GPINN: Physics-Informed Neural Network with Graph Embedding

1st Yuyang Miao*
Department of Electrical and Electronic Engineering
Imperial College London
London, United Kingdom
yuyang.miao20@imperial.ac.uk

2th Haolin Li
Department of Aeronautics
Imperial College London
London, United Kingdom
haolin.li20@imperial.ac.uk

3th Danilo Mandic
Department of Electrical and Electronic Engineering
Imperial College London
London, United Kingdom
d.mandic@imperial.ac.uk

***Abstract*—Solving partial differential equations (PDEs) is of great importance in numerous fields including physics, engineering, finance, and scientific computing. Physics-Informed Neural Networks (PINNs) have gained interest due to their ability to solve PDEs in the strong form, which differs from the traditional methods like the Finite Element Method (FEM) which exhibits the weak form. However, the PINN methods normally operate in the Euclidean space only and lack spatial contextual awareness. This deficiency leads to distance inequality, whereby the Euclidean distance might not align with the actual physical distance. Such misalignment may lead the model to produce high errors with no physical meanings. In this study, we propose the Physics-Informed Neural Network with Graph Embedding (GPINN), which utilises the eigenvectors of graph Laplacian to transform the input space from a pure Euclidean space to a joint Euclidean and topological (graph-based) space and make the model spatial-context aware. Two case studies are conducted to model heat propagation and linear elasticity and show the enhancement from PINN to GPINN.**

***Index Terms*—Physics-Informed Neural Network, Graph Theory**

I. Introduction

The Physics-Informed Neural Network (PINN) has demonstrated potential in solving partial differential equations (PDEs) by generating solutions embodied within the framework of neural networks [1]. The PINN paradigm uniquely leverages the strong form of PDEs yielding solution expressions that are continuous and differentiable, which differs from conventional numerical approaches, such as the Finite Element Method (FEM), that utilise the weak form of PDEs to obtain discrete solutions. The robustness of PINNs has been underscored by their capacity to address inverse problems. Such problems often prove unassailable for FEM due to the absence of complete boundary conditions in certain complex scenarios [2].

A remarkable property of the Physics-Informed Neural Network approach is the ability to integrate physical principles into the equation-solving process, which is achieved by calculating the residual of a partial differential equation (PDE), given that the neural network (NN) solution is differentiable throughout the entire effective domain. This residual is subsequently included in the loss function during NN training [1]. Incorporating this physical information substantially enhances the robustness of the solution, mitigating the overfitting issue in forward problems and enabling the inference of global solutions from sparse local information in inverse problems [2], [3].

However, in the traditional PINN model, the training of NN is not informed with the spatial contextual information. The reason is that the input space of PINN, the traditional Euclidean space, does not align consistently with the physical space (real-world environment). This deficiency brings the distance inequality: the Euclidean distance between two points may not be valid due to the bounded nature of the domain. This discrepancy poses significant challenges when employing PINNs for problems associated with complex geometries or highly discontinuous solution fields, e.g. crack or fracture problems [4]–[6]. Therefore, enriching the PINN model with spatial contextual information and aligning the input space more closely with the physical space could significantly enhance the performance of PINN.

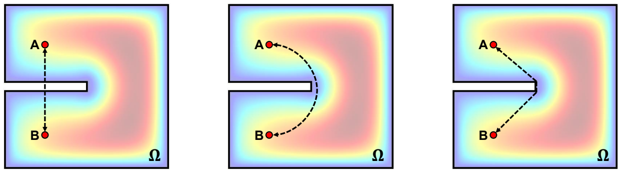
To address these challenges, we propose a Physics-Informed Neural Network with Graph Embedding (GPINN) method. This approach incorporates an additional dimension to address the problems observed in higher dimensional joint Euclidean and topological (graph based) space. An extra dimension is informed by the graph theory and it quantitatively uncovers the influential relationships between different parts of the domain which comprises the spatial contextual information. The rest of this paper is organized as follows: Section II introduces GPINN, covers the basics of graph theory, and explains the method of determining the extra dimension using the Fiedler vector. Section III presents two case studies where we apply the developed GPINN to a heat propagation problem and a cracking modelling method in solid mechanics. Finally, Section IV offers some concluding remarks.

II. Methodology

This section presents the methods and fundamental knowledge underpinning our work. We first introduce the proposed Graph-Embedded Physics-Informed Neural Network

*Corresponding Author

This paper has been accepted by IJCNN 2024



(a) (b) (c)

Fig. 1: Distances of heat propagation (a) in the input space, (b) through a possible propagation path in physics and (c) in the shortest propagation path in physics.

(GPINN), followed by an introduction to the required graph theory.

