Physics-Informed Kolmogorov-Arnold Networks for multi-material elasticity problems in electronic packaging

Yanpeng Gong[a], Yida He[a], Yue Mei[b,*], Xiaoying Zhuang[c,d,*], Fei Qin[a], Timon Rabczuk[e]

[a]*Institute of Electronics Packaging Technology and Reliability, Department of Mechanics, Beijing University of Technology, Beijing, 100124, China*
[b]*State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, International Research Center for Computational Mechanics, Dalian University of Technology, Dalian 116023, China*
[c]*Chair of Computational Science and Simulation Technology, Institute of Photonics, Department of Mathematics and Physics, Leibniz University Hannover, 30167 Hannover, Germany*
[d]*Department of Geotechnical Engineering, College of Civil Engineering, Tongji University, Shanghai, 200092, China*
[e]*Institute of Structural Mechanics, Bauhaus University Weimar, 99423 Weimar, Germany*

Abstract

This paper proposes a Physics-Informed Kolmogorov-Arnold Network (PIKAN) method for analyzing elasticity problems in electronic packaging multi-material structures. The core innovation lies in replacing Multi-Layer Perceptrons (MLPs) with Kolmogorov-Arnold Networks (KANs) within the energy-based Physics-Informed Neural Networks (PINNs) framework. The method constructs admissible displacement fields that automatically satisfy essential boundary conditions and employs various numerical integra-

---

[*]Corresponding author

Email addresses: `yanpeng.gong@bjut.edu.cn` (Yanpeng Gong), `meiyue@dlut.edu.cn` (Yue Mei), `zhuang@iop.uni-hannover.de` (Xiaoying Zhuang)

tion schemes to compute loss functions for network optimization. Unlike traditional PINNs that require domain decomposition and penalty terms for multi-material problems, KANs' trainable B-spline activation functions provide inherent piecewise function characteristics that naturally accommodate material property discontinuities. Consequently, this approach requires only a single KAN to achieve accurate approximation across the entire computational domain without subdomain partitioning and interface continuity constraints. Numerical validation demonstrates PIKAN's accuracy and robustness for multi-material elasticity problems. The method maintains high accuracy while significantly reducing computational complexity compared to domain decomposition approaches. Results confirm PIKAN's unique advantages in solving multi-material problems and its significant potential for electronic packaging structure analysis. Source codes are available at https://github.com/yanpeng-gong/PIKAN-MultiMaterial.

1. Introduction

Moore's Law, proposed by Gordon Moore in 1965, accurately predicted the semiconductor industry's development trajectory and continues to drive innovation in electronic packaging technology. However, with the emergence of concepts such as 'More than Moore' and trends toward high-density integration and functional diversification of electronic devices, the semiconductor industry has entered a new phase highly dependent on packaging technology development [1]. Electronic packaging provides mechanical sup-

port, power distribution, electrical interconnection, and signal transmission to ensure product reliability [2]. Packaging structures typically involve multiple materials with different properties, featuring geometric discontinuities at interfaces and exhibiting multi-material, multi-scale characteristics. The reliability of such structures has become a critical design consideration, creating an urgent need for accurate and effective modeling methods in packaging structure analysis.

Numerical simulation technology has become the mainstream method for electronic packaging reliability analysis due to its advantages of easy implementation, high accuracy, and freedom from environmental constraints. In this research field, researchers have developed various numerical simulation methods to address different reliability problems in electronic packaging structures, including the finite element method (FEM) [3, 4], boundary element method (BEM) [5], FEM-BEM coupling method [6, 7], and finite difference method (FDM) [8]. Liu et al. [9] proposed a layered medium integral equation solver based on parallel computing and fast Fourier transform, achieving efficient extraction of electronic packaging network parameters. Gong et al. [10] developed an isogeometric boundary element method that accurately analyzes heat transfer problems in multi-scale electronic packaging structures with arbitrary heat sources using fewer degrees of freedom. Feng et al. [11] addressed the over-stiffness problem of traditional FEM in IGBT module multi-physics analysis through an improved NS-FEM method, enhancing computational performance. Qin and collaborators [12] used a coupling method combining finite elements and boundary elements to study elastoplastic behavior in multi-scale electronic packaging structures. Addi-

tionally, thermomechanical coupling phase field models have been applied to study damage evolution and failure mechanisms in various electronic packaging interconnect structures [13, 14].

The development of artificial intelligence (AI) technology and enhanced computational resources have provided opportunities to develop deep learning as a new approach for solving electronic packaging reliability problems. Kang et al. [15] proposed a machine learning-based thermomechanical analysis model for MMIC packaging that rapidly evaluates the impact of design parameters—including material properties, geometric features, and thermal boundary conditions-on temperature and stress distributions, providing accurate simulation alternatives for electronic packaging design. Zaghari et al. [16] investigated the collaborative design of through-hole interconnect structures in silicon and glass interposers using machine learning-based multi-objective optimization methods, analyzing the effects of different geometric parameters on electrical, thermal, and mechanical performance. Yu et al. [17] developed a GUI-based AI platform that utilizes multiple deep learning models to achieve high-precision prediction of warpage behavior in fan-out wafer-level packaging. Hou et al. [18] established a hybrid machine learning framework based on SVM and SVR for predicting bonding wire loop profiles. This approach achieves significant computational efficiency gains compared to finite element simulations, providing an effective tool for wire geometry optimization in high-density chip packaging designs. The aforementioned methods typically train neural networks based on real experimental data or high-confidence numerical calculation results (such as reliable solutions obtained through FEM) to construct surrogate models. However, training such models

often requires substantial data support. In most practical engineering problems, available observational or measurement data is typically limited, which may cause trained models' prediction results to deviate from known physical laws or lose physical consistency [19], thereby reducing prediction accuracy and constraining model generalization capability and interpretability. Machine learning techniques, when coupled with conventional computational methods, offer tremendous potential to revolutionize modeling and simulation in engineering and the sciences, opening new avenues for next-generation computer-aided analysis and design [20].

