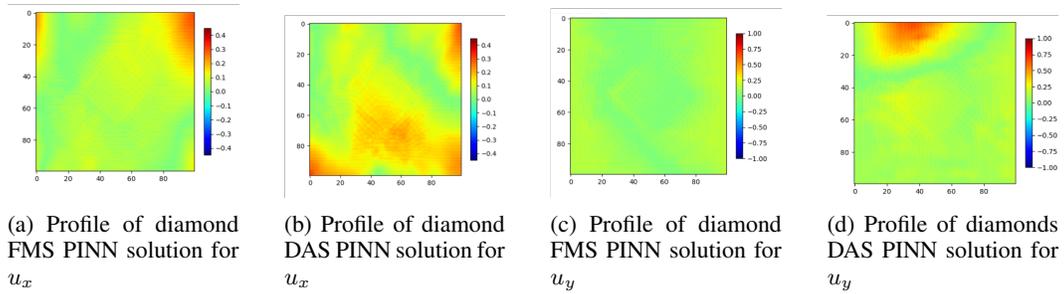


Figure 15: Comparison of error profiles for DAS PINN and FMS PINN for diamond problem

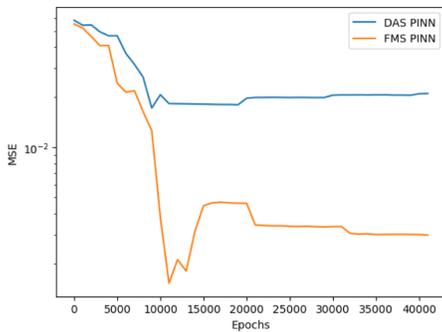
Here is comparison in terms of MAE for 2 configurations: 2 circles and diamond.

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 16:

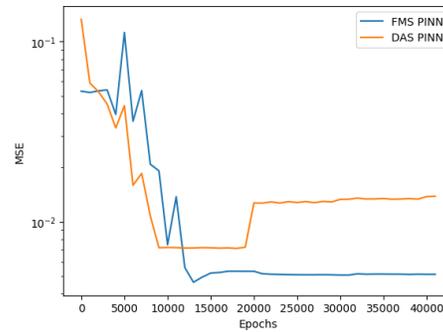
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

Table 3: Comparison of Elasticity PINN with Normalizing flow PINN in terms of MAE

Method	2 circles u_x	2 circles u_y	diamond u_x	diamonds u_y
FMS PINN	0.04184	0.07817	0.073	0.097
DAS PINN	0.12	0.17	0.10	0.14



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



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

Figure 16: Comparison of MSE for diamond and 2 circles setup for FMS PINN and DAS PINN