A. PINN with Graph Embedding

Partial differential equations (PDE) generally take the form [1]

$$\boldsymbol{u}(\boldsymbol{x},t)+\mathcal{N}(\boldsymbol{u}(\boldsymbol{x},t))=0, \boldsymbol{x}\in\Omega, t\in[0,T] \quad (1)$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ represents a solution field dependent on the spatial ($\boldsymbol{x}$) and time (t) coordinates. PINN directly solves the strong form presented by Eq.1 by expressing the solution through a neural network mapping that relates the inputs $x\in\Omega\subset\mathbb{R}^d$ and $t\in[0,T]$ to the output $\boldsymbol{u}(\boldsymbol{x},t)\in\mathbb{R}$ [1] in the form

$$\boldsymbol{u}_{NN}(\boldsymbol{x},t):\Omega\rightarrow\mathbb{R} \quad (2)$$

Solving a partial differential equation is now converted into an optimisation problem in the context of PINN, where the neural network is trained on the loss function of the form

$$\mathcal{L}=\omega_1\mathcal{L}_{\text{PDE}}+\omega_2\mathcal{L}_{\text{Data}}+\omega_3\mathcal{L}_{\text{IC}}+\omega_4\mathcal{L}_{\text{BC}} \quad (3)$$

where $\mathcal{L}_{\text{PDE}}$, $\mathcal{L}_{\text{Data}}$, $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{BC}}$ denote respectively the residual of PDE i.e. $\mathcal{L}_{\text{PDE}}=u_{NN}(x,t)+\mathcal{N}(u_{NN}(x,t))$; the loss of the sampling points; the loss of the initial condition and the loss of the boundary conditions, and ωs are their scaling factors.

The data-driven solution offered by PINN is continuous and differentiable across the domain of $\mathbb{R}^d$ and the time interval $[0,T]$. This type of solution has distinct advantages over numerical solutions with condensed information. However, such a continuous input space also restricts its use in discontinuous fields or fields with non-differentiable segments due to its lack of spatial contextual information. Such a deficiency will introduce the distance inequality, that is, the Euclidean distance between two points in the input space does not always correspond to their physical distance in Ω. An example of this physical rules unawareness is illustrated in Fig.1 in a heat propagation scenario in a two-dimensional 'house' Ω. The shortest path between points A and B in the PINN's input Euclidean space is depicted in Fig.1(a). However, in a physical context, heat actually propagates along other paths and the shortest Euclidean path is impossible due to wall blockage. Fig.1(b) and (c) give one possible path and one shortest path in the domain Ω.

To address this issue, this work proposes a method that incorporates graph embedding into the Physics-Informed Neural Networks to align the input space more closely with the physical spaces. As demonstrated by [7], a topological space defined by graph theory can more accurately capture spatial contextual characteristics compared to a Euclidean space. Consequently, the solution of Eq.1 is modified as follows

$$\boldsymbol{u}_{NN}(\boldsymbol{x},t,\boldsymbol{z}):\Omega\rightarrow\mathbb{R} \quad (4)$$

where the introduced extra dimension is denoted by $\boldsymbol{z}$ and $\boldsymbol{z}\in\mathbb{R}$. $\boldsymbol{z}$ is also a function of $\boldsymbol{x}$, i.e. $\boldsymbol{z}=f(\boldsymbol{x})$. The representation of $\boldsymbol{z}=f(\boldsymbol{x})$ can be done by several ways, including but not limited to meshfree approximation methods (e.g. the Radial Basis Function method), regression methods and even the neural network itself.

Remark. GPINN transforms the operating space from an Euclidean space to a joint Euclidean and topological (graph-based) space through an extra dimension, which ensures a closer alignment between the problem domain and the physical space of the system under consideration.

Remark. The extra dimension is uniquely defined by the specific geometry under consideration, indicating that its identification is topology-specific and independent of the initial or boundary conditions.

The subsequent subsections provide detailed information on how to determine the extra dimension for a prescribed domain. The architecture of the Physics-Informed Neural Network with Graph Embedding (GPINN) is depicted in Fig.2.

B. Graph Theory

A graph, denoted by $G=(V,E)$, is formed by a set of vertices V interconnected by a set of edges E. The adjacency matrix A denotes the connectivity details of a graph, with $A_{i,j}\neq 0$ indicating an edge between the i^{th} and j^{th} vertices.

Graphs can be categorised into two main types depending on whether their edges bear directionality: undirected and directed graphs. In the case of undirected graphs, the adjacency matrix is symmetric, thereby satisfying $A_{i,j}=A_{j,i}$.

This study exclusively considers undirected graphs due to the fact that the physical interactions are mutual. The degree of a vertex, a metric representing the number of nodes it connects to, can be found in the degree matrix D, where $D_{i,i}$ corresponds to the degree of vertex i and all off-diagonal elements are equal to zero. The graph Laplacian matrix is subsequently defined as $L=D-A$.

C. Complex Geometries defined in Topological Space

A component's mesh can be conceptualized as a graph $G_m(V,E)$, where the set of vertices V signifies the element points and E represents the edges. It is worth noting that a graph merely encodes the connectivity among nodes without preserving their exact positions. Intriguing insights can be gleaned when the mesh is treated strictly as a graph.