To reduce neural networks' dependence on high-quality data, researchers embed physical laws into loss functions to enhance models through physics-informed constraints. Physics-Informed Neural Networks (PINNs), proposed by Raissi et al. [21] in 2019, became the pioneering work in this direction. The implementation of PINNs benefits from the mature application of automatic differentiation functionality in machine learning frameworks such as PyTorch and TensorFlow. As a meshfree method using neural networks as approximation functions, PINNs inherit the advantages of meshfree methods in handling mesh distortion and large deformation problems [22] and can solve PDE problems that traditional numerical algorithms can address. PINNs encompass two forms: strong-form original PINNs and energy-form PINNs. Samaniego et al. [23] introduced the Deep Ritz Method proposed by E and Yu [24] into computational mechanics, forming energy-form PINNs called the Deep Energy Method (DEM). Strong-form PINNs construct loss functions by directly combining PDEs through weighted residual methods, while DEM constructs loss functions based on variational principles such as

the minimum potential energy principle in mechanics. In electronic packaging structure analysis, although there are limited studies using traditional data-driven surrogate models based on physical simulation data [25], research directly applying PINNs to such simulations has rarely been reported.

Multi-Layer Perceptrons (MLPs) serve as the core architecture of PINNs and the standard model for approximating nonlinear functions in machine learning. The universal approximation theorem [26] demonstrates that MLPs can approximate any continuous function with arbitrary precision given sufficient hidden neurons and appropriate weight configurations. However, their limited parameter efficiency significantly increases computational complexity and causes convergence difficulties when handling complex problems [27]. Recently, Liu et al. [28] proposed Kolmogorov-Arnold Networks (KANs) as a novel alternative to MLPs. While structurally similar to MLPs, KANs use learnable one-dimensional functions [29], to replace linear weight parameters, with nodes simplified to perform input summation. KANs integrate spline function approximation capabilities with MLP network architecture, typically achieving model expression with smaller computational graphs. Given that MLPs suffer from large parameter scales and limited interpretability, while KANs offer fewer parameters and B-spline-based activation functions that align more naturally with numerical PDE solving algorithms, integrating KANs into PINNs to replace MLPs is theoretically well-motivated.

Electronic packaging structures are typically composed of heterogeneous materials with different properties, exhibiting multi-material, multi-interface structural characteristics. When applying PINNs to solve multi-material problems, the current mainstream approach uses different neural networks

to describe physical fields in different material domains. Diao et al. [30] proposed a strong-form PINNs framework for solving multi-material solid mechanics problems by partitioning the computational domain through domain decomposition, utilizing independent sub-networks to characterize mechanical behavior of different materials, and introducing interface regularization terms to handle contact relationships and discontinuities. Wang et al. [31] focused on variational problems and proposed a conservative energy method based on subdomain neural networks for solving heterogeneous material and complex geometry problems. However, PINNs based on subdomain partitioning have inherent limitations. To accurately describe interface contact relationships, additional hyperparameters must be introduced, which typically rely on empirical parameter tuning. Although research has explored hyperparameter optimization strategies [32, 33], obtaining optimal parameter combinations remains challenging when solving specific problems. Furthermore, this method requires independent optimization of neural networks for each subdomain, which significantly increases training complexity and often causes the optimization process to converge to local rather than global optima.

Based on the aforementioned research background, this paper adopts a Physics-Informed Kolmogorov-Arnold Network (PIKAN) method for solving multi-material problems in solid mechanics and extends its application to electronic packaging multi-material structure analysis. This method replaces traditional multi-layer perceptrons with KANs and combines them with the DEM. Compared to strong-form PINNs, DEM has stronger physical interpretability and avoids high-order differential operations, resulting in higher

computational efficiency and superior accuracy [34]. In PIKAN, admissible displacement fields are constructed [31, 32] to strictly satisfy essential displacement boundary conditions of geometric structures. Furthermore, benefiting from KANs' trainable one-dimensional activation functions and inherent piecewise function characteristics, this method requires no subdomain partitioning during implementation and can approximate the entire solution domain using only a single KAN. This characteristic effectively circumvents the need for additional interface information and excessive hyperparameters, significantly reducing training difficulty and computational cost. Numerical example verification demonstrates that the proposed PIKAN method performs well when handling structures containing multiple materials and different interface configurations, broadening the application of PINNs in multi-material problems. Meanwhile, its application in electronic packaging multi-material structure analysis validates the method's potential and value in electronic packaging structure reliability analysis.

This paper is organized as follows: Section 2 presents the governing equations for elasticity problems in solid mechanics, the DEM, and the construction method for admissible displacement fields. Section 3 provides a detailed description of the proposed PIKAN method. Section 4 presents specific strategies for applying PIKAN to solve multi-material problems. Section 5 validates the effectiveness of the proposed method through multi-material model examples and applies it to electronic packaging multi-material structure analysis. Finally, Section 6 summarizes the research conclusions.

2. Deep energy method-based computational solid mechanics

This section reviews the fundamental equations of elasticity problems, then introduces PINNs with energy-based loss functions for solving such problems. Finally, we present the construction of admissible displacement fields that naturally satisfy essential displacement boundary conditions.

2.1. Fundamental equations of elasticity problems

The governing equation at any point $\mathbf{x}$ within the elastic body domain Ω is

$$\sigma_{ij,j} + \bar{f}_i = 0, \quad \mathbf{x} \in \Omega \tag{1}$$

where σ is the Cauchy stress tensor, $\bar{f}_i$ is the body force per unit volume, and $(\cdot)_{,j} = \partial(\cdot)/\partial x_j$. For isotropic linear elastic materials, the stress-strain relationship is expressed as

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + 2\mu \varepsilon_{ij} = D_{ijkl} \varepsilon_{kl} \tag{2}$$

where δ_{ij} is the Kronecker delta, ε_{ij} is the strain tensor, D_{ijkl} is the elasticity tensor, and λ and μ are the Lamé constants characterizing material properties.

Under the small deformation assumption, the strain-displacement relationship is expressed as

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{3}$$

where u_i represents the i-th displacement component. The governing equations are closed through boundary conditions. The complete boundary of the elastic body Ω is defined as Γ. A portion of the boundary where displacements are prescribed is called the essential (Dirichlet) boundary, denoted as

9

Γ_u. The portion where tractions are prescribed is called the natural (Neumann) boundary, denoted as Γ_σ. These two boundary segments constitute the complete boundary

$$\Gamma_u \cup \Gamma_\sigma = \Gamma, \quad \Gamma_u \cap \Gamma_\sigma = \emptyset \tag{4}$$

The displacement boundary conditions are expressed as

$$u_i = \bar{u}_i, \quad \mathbf{x} \in \Gamma_u \tag{5}$$

where $\bar{u}_i$ is the prescribed displacement on the corresponding boundary. The traction boundary conditions are expressed as

$$p_i = \sigma_{ij} n_j = \bar{p}_i, \quad \mathbf{x} \in \Gamma_\sigma \tag{6}$$

where n_j is the j-th component of the outward unit normal vector, and $\bar{p}_i$ is the prescribed traction on the corresponding boundary.