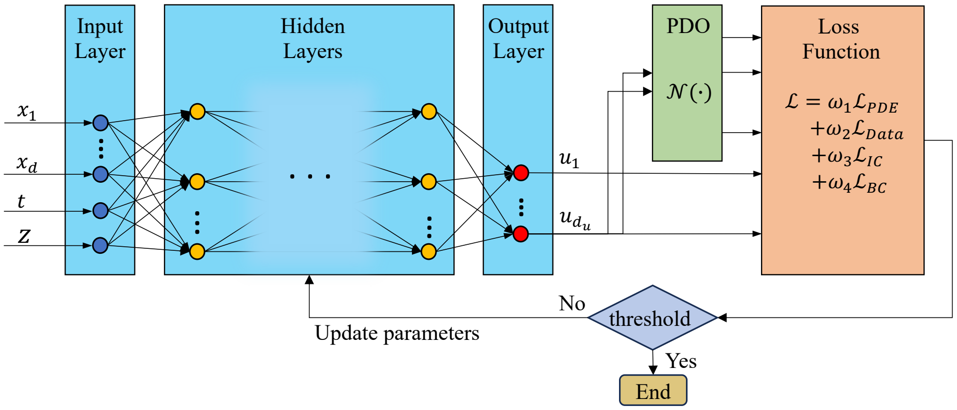


Fig. 2: Schematic of a Physics-Informed Neural Network with Graph Embedding (GPINN). The neural network constructs the relation between the input spatial coordinate $\boldsymbol{x}$, time t and the extra dimension $\boldsymbol{z}$ and the output $\boldsymbol{u}$.

When the mesh is interpreted as a graph, it becomes apparent that many components with complex geometry resemble dumbbell-shaped graphs featuring dense clusters interconnected by tubes. An instance of such a connection is shown in Fig.3 for the structure presented in Fig.1, whereby Fig.3a is the structure in Fig.1 transformed to a mesh. Fig.3b illustrates the corresponding graph in the eigenspace, where the axes are normalized first three eigenvectors of the graph Laplacian. For convenience, in both figures, nodes located at crucial positions are colour-coded. In this way, the spatial contextual information can be reconceived within the graph domain as node labelling, whereby the labels should:

- Conform to the pattern of the overall shape of a component (comprises spatial contextual information).
- Their value differences should compensate for the distance inequality.
- Maintain smoothness between clusters to simulate a continuous field on a graph.

The eigenvector of the graph Laplacian associated with the smallest non-zero eigenvalue (Fiedler vector) potentially fulfils the above stipulations. The subsequent chapter will establish why the Fiedler vector meets these criteria.

D. Fiedler Vector: Relation with Heat Equation

The relationship between the Fiedler vector and the discrete heat transfer equation was addressed in [7]. Consider a discrete heat transfer equation on a graph, given by

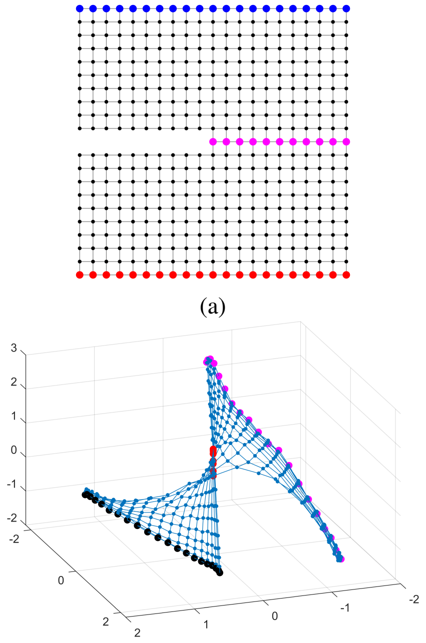
$$\frac{d\mathbf{f}}{dt} = \mathbf{L}\mathbf{f}, \mathbf{f}(0) = \mathbf{f}_0 \tag{5}$$

where $\mathbf{L}$ denotes the graph Laplacian. The solution to Eq.5 is

$$\mathbf{f}(t) = \sum_{i=1}^{n} (\mathbf{f}_0, \mathbf{u}_i)\, e^{-\lambda_i t}\mathbf{u}_i \tag{6}$$

where e_i denotes the orthonormal basis of eigenvectors of L. Since $\lambda_1 = 0$ and $u_1 = [1, 1, ..., 1, 1]/\sqrt{N}$, the solution Eq.6 can be rewritten as

$$\mathbf{f}(t) \approx (\mathbf{f}_0, \mathbf{u}_1)\, \mathbf{u}_1 + (\mathbf{f}_0, \mathbf{u}_2)\, e^{-\lambda_2 t}\mathbf{u}_2 + R \tag{7}$$



(a)

(b)

Fig. 3: (a) Graph of a complex domain, (b) The graph layout in the eigenspace

where the first term is the final state average signal and the remainder R goes to zero faster than the term $(\mathbf{f}_0, \mathbf{u}_2)\, e^{-\lambda_2 t}\mathbf{u}_2$. Therefore, the second eigenvector $\mathbf{u_2}$ with a constant bias could model the transient state of the heat transfer.

Conjecture 1. *Rauch's hot spots conjecture:*
Let $\mathcal{M}$ be an open connected bounded subset. Let $f(\sigma, p)$ be the solution of heat equation, then

$$\frac{\partial f}{\partial \sigma} = \Delta f \tag{8}$$

with the initial condition $f(0, p) = g(p)$ and the Neumann boundary condition $\frac{\partial f}{\partial n}(\sigma, p) = 0$ on the boundary $\partial\mathcal{M}$. For most initial conditions, if p_{hot} is a point at which the function

$f(\cdot, p)$ attains its maximum (hot spot), then the distance from p_{hot} to $\partial\mathcal{M}$ tends to zero as $\sigma \to \infty$ [8]. The minimum point (cold spot) follows a similar rule.

Since the second eigenvector models the heat transient stage, we can employ the conjecture [7]:

Conjecture 2. *Given a graph $G = (V, E)$, If $v^*, w^* \in V$, then*

$$|u_2(v) - u_2(w)| \leq |u_2(v^*) - u_2(w^*)| \quad \forall (v, w) \in V^2$$
$$d(v, w) \leq d(v^*, w^*) \quad \forall (v, w) \in V^2$$

where $d(v, w)$ is the geodesic between (v) and (w), in other words, the level of the extremeness of the value of the Fiedler vector at a point reflects its geometric information.

This hypothesis posits that the hot and cold spots present the greatest geodesic distance (shortest distance in graph) in comparison to any other pair of nodes. It also suggests that the Fiedler vector values of nodes positioned at a significant geodesic distance from each other will exhibit distinct values. Specifically, if two nodes are part of two separate dense clusters, their Fiedler vector values will diverge due to the considerable geodesic distance across the tube. The Fiedler Vector can be interpreted as a 1D embedding of the graph, whereby its values encapsulate information about the structure of a graph and spatial contextual information.

In relation to the requirement of smoothness, we can rephrase the definition of the eigenvector and eigenvalue, $\mathbf{L}\boldsymbol{x} = \lambda\boldsymbol{x}$, as follows

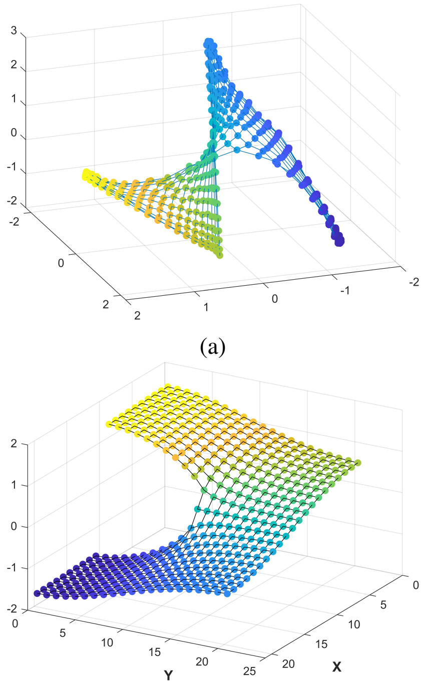
$$\begin{aligned} \mathbf{u}_k^T \mathbf{L} \mathbf{u}_k &= \sum_{m=0}^{N-1} u_k(m) \sum_{n=0}^{N-1} A_{mn} \left(u_k(m) - u_k(n)\right) \\ &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{mn} \left(u_k^2(m) - u_k(m) u_k(n)\right) \end{aligned} \tag{9}$$

Next, owing to the symmetry of the adjacency matrix A ($A_{mn} = A_{nm}$), we have

$$\begin{aligned} \mathbf{u}_k^T \mathbf{L} \mathbf{u}_k =& \frac{1}{2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{mn} \left(u_k^2(m) - u_k(m) u_k(n)\right) \\ &+ \frac{1}{2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{mn} \left(u_k^2(n) - u_k(n) u_k(m)\right) \\ =& \frac{1}{2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{mn} \left(u_k(n) - u_k(m)\right)^2 = \lambda \end{aligned} \tag{10}$$

It is evident that the eigenvalues mirror the variation between nodes and their adjacent counterparts. Hence, the lower the eigenvalue, the smoother the corresponding eigenvector. Therefore, the second eigenvector will be the smoothest one, barring the first eigenvector, which is a constant vector.

Fig.4a illustrates the Fiedler vector in the form of node labels, while Fig.4b shows how it is projected back onto the component and the resulting joint space. Observe that the value of the Fiedler vector can unveil the spatial contextual information of the component, or in other terms, it mirrors the shape of the component. The nodes that are far away in physical distance have distinct differences in their Fiedler vector values. The new input space is then developed as $[\boldsymbol{x}, t, z]$ where z is defined as the obtained Fiedler vector. The implementation of such a case usually uses a finite element mesh to construct the graph and obtain the Fiedler vector value on the graph nodes. Further, FE extrapolation is employed to the value in the whole field. Code for calculating the Fiedler vector from a FE mesh is available at https://github.com/hl4220/Physics-informed-Neural-Network-with-Graph-Embedding.git.



(a)

(b)

Fig. 4: (a) Visualised graph in the eigenspace, (b) Visualised graph layout in higher-dimensional joint space.

III. Results

We considered two case studies: a heat conduction model and a crack modelling in solid mechanics. Both problems were examined using Physics-Informed Neural Network and the enhanced PINN with Graph Embedding (GPINN). The PINN and GPINN models are both feedforward neural networks that share the same parameters except for the input dimension. The processes of graph construction and extra dimension extraction are explained, and the results are compared with high-precision finite element method solutions serving as ground truth.