2.2. Deep energy method for solving elasticity problems

For elastic systems, DEM utilizes the principle of minimum potential energy from variational elasticity theory. Based on the principle of minimum potential energy, an energy-based loss function that utilizes the total potential energy of the system as the optimization objective has been proposed within the variational PINNs framework [23]. The core idea is to directly construct the total potential energy of the elastic body as the loss function for neural network training. For problems without body forces, the specific mathematical expression of this loss function is

$$\mathcal{L} = \Psi_{\text{in}} - \Psi_{\text{ex}} \tag{7}$$

10

where Ψ_{in} and Ψ_{ex} represent the internal strain energy and the potential energy of external forces, respectively, with their specific expressions given by

$$\Psi_{\text{in}} = \int_\Omega \frac{1}{2} D_{ijkl}\varepsilon_{ij}\varepsilon_{kl}\mathrm{d}\Omega = \int_\Omega \frac{1}{2}\sigma_{ij}\varepsilon_{ij}\mathrm{d}\Omega \tag{8}$$

$$\Psi_{\text{in}} = \int_{\Gamma_\sigma} \bar{p}_i u_i \mathrm{d}\Gamma \tag{9}$$

Therefore, the energy-based loss function in Eq. (7) can be written as

$$\mathcal{L} = \int_\Omega \frac{1}{2}\sigma_{ij}\varepsilon_{ij}\mathrm{d}\Omega - \int_{\Gamma_\sigma} \bar{p}_i u_i \mathrm{d}\Gamma \tag{10}$$

While DEM shares similarities with FEM in using energy-based variational weak forms, the key distinction lies in its direct optimization of the total potential energy functional, eliminating the need for preliminary derivation and discretization of weak form governing equations required in FEM.

In DEM, the displacement field $\mathbf{u}(\mathbf{x})$ satisfying the prescribed essential boundary conditions is approximated through a neural network

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x};\boldsymbol{\theta})$$
$$\text{s.t. } \mathbf{u}(\mathbf{x}) = \overline{\mathbf{u}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma_u \tag{11}$$

where $\mathbf{F}(\cdot)$ represents the neural network mapping, $\overline{\mathbf{u}}(\mathbf{x})$ denotes the essential boundary conditions, and $\boldsymbol{\theta}$ are the trainable parameters of the neural network.

Based on the principle of minimum potential energy, among all kinematically admissible displacement fields $\mathbf{u}(\mathbf{x})$ that satisfy essential boundary conditions, we seek the solution that minimizes the total potential energy functional

$$\mathbf{u}(\mathbf{x}) = \underset{\mathbf{u}(\mathbf{x})}{\arg\min}\mathcal{L}(\mathbf{u}(\mathbf{x})) = \underset{\mathbf{u}(\mathbf{x})}{\arg\min}\left\{\int_\Omega \psi(\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x})))\mathrm{d}\Omega - \int_{\Gamma_\sigma} \overline{\mathbf{p}}\cdot\mathbf{u}(\mathbf{x})\mathrm{d}\Gamma\right\} \tag{12}$$

where $\psi(\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x})))$ is the strain energy density function, which is a function of the strain tensor $\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}))$.

2.3. Construction of admissible displacement fields

As described previously, DEM selects the displacement field that minimizes potential energy among all admissible displacement fields. The key to implementing admissible displacement fields lies in constructing displacement trial functions that naturally satisfy essential boundary conditions, embedding these constraints directly into the network architecture to ensure that the generated displacement field always satisfies essential boundary conditions during the optimization process.
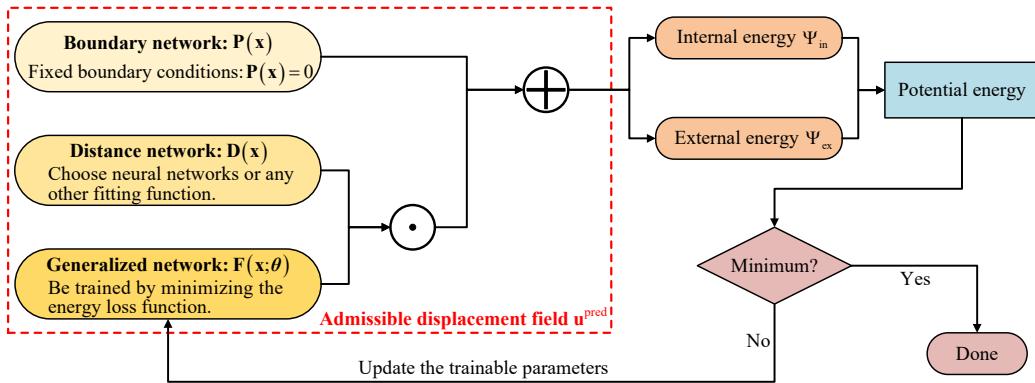

Figure 1: DEM process overview.

As shown in Fig. 1, the general construction of admissible displacement fields can be expressed as

$$\mathbf{u}^{\mathrm{pred}}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) + \mathbf{D}(\mathbf{x}) \odot \mathbf{F}(\mathbf{x}; \boldsymbol{\theta}) \tag{13}$$

where $\odot$ denotes element-wise product, $\mathbf{P}(\mathbf{x})$ is the boundary network that satisfies essential boundary conditions, and $\mathbf{F}(\mathbf{x}; \boldsymbol{\theta})$ represents the neural net-

work approximation. $\mathbf{D}(\mathbf{x})$ is the distance function, representing the shortest distance from the current point to the essential boundary Γ_u

$$\mathbf{D}(\mathbf{x}) = \min_{\mathbf{y} \in \Gamma_u} \|\mathbf{x} - \mathbf{y}\| \qquad (14)$$

The key advantage of this formulation is that when coordinate points are located on the displacement boundary Γ_u, the distance function $\mathbf{D}(\mathbf{x}) = 0$. According to Eq. (13), the constructed displacement field output becomes exactly the prescribed essential boundary condition $\mathbf{P}(\mathbf{x})$. Consequently, essential boundary conditions are automatically satisfied during neural network training without requiring additional penalty terms and hyperparameter tuning. For detailed discussion of distance functions in this context, refer to [35].