A. Model of heat propagation

1) Problem Formulation: In this study, the heat propagation problem was defined to find the steady temperature field in a 2D 'house' with the domain and boundary conditions demonstrated in Fig.5. The house presents a square area with two walls that separate the house to some extent. A circular heat source (highlighted in red in Fig.5) was located at a corner

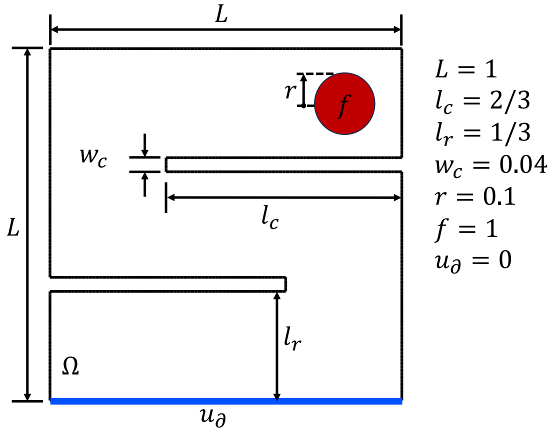


Fig. 5: Schematic of the heat propagation problem. The domain of the 2D 'house' is defined as Ω; there is a heat source f in Ω; the Dirichlet boundary condition is assigned on the boundary at the bottom of Ω that represents a 'window' whose temperature u_∂ is the same as 'outside'.

of the house and the boundary away from the heat source (blue bottom line) was the 'window' of the house that was thin enough to keep the temperature identical to that of the 'outside'.

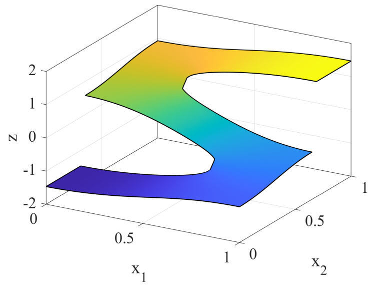


Fig. 6: Input spatial model in the topological space. A 2D input space is expanded to 3D space by incorporating the extra dimension. The Fiedler vector determines the extra dimension to keep the physical consistency of the input model. The colour map represents the value distribution of the extra dimension.

2) Model Architecture & Dataset & Training Strategy: The PINN model was a feedforward neural network with an input dimension of 2 for the Euclidean coordinates $[x, y]$ and an output dimension of 1 for the scalar field of temperature. The hidden sizes of hidden layers were $[64, 64, 128, 128, 64]$, and tanh functions were used as activation functions. The GPINN model shared the same architecture except for the input dimension which was now $[x, y, z]$ where z denotes the Fiedler vector dimension. The FEM results derived from a very fine mesh (256×256) serve as ground truth and the whole point space in Fig.7. Next, 2000, 1000 and 100 points were randomly sampled from the fine mesh and utilised for $\mathcal{L}_{\text{PDE}}$, $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{BC}}$ respectively. And 300 points were sampled and considered as the test set. An Adam optimizer with a learning rate of 0.005 was selected and the model was trained for 50,000 epochs. The relative error of the temperature prediction was used as the evaluation metric.

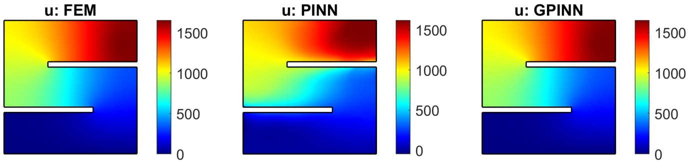


Fig. 7: Reference (FEM) and sample (NN) solutions of the steady temperature field.

3) Loss Functions: A steady-state temperature field follows the Poisson equation for the balance of internal heat sources as well as thermal boundary conditions [3], [9], and is given by

$$\begin{aligned} \Delta u(\boldsymbol{x}) &= f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= u_\partial(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \\ \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n} &= v_\partial(\boldsymbol{x}) \quad \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{11}$$

where the second and the third equations denote the Dirichlet and Neumann boundary conditions, respectively, while Δ denotes the Laplace operator. The loss function employed in PINN/GPINN can be expressed as:

$$\begin{aligned} \mathcal{L} &= \omega_1 \mathcal{L}_{\text{PDE}} + \omega_2 \mathcal{L}_{\text{Data}} + \omega_4 \mathcal{L}_{\text{BC}}, \\ \mathcal{L}_{\text{PDE}} &= \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \Delta u(\boldsymbol{x}_p) - f(\boldsymbol{x}_p) \right|^2, \\ \mathcal{L}_{\text{data}} &= \frac{1}{N_D} \sum_{i=1}^{N_D} \left| u(\boldsymbol{x}_D) - u^*(\boldsymbol{x}_D) \right|^2, \\ \mathcal{L}_{\text{BC}} &= \frac{1}{N_{dbc}} \sum_{i=1}^{N_{dbc}} \left| u(\boldsymbol{x}_{dbc}) - u_\partial(\boldsymbol{x}_{dbc}) \right|^2 + \\ & \frac{1}{N_{nbc}} \sum_{i=1}^{N_{nbc}} \left| \nabla u(\boldsymbol{x}_{nbc}) \cdot \boldsymbol{n} - v_\partial(\boldsymbol{x}_{nbc}) \right|^2, \end{aligned} \tag{12}$$

The weights of the losses were assigned as $\omega_1 = 1e-5$, $\omega_2 = 1$, $\omega_3 = 1$.

4) Results: The problem was investigated by both PINN and GPINN. Fig.7 illustrates the performances of the PINN and GPINN compared with the FEM results. Fig.8 presents the relative errors of the PINN and GPINN to the FEM results. From Fig.7 and 8, GPINN produced satisfactory outcomes when compared to the reference FEM results, particularly in this problem where the traditional PINN exhibited inadequate performance.