$\mathbf{F}(\mathbf{x}; \boldsymbol{\theta})$ is the generalized network, whose neural network parameters $\boldsymbol{\theta}$ are trained and optimized through the minimum potential energy loss function

$$\begin{aligned} \mathbf{F}(\mathbf{x}; \boldsymbol{\theta}) &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}\left(\mathbf{u}^{\mathrm{pred}}(\mathbf{x}; \boldsymbol{\theta})\right) \\ &= \arg\min_{\boldsymbol{\theta}} \left\{ \int_{\Omega} \psi\left(\boldsymbol{\varepsilon}(\mathbf{u}^{\mathrm{pred}}(\mathbf{x}; \boldsymbol{\theta}))\right) \mathrm{d}\Omega - \int_{\Gamma_\sigma} \overline{\mathbf{p}} \cdot \mathbf{u}^{\mathrm{pred}}(\mathbf{x}; \boldsymbol{\theta}) \mathrm{d}\Gamma \right\}. \end{aligned} \qquad (15)$$

Note that $\mathbf{F}(\mathbf{x}; \boldsymbol{\theta})$ represents the generalized network output, while the actual displacement field is given by Eq. (13).

3. Methodology of PIKAN

The previous section provided a detailed introduction to using DEM for solving elasticity problems. In this work, instead of using the MLP traditionally employed in DEM, we adopt KAN as the neural network architecture. This section first introduces the fundamental principles of KAN, then presents the PIKAN method and finally describes the sample point distribution strategies employed in the analysis.
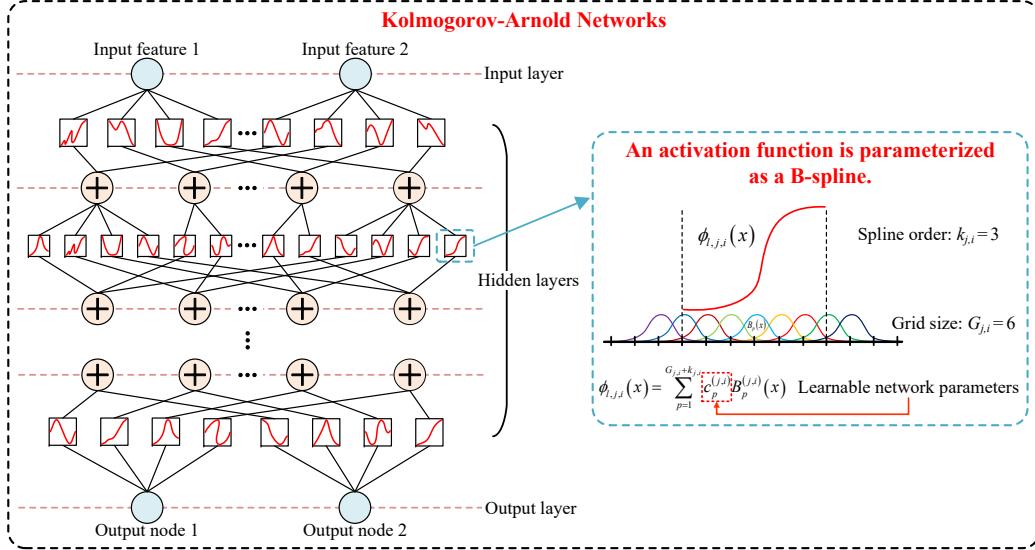
Figure 2: Kolmogorov-Arnold Network schematic.

3.1. Kolmogorov-Arnold Networks (KANs)

KANs are based on the Kolmogorov-Arnold representation theorem and feature a fully connected structure. Traditional MLPs achieve layer-wise transformations by applying fixed nonlinear activation functions to linear combinations of inputs at each node, where weights and biases are learned parameters. During backpropagation training, MLPs update these parameters by computing gradients of the loss function with respect to weights and biases.

In contrast, KANs move activation functions from nodes to edges, where each edge contains learnable univariate functions parameterized as B-spline functions. Instead of applying fixed activation functions at nodes after linear transformations (as in MLPs), KANs apply learnable activation functions on each edge connection. Nodes in KANs perform only summation operations

14

on the outputs from incoming edges. This design enables each edge function to adaptively learn both the weight and activation behavior simultaneously, allowing for more flexible nonlinear transformations and enhanced model expressiveness and interpretability. During training, KAN updates the B-spline coefficients of these edge functions through backpropagation. The KAN architecture is illustrated in Fig. 2.

The core idea of the Kolmogorov-Arnold representation theorem is that any continuous multivariate function f defined on a bounded domain can be represented as a combination of finitely many univariate continuous functions and additions

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) = \sum_{h=1}^{2n+1} \Phi_h \left(\sum_{g=1}^{n} \phi_{h,g}(x_g) \right) \tag{16}$$

where $\phi_{h,g}$ and Φ_h are univariate continuous functions. The starting point of KAN is to parameterize Eq. (16) explicitly, parameterizing all univariate functions $\phi_{h,g}$ and Φ_h as B-spline functions with learnable coefficients.

Define a KAN layer with N_{in} input neurons and N_{out} output neurons

$$\mathbf{\Phi} = \{\phi_{h,g}\}, \quad g = 1, 2, 3, \ldots, N_{\text{in}}; \quad h = 1, 2, 3, \ldots, N_{\text{out}} \tag{17}$$

where $\phi_{h,g}$ has trainable parameters. The original Eq. (16) represents a combination of two KAN layers: the inner function with $N_{\text{in}} = n, N_{\text{out}} = 2n + 1$, and the outer function with $N_{\text{in}} = 2n + 1, N_{\text{out}} = 1$. By stacking more KAN layers, deeper KANs can be obtained. The shape of a KAN is represented by an integer array $[N_0, N_1, N_2, \ldots, N_L]$, where N_i is the number of neurons in the i-th layer. Between the l-th and $(l + 1)$-th layers, there are $N_l N_{l+1}$ activation functions. The activation function connecting the i-th

15

neuron (l, i) in the l-th layer and the j-th neuron $(l + 1, j)$ in the $(l + 1)$-th layer is denoted as

$$\phi_{l,j,i}, \quad l = 0, 1, \ldots, L - 1; \quad i = 1, 2, \ldots, N_l; \quad j = 1, 2, \ldots, N_{l+1} \tag{18}$$

The activation value of the $(l + 1)$-th layer is computed as

$$x_{l+1,j} = \sum_{i=1}^{N_l} \phi_{l,j,i}(x_{l,i}), \quad j = 1, 2, \ldots, N_{l+1} \tag{19}$$

In vector form, this can be written as

$$\mathbf{x}_{l+1} = \begin{bmatrix} \sum_{i=1}^{N_l} \phi_{l,1,i}(x_{l,i}) \\ \sum_{i=1}^{N_l} \phi_{l,2,i}(x_{l,i}) \\ \vdots \\ \sum_{i=1}^{N_l} \phi_{l,N_{l+1},i}(x_{l,i}) \end{bmatrix} = \mathbf{\Phi}_l(\mathbf{x}_l) \tag{20}$$

where $\mathbf{\Phi}_l$ represents the function matrix of the l-th KAN layer.