Fig.7(b) shows the limitation of traditional PINNs in dealing with relatively discontinuous fields. The large relative errors usually occur at the walls which create discontinuity. This reveals the drawback of PINNs which employ the Euclidean space directly in modelling that does not correspond to the real physical distance. After incorporating an extra-dimension that expands the problem from 2D Euclidean space to 3D joint space, the PINN model was injected with spatial contextual information. The extra-dimension z was determined

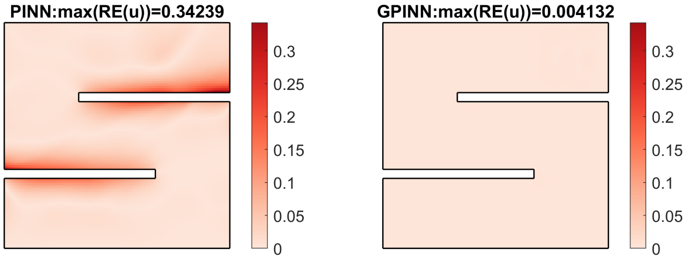


Fig. 8: Relative errors (RE) of the NN solutions to the reference FEM solution: $\mathrm{RE}(u) = |u - u^*| \,/\, \max\left(|u^*|\right)$. The subfigure on the left side is the relative error of PINN while the right one represents the GPINN.

by the Fiedler vector from graph theory as introduced in Section Fiedler Vector: Relation with Heat Equation. The input 3D coordinate model in the joint space is shown in Fig.6.

B. Model of single-side crack

1) Problem Formulation: The second case study focused on a linear elastic simulation of a single-side crack model. Traditional PINNs usually exhibit weaknesses in dealing with crack problems, due to the strong dscontinuity near the crack [4]–[6], [10], [11]. This case study involves a tensile test for a model with a single-sided crack. The problem schematic is depicted in Fig.9.

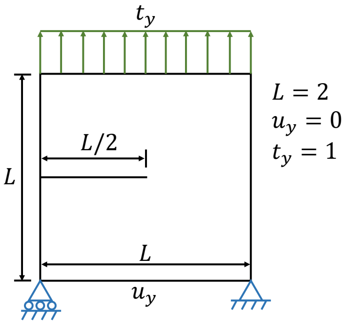


Fig. 9: Schematic of the single-side crack tensile test. The Dirichlet and Neumann boundary conditions are assigned as indicated.

2) Model Architecture & Dataset & Training Strategy: The model architectures of PINN and GPINN were same as the models in the heat propagation task, except for the output dimension which was 2 now here since the desired displacement field is a vector field with directions x and y. The FEM results derived from a very fine mesh (256×256) served as ground truth and the whole point space. The size of the training points for the losses $\mathcal{L}_{\text{PDE}}$, $\mathcal{L}_{\text{data}}$ and $\mathcal{L}_{\text{BC}}$ were respectively 1000, 1000 and 200, while 300 points were employed for the test set. An Adam optimizer with the learning rate of 0.0003 was selected and the model was trained over 50,000 epochs. The relative error of the displacement prediction was used as the evaluation metric.

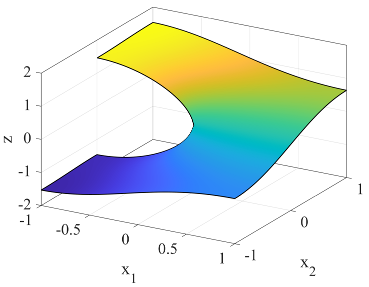


Fig. 10: Input spatial model in the topological space. A 2D input space is expanded to 2D space by the incorporated extra dimensions. This extra dimensions is determined by the Fiedler vector to maintain physical consistency of the input model. The colour map represents the value distribution of the extra-dimension.

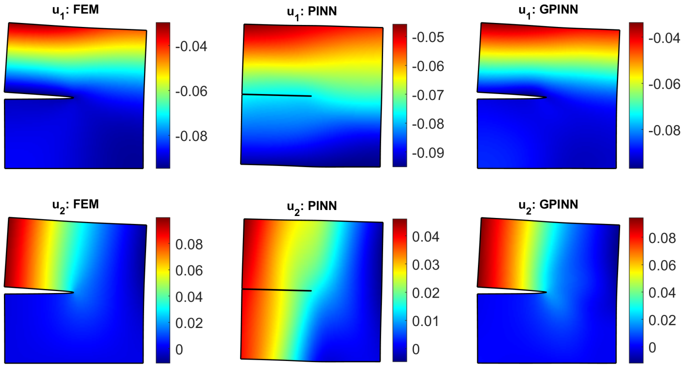


Fig. 11: Reference (FEM) and Sample (NN) solutions of the displacement field.