Since the activation functions are parameterized as B-spline functions, each activation function $\phi_{l,j,i}$ can be expressed as

$$\phi_{l,j,i}(x) = \sum_{p=1}^{G_{j,i}+k_{j,i}} c_p^{(j,i)} B_p^{(j,i)}(x) \tag{21}$$

where $G_{j,i}$ and $k_{j,i}$ are the grid size and spline order for the activation function connecting neuron (l, i) to neuron $(l + 1, j)$, $c_p^{(j,i)}$ are the trainable B-spline coefficients, and $B_p^{(j,i)}(x)$ are the corresponding B-spline basis functions.

The total number of trainable parameters in layer l is

$$N_{\text{params},l} = \sum_{i=1}^{N_l} \sum_{j=1}^{N_{l+1}} (G_{j,i} + k_{j,i}) \tag{22}$$

16

To enhance the function approximation capability, scaling coefficients are introduced for each activation function [36]. For each activation function $\phi_{l,j,i}$, a corresponding scaling coefficient $m_{j,i}$ is defined. The scaling coefficient matrix $\mathbf{M}_l$ is

$$\mathbf{M}_l = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,N_l} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,N_l} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N_{l+1},1} & m_{N_{l+1},2} & \cdots & m_{N_{l+1},N_l} \end{bmatrix} \tag{23}$$

The scaled activation functions become

$$\phi_{l,j,i}^*(x) = m_{j,i} \cdot \phi_{l,j,i}(x) \tag{24}$$

Additionally, a linear transformation with weight matrix $\mathbf{W}_l$ and activation function $\sigma(\cdot)$ is introduced. The final output of the KAN layer is computed as

$$x_{l+1,j} = \sum_{i=1}^{N_l} m_{j,i} \cdot \phi_{l,j,i}(x_{l,i}) + \sum_{i=1}^{N_l} w_{j,i} \cdot \sigma(x_{l,i}), \quad j = 1, 2, \ldots, N_{l+1} \tag{25}$$

where $w_{j,i}$ are elements of the weight matrix $\mathbf{W}_l$. The introduction of $\sigma(\cdot)$ enables the network to capture smoother solutions, complementing the B-spline functions. The scaling coefficients $m_{j,i}$ and weights $w_{j,i}$ are both trainable parameters optimized during training.

In this work, when constructing KAN, we set the grid size $G_{j,i}$ and B-spline order $k_{j,i}$ to be consistent across all activation functions within each KAN layer. The trainable parameters in KAN layers are given in Table 1. For a KAN network with L layers, given an input vector $\mathbf{x}^{(0)} \in \mathbb{R}^{N_0}$, the

Table 1: Trainable parameters of KAN layers

Parameter	Type	Quantity per layer	Description
$c_p^{(j,i)}$	B-spline coefficients	$N_l \times N_{l+1} \times (G_{j,i} + k_{j,i})$	Coefficients for B-spline basis functions
$m_{j,i}$	Scaling factors	$N_l \times N_{l+1}$	Element-wise scaling factors for activation functions
$w_{j,i}$	Linear weights	$N_l \times N_{l+1}$	Weights for auxiliary linear transformation with $\sigma(\cdot)$

forward propagation through the network is computed sequentially as

$$
\begin{aligned}
\mathbf{x}^{(1)} &= \Phi_0^*(\mathbf{x}^{(0)}) \\
\mathbf{x}^{(2)} &= \Phi_1^*(\mathbf{x}^{(1)}) \\
&\vdots \\
\mathbf{x}^{(L)} &= \Phi_{L-1}^*(\mathbf{x}^{(L-1)})
\end{aligned}
\tag{26}
$$

where $\mathbf{\Phi}_l^*(\mathbf{x}^{(l)}) = \mathbf{\Phi}_l(\mathbf{x}^{(l)}) \odot \mathbf{M}_l + \mathbf{W}_l \cdot \sigma(\mathbf{x}^{(l)})$, $0 \leq l \leq (L-1)$ and $\mathbf{x}^{(L)}$ is the network's final output.

3.2. Physics-Informed Kolmogorov-Arnold Networks (PIKANs)

PIKAN used in this work essentially implement the DEM using KAN network architecture, replacing the MLP traditionally employed in DEM with KAN. Consistent with traditional DEM applications in computational solid mechanics, the loss function construction is based on the principle of minimum potential energy. As shown in Eq. (15), KAN is trained by minimizing the energy-based loss function. Fig. 3 presents a schematic diagram of the complete PIKAN framework.

3.2.1. Geometric scale normalization

Since KAN uses B-spline functions to parameterize activation functions, it is necessary to preset the grid range of B-splines. In this work, the grid

range is typically set to $[0, 1]$ or $[-1, 1]$. The algorithm takes the coordinates of sample points within the computational domain as inputs to the neural network. However, due to the arbitrary geometric dimensions of different models, the network inputs may exceed the preset B-spline grid range, necessitating input normalization to accommodate the grid range settings.
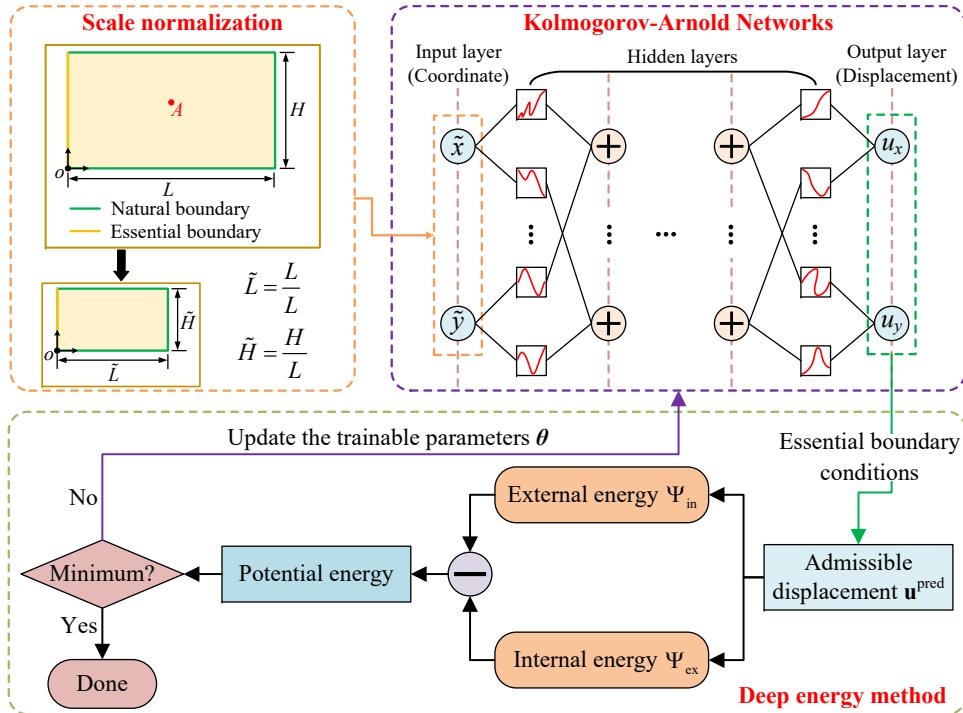

Figure 3: PIKAN schematic: Deep Energy Method using KAN instead of MLP.

As shown in the upper left corner of Fig. 3, we consider a two-dimensional geometric model domain with characteristic length L and characteristic height H, where $L \geq H$, and point o is the coordinate origin. For any point $A(x_A, y_A)$ within the domain, we perform coordinate normalization

$$\tilde{x}_A = \frac{x_A}{L}, \quad \tilde{y}_A = \frac{y_A}{L} \tag{27}$$

19

The normalized coordinates $(\tilde{x}_A, \tilde{y}_A)$ are then used as inputs to the KAN network. This normalization process scales the input coordinates to ensure they are compatible with the preset B-spline grid range (e.g., $[0, 1]$ or $[-1, 1]$), allowing the B-spline activation functions to operate effectively within their defined domains.

We explain the necessity of geometric normalization through a mathematical analysis. For the model in Fig. 3, the left boundary is fixed ($u_x = u_y = 0$). Without coordinate normalization, the admissible displacement field $u_x^{\text{pred}}(A)$ in the x-direction at point $A(x_A, y_A)$ is expressed as

$$u_x^{\text{pred}}(A) = x_A \cdot \mathbf{F}\left(x_A, y_A; \boldsymbol{\theta}\right) \tag{28}$$

During potential energy computation, displacements undergo differentiation with respect to coordinates. Differentiating Eq. (28) with respect to coordinate x yields the normal strain in the x-direction

$$\varepsilon_{xx}^{\text{pred}}(A) = \frac{\partial u_x^{\text{pred}}(A)}{\partial x} = \mathbf{F}\left(x_A, y_A; \boldsymbol{\theta}\right) + x_A \frac{\partial \mathbf{F}\left(x_A, y_A; \boldsymbol{\theta}\right)}{\partial x} \tag{29}$$

The neural network output $\mathbf{F}\left(x_A, y_A; \boldsymbol{\theta}\right)$ is typically bounded within a finite range due to the activation functions used. However, the partial derivative term $\partial \mathbf{F}\left(x_A, y_A; \boldsymbol{\theta}\right) / \partial x$ can vary significantly with the input coordinates. When coordinate x_A becomes large, the second term $x_A \frac{\partial \mathbf{F}}{\partial x}$ in Eq. (29) can become excessively large, causing the strain magnitude to be dominated by the coordinate value rather than the actual physical deformation. This leads to numerical instability and physically unrealistic results.

Introducing geometric normalization reformulates Eq. (28) as

$$u_x^{\text{pred}}(A) = x_A \cdot \mathbf{F}\left(\tilde{x}, \tilde{y}; \boldsymbol{\theta}\right), \tag{30}$$

20

where $\tilde{x} = x_A/L$ and $\tilde{y} = y_A/L$ are the normalized coordinates. The corresponding strain expression becomes

$$\varepsilon_{xx}^{\mathrm{pred}}(A) = \frac{\partial u_x^{\mathrm{pred}}(A)}{\partial x} = \mathbf{F}\left(\tilde{x}, \tilde{y}; \boldsymbol{\theta}\right) + \tilde{x}\frac{\partial \mathbf{F}\left(\tilde{x}, \tilde{y}; \boldsymbol{\theta}\right)}{\partial x}. \tag{31}$$

This normalization properly scales coordinate effects by ensuring that both terms in the strain expression remain within reasonable bounds. The coordinate multiplier $\tilde{x}(= x_A/L)$ in the second term is now normalized, preventing numerical instabilities and improving training convergence. Geometric normalization is therefore essential as it eliminates problematic scaling effects of coordinate derivatives, controls network inputs within a bounded range suitable for the B-spline grid, and enhances neural network fitting performance.

3.2.2. KAN layer output transformation

Since individual KAN layer outputs may exceed the preset B-spline grid range, potentially causing numerical issues in subsequent layers, we apply a tanh activation function to intermediate layer outputs to ensure they remain within the range $[-1, 1]$,

$$x_{l+1,j}^{\mathrm{new}} = \tanh\left(x_{l+1,j}\right) = \tanh\left(\sum_{i=1}^{N_l} m_{j,i} \cdot \phi_{l,j,i}(x_{l,i}) + \sum_{i=1}^{N_l} w_{j,i} \cdot \sigma(x_{l,i})\right) \tag{32}$$

for $j = 1, 2, \ldots, N_{l+1}$ and layers $l = 0, 1, \ldots, L-2$. Note that the final layer $(L-1)$ does not require the tanh activation function, since the output displacement field represents physical quantities with unbounded ranges that should not be artificially constrained.

PIKAN comprises two main components: (i) Neural network component: Based on the above description, once the B-spline grid size, grid range, spline order, and network architecture are determined, we establish a KAN neural

network for predicting displacement fields in elasticity problems. The geometrically normalized two-dimensional coordinates $(\tilde{x}, \tilde{y})$ of sample points within the computational domain and on boundaries serve as inputs, while the corresponding generalized displacement fields (u_x, u_y) serve as outputs. These outputs are used to construct admissible displacements that automatically satisfy essential boundary conditions. (ii) Physics-informed loss function: The physics-informed loss function in PIKAN is the total potential energy of the system computed through the admissible displacement field. By minimizing this potential energy, the trainable parameters in each KAN layer (as shown in Table 1) are optimized. Once neural network training is completed, the model can predict displacement fields that approximate the true solution.