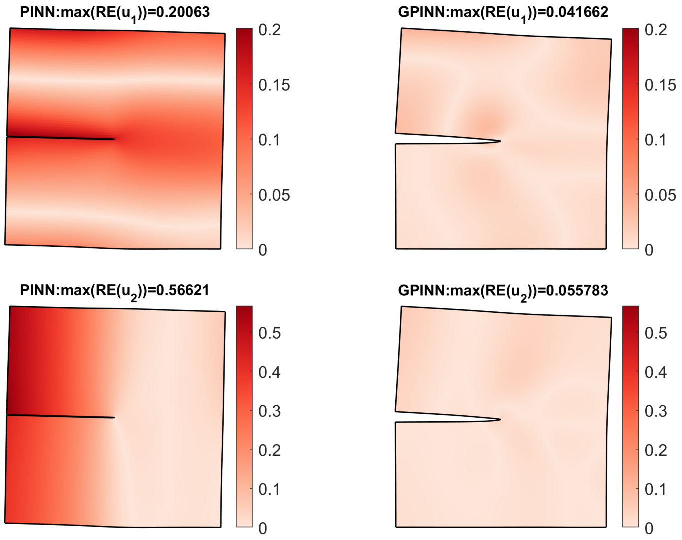


Fig. 12: Relative errors (RE) of the NN solutions to the reference FEM solution: $\mathrm{RE}(u) = |u - u^*| \,/\, \max\left(|u^*|\right)$. The subfigures on the left side are the relative errors of PINN while the right ones represent the relative errors GPINN.

3) Loss Functions: The governing equation of the linear elasticity is given by

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) &= \boldsymbol{0}, \boldsymbol{x} \in \Omega, \\ \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) &= \boldsymbol{t}(\boldsymbol{x}), \boldsymbol{x} \in \partial\Omega, \\ \boldsymbol{u}(\boldsymbol{x}) &= \boldsymbol{u}_{\partial}(\boldsymbol{x}), \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{13}$$

where $\boldsymbol{u}$ and $\boldsymbol{\sigma}$ denote respectively the displacement and stress vectors, respectively, being effective on the domain Ω with the boundary $\partial\Omega$. The second and third equations in Eq.13 denote the Neumann and Dirichlet boundary conditions, respectively, where $\boldsymbol{t}$ and $\boldsymbol{u}_{bc}$ are the applied traction and displacement conditions. Note that the output field $\boldsymbol{u}$ is a vector field that differs from the temperature field in the previous problem [12]. The bold symbols utilised in those equations thus refer to vectors rather than scalars as shown Eq.13.

In this work, the energy-based loss function was employed for the PINN and GPINN models. Energy-based loss functions aim at minimising the potential energy of the entire structure, which considers global information and requires lower differential order compared to collocation loss function [13]. It has also been shown to be better in modelling crack problems [10]. The loss function employed in the PINN and GPINN in this case study is stated as [10], [13]:

$$\begin{aligned} \mathcal{L} = \omega_1 \mathcal{L}_{PDE} + \omega_2 \mathcal{L}_{\text{Data},\boldsymbol{u}} + \omega_3 \mathcal{L}_{\text{Data},\boldsymbol{\sigma}} + \\ \omega_4 \mathcal{L}_{\text{BC},\boldsymbol{u}} + \omega_5 \mathcal{L}_{\text{BC},\boldsymbol{\sigma}}, \\ \mathcal{L}_{PDE} = \int_{\Omega} \frac{1}{2} \boldsymbol{\sigma}(\boldsymbol{x}_p) \boldsymbol{\varepsilon}(\boldsymbol{x}_p) \mathrm{d}\Omega - \\ \int_{\partial\Omega} \boldsymbol{t}(\boldsymbol{x}_{nbc}) \boldsymbol{u}(\boldsymbol{x}_{nbc}) \mathrm{d}\partial\Omega, \\ \mathcal{L}_{\text{Data},\boldsymbol{u}} = \frac{1}{N_D} (\sum_{i=1}^{N_D} \|\boldsymbol{u}(\boldsymbol{x}_D) - \boldsymbol{u}^*(\boldsymbol{x}_D)\|^2) \\ \mathcal{L}_{\text{Data},\boldsymbol{\sigma}} = \frac{1}{N_D} (\sum_{i=1}^{N_D} \|\boldsymbol{\sigma}(\boldsymbol{x}_D) - \boldsymbol{\sigma}^*(\boldsymbol{x}_D)\|^2), \\ \mathcal{L}_{\text{BC},\boldsymbol{u}} = \frac{1}{N_{dbc}} \sum_{i=1}^{N_{dbc}} \|\boldsymbol{u}(\boldsymbol{x}_{dbc}) - \boldsymbol{u}_{bc}\|^2 \\ \mathcal{L}_{\text{BC},\boldsymbol{\sigma}} = \frac{1}{N_{nbc}} \sum_{i=1}^{N_{nbc}} \|\boldsymbol{\sigma}(\boldsymbol{x}_{nbc}) \cdot \boldsymbol{n} - \boldsymbol{t}(\boldsymbol{x}_{nbc})\|^2, \end{aligned} \tag{14}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ denote the stress and strain