3.3. Sample point distribution strategies

We employ various numerical integration schemes to approximate the energy function, including Monte Carlo, trapezoidal, and Simpson integration. For these schemes, sample points follow uniform distribution throughout the computational domain, maintaining consistent spacing to form a regular grid. Additional boundary points can be incorporated along curved boundaries to better capture domain geometry, as illustrated by the purple points in Figs. 4 and 5.
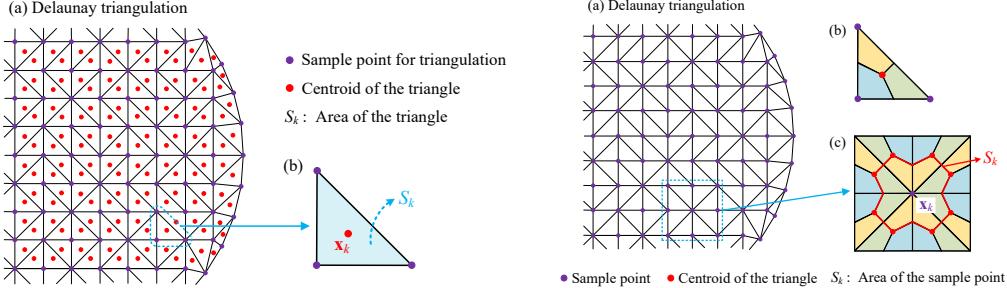
Figure 4: Triangular integration scheme: sample points and control areas.

Figure 5: Delaunay integration scheme: sample points and control areas.

Additionally, we implement two triangular mesh-based integration methods for computing strain energy in the loss function. The first method employs triangular integration rules by subdividing the computational domain into triangular meshes and using triangle centroids as sample points for neural network training. Each triangle's area serves as the control area for its corresponding centroid. The triangular integration formula is

$$\int_{\Omega} F(\mathbf{x}) \mathrm{d}\Omega = \sum_{k=1}^{N} F(\mathbf{x}_k) \cdot S_k \tag{33}$$

where N is the number of triangular elements, $\mathbf{x}_k$ is the centroid of the k-th triangle, S_k is its area, and F is the integrand function.

Fig. 4 illustrates the triangular integration scheme, where purple points represent the original uniform grid, red points are triangle centroids serving as neural network training points, and the blue triangle area in Fig. 4(b) shows the control area for sample point $\mathbf{x}_k$. The mesh density should be chosen based on problem complexity.

The second method, shown in Fig. 5, uses the same integral formula as Eq. (33) but employs the uniform grid points (purple) as training points

rather than centroids. Each triangle is subdivided by connecting its centroid to the midpoints of its edges, creating three equal areas distributed to adjacent training points. For instance, in Fig. 5(b), the green area is assigned to the bottom-right sample point. Each training point's control area comprises one-third of all surrounding triangles. This approach is termed Delaunay integration, where the red-outlined regions in Fig. 5(c) represent the control area S_k for point $\mathbf{x}_k$.

4. Multi-material elasticity problems

Material property discontinuities at interfaces can cause difficulties in learning smooth displacement fields, leading to poor prediction accuracy without properly enforcing interface continuity conditions. Consequently, existing research predominantly employs domain decomposition approaches, where separate neural networks are assigned to different material domains according to material composition. Each neural network independently predicts the displacement field within its respective material domain. Regardless of whether strong-form or energy-form loss functions are used, interface continuity conditions must be incorporated into the loss function to ensure solution accuracy and physical consistency. Consequently, the more materials involved, the more interface penalty terms required in the loss function, significantly increasing training complexity and computational cost. For complex multi-material structures, this approach may result in convergence failure or non-smooth solutions.

Recent work has demonstrated KAN's effectiveness in solving PDEs [28] and heterogeneous problems [36]. Based on spline interpolation theory, B-

splines naturally exhibit piecewise polynomial characteristics with controllable smoothness properties. In multi-material elasticity problems, displacement fields remain continuous across material interfaces while stress fields (and consequently displacement gradients) may exhibit discontinuities due to material property changes. The piecewise nature of B-spline functions makes KAN theoretically well-suited for capturing such solution behaviors without explicit domain decomposition.

Therefore, when solving multi-material problems, this work abandons the traditional domain decomposition approach that assigns separate neural networks to different material regions. Instead, we employ a single KAN network to predict displacement fields across the entire computational domain without requiring subdomain division. We construct a unified energy-based loss function for the complete multi-material model and optimize all trainable KAN parameters simultaneously, obtaining a neural network capable of predicting physically consistent displacement fields that naturally accommodate material property variations.

Fig. 6 illustrates the PIKAN framework for solving multi-material elasticity problems. As shown in the upper left, sample point coordinates are generated throughout different material domains according to geometric features and material distribution. The training point arrangement follows the integration schemes introduced in Section 3.3, with arrangements varying by integration method. Simultaneously, boundary condition points are generated, and all normalized coordinates serve as neural network inputs. It should be noted that since we employ a single KAN network to predict the entire computational domain, no interface continuity penalty terms are re-
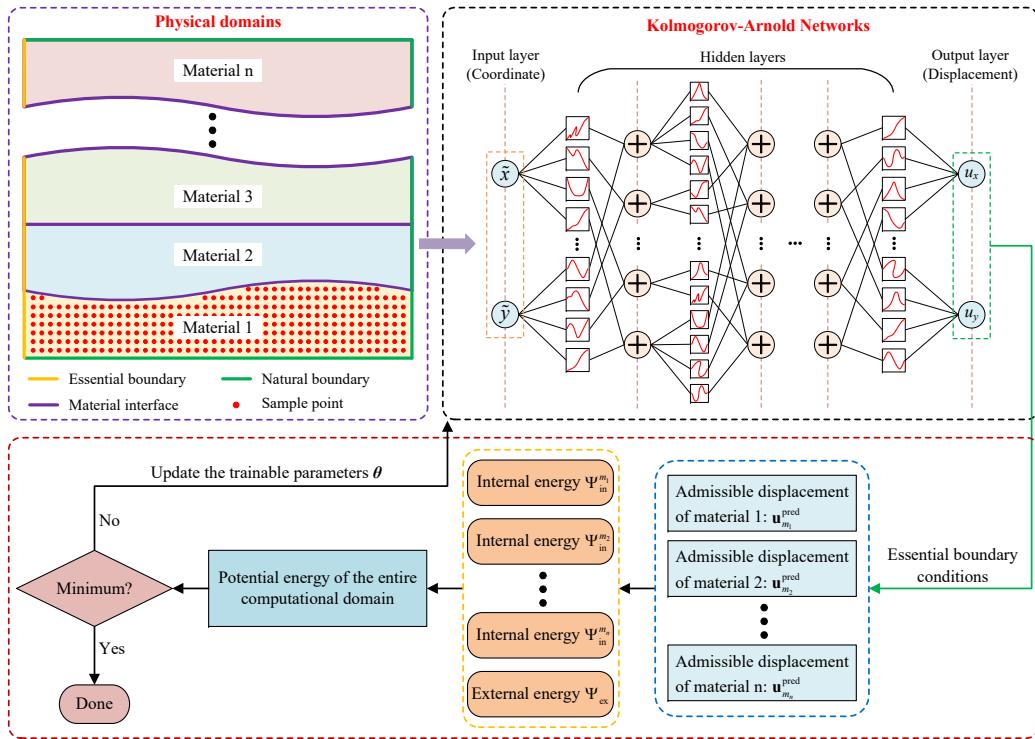
Figure 6: PIKAN flowchart for multi-material elasticity problems.