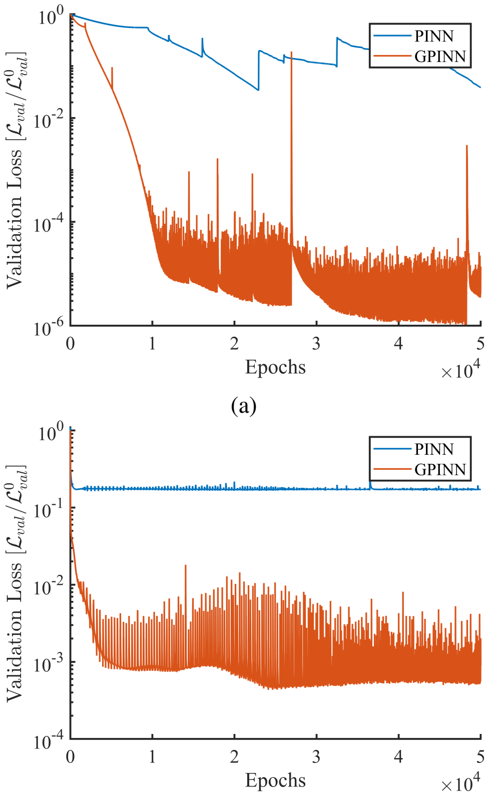
$$\begin{aligned} \boldsymbol{\varepsilon}(\boldsymbol{x}) &= \nabla \boldsymbol{u}(\boldsymbol{x}), \\ \boldsymbol{\sigma}(\boldsymbol{x}) &= \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{x}), \end{aligned} \tag{15}$$

The weights of losses were assigned as $\omega_1 = 1e-5$, $\omega_2 = 6$, $\omega_3 = 2e-6$, $\omega_4 = 6$, $\omega_5 = 1e-5$.

4) Results: The results of both PINN and GPINN methods are presented in Fig.11, alongside the reference FEM results. The distribution of relative error is displayed in Fig.12. Observe that the GPINN method yielded promising results in comparison to the reference FEM results. Traditional PINN, however, seems to have fallen short in accurately capturing the solution features due to the lack of spatial contextual information. The input model enriched with the extra dimension is presented in Fig.10.

IV. Discussion

Fig. 13 provides a detailed comparison between the GPINN and traditional PINN, in terms of the evolution of the test loss for both the two problems. Upon examination, it is evident that GPINN significantly speeds up the convergence during the training and consistently delivers more accurate results.



(a)

(b)

Fig. 13: Evolution of the normalised validation loss ($\mathcal{L}_{\text{val}}/\mathcal{L}^0_{\text{val}}$) for (a) heat propagation (b) crack modelling problems. The validation losses are normalised by their value at the first epoch.

V. Conclusion

We have introduced a novel GPINN method with the aim to perform PINN in the joint Euclidean and topological (graph based) space. This has enabled better capturing of the physical characteristics of a structure and has enriched the traditional PINN model with spatial contextual information. This has been achieved by incorporating an extra dimension into the input space, thus creating a graph-based spatial model which offers a better capture of the pathological property of a structure. This extra dimension has been derived from the graph theory, by utilizing the Fiedler vector, which is the eigenvector corresponding to the second smallest eigenvalue of graph Laplacian. Fig. 13 shows that the graph embedding significantly enhances PINNs in both performance and training speed when dealing with problems within complex domains. The potential of this proposed GPINN method in engineering applications is significant, considering its ease of implementation and the substantial improvements it brings to the performance of PINN.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[2] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Optics express*, vol. 28, no. 8, pp. 11 618–11 633, 2020.

[3] T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb, "Can physics-informed neural networks beat the finite element method?" *arXiv preprint arXiv:2302.04107*, 2023.

[4] S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk, "Transfer learning enhanced physics informed neural network for phase-field modeling of fracture," *Theoretical and Applied Fracture Mechanics*, vol. 106, p. 102447, 2020.

[5] Y. Gu, C. Zhang, P. Zhang, M. V. Golub, and B. Yu, "Enriched physics-informed neural networks for 2d in-plane crack analysis: Theory and matlab code," *International Journal of Solids and Structures*, vol. 276, p. 112321, 2023.

[6] H. Lian, P. Zhao, M. Zhang, P. Wang, and Y. Li, "Physics informed neural networks for phase field fracture modeling enhanced by length-scale decoupling degradation functions," *Frontiers in Physics*, vol. 11, p. 177, 2023.

[7] M. K. Chung, S. Seo, N. Adluru, and H. K. Vorperian, "Hot spots conjecture and its application to modeling tubular structures," in *Machine Learning in Medical Imaging: Second International Workshop, MLMI 2011, Held in Conjunction with MICCAI 2011, Toronto, Canada, September 18, 2011. Proceedings 2*. Springer, 2011, pp. 225–232.

[8] R. Banuelos and K. Burdzy, "On the hot spots conjecture of j rauch," *Journal of functional analysis*, vol. 164, no. 1, pp. 1–33, 1999.

[9] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *Journal of Heat Transfer*, vol. 143, no. 6, 2021.

[10] B. Zheng, T. Li, H. Qi, L. Gao, X. Liu, and L. Yuan, "Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data," *International Journal of Mechanical Sciences*, vol. 223, p. 107282, 2022.

[11] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, 2021.

[12] R. Arora, P. Kakkar, B. Dey, and A. Chakraborty, "Physics-informed neural networks for modeling rate-and temperature-dependent plasticity," *arXiv preprint arXiv:2201.08363*, 2022.

[13] J. Bai, H. Jeong, C. Batuwatta-Gamage, S. Xiao, Q. Wang, C. Rathnayaka, L. Alzubaidi, G.-R. Liu, and Y. Gu, "An introduction to programming physics-informed neural network-based computational solid mechanics," *arXiv preprint arXiv:2210.09060*, 2022.