quired in the loss function, eliminating the need for explicit interface point generation.

A Kolmogorov-Arnold network $\mathbf{F}(\mathbf{x}; \boldsymbol{\theta})$ is constructed to predict displacement fields across all material domains simultaneously. Material properties are incorporated during energy calculation, where strain energy density is computed using the appropriate material constants for each domain based on point locations. This approach constructs a unified admissible displacement field $\mathbf{u}^{\mathrm{pred}}(\mathbf{x})$ that naturally accommodates material discontinuities through KAN's piecewise function characteristics.

The strain energies from all material domains are summed to obtain total strain energy, and subtracting external work yields the potential energy of the entire computational domain. Using the principle of minimum potential energy, an optimizer minimizes the loss function with respect to KAN parameters $\boldsymbol{\theta}$

$$\mathbf{F}(\mathbf{x}; \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{PIKAN}} = \arg\min_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{n} \Psi_{\mathrm{in}}^{m_i} - \Psi_{\mathrm{ex}} \right\} \qquad (34)$$

until convergence is achieved. $\Psi_{\mathrm{in}}^{m_i}$ represents the strain energy in the i-th material domain and Ψ_{ex} is the total external work. Algorithm 1 summarizes the complete PIKAN procedure for multi-material elasticity problems.

5. Numerical examples

This section demonstrates the accuracy of PIKAN through several numerical examples. The first two examples explore PIKAN's application to multi-material problems, while the latter two analyze electronic packaging structures. Unless otherwise specified, all geometric dimensions in the fol-

---

Algorithm 1 PIKAN algorithm for multi-material problems

1: **Input:**

2:     Physical domains of different materials $\Omega_1, \Omega_2, \ldots, \Omega_n$

3:     Boundary conditions: essential boundary Γ_u and natural boundary Γ_σ

4:     Material parameters: $E_1, \nu_1, E_2, \nu_2, \ldots, E_n, \nu_n$

5:     Domain sample points $\mathbf{x}$ from all material domains

6:     Natural boundary condition points $\mathbf{x}_\sigma$ from Γ_σ

7:     KAN architecture and hyperparameters

8:     Neural network optimizer

9: **Output:** Optimized KAN parameters $\boldsymbol{\theta}^*$

10: **Initialize:** KAN parameters $\boldsymbol{\theta} = \{c_p^{(j,i)}, m_{j,i}, w_{j,i}\}$

11: Normalize all sample point coordinates

12: **while** Loss function not converged **do**

13:     Obtain displacement predictions $\mathbf{F}(\mathbf{x})$ from single KAN for all sample points

14:     Construct admissible displacement field $\mathbf{u}^{\mathrm{pred}}(\mathbf{x})$

15:     Compute displacement gradients $\nabla \mathbf{u}^{\mathrm{pred}}$ using automatic differentiation

16:     **for** each material domain $i = 1, 2, \ldots, n$ **do**

17:         Compute strain energy $\Psi_{\mathrm{in}}^{m_i}$ using material parameters (E_i, ν_i)

18:     **end for**

19:     Compute external work potential energy: Ψ_{ex}

20:     Calculate loss function: $\mathcal{L}_{\mathrm{PIKAN}} = \sum_{i=1}^{n} \Psi_{\mathrm{in}}^{m_i} - \Psi_{\mathrm{ex}}$

21:     Update KAN parameters $\boldsymbol{\theta}$ using optimizer and backpropagation

22: **end while**

23: **Return:** Optimized parameters $\boldsymbol{\theta}^*$

---

lowing examples are given in millimeters (mm). To assess accuracy and convergence, we consider three error metrics: absolute error, relative error and relative L_2 norm error. The absolute error is defined as

$$\text{Abs} = |\mathbf{u}_{\text{pred}} - \mathbf{u}_{\text{ref}}| \tag{35}$$

where $\mathbf{u}_{\text{pred}}$ represents the neural network prediction and $\mathbf{u}_{\text{ref}}$ is the reference solution. This metric evaluates prediction accuracy at each sample point. The relative error is computed by

$$\text{Re} = \left| \frac{\mathbf{u}_{\text{pred}} - \mathbf{u}_{\text{ref}}}{\mathbf{u}_{\text{ref}}} \right| \times 100\% \tag{36}$$

The relative L_2 norm error is defined as

$$L_2 = \frac{\sqrt{\sum_{i=1}^{N} \|\mathbf{u}_{\text{pred}}^i - \mathbf{u}_{\text{ref}}^i\|^2}}{\sqrt{\sum_{i=1}^{N} \|\mathbf{u}_{\text{ref}}^i\|^2}} \tag{37}$$

where N represents the total number of sample points. Since analytical solutions are unavailable for these examples, we use FEM results as reference solutions to validate the proposed method's accuracy. In the FEM setup, four-node bilinear quadrilateral elements with reduced integration (CPS4R) were employed, and mesh convergence studies were conducted for all analyses to ensure the validity of the reference solutions.

All examples employ a consistent computational framework. We use the L-BFGS optimizer with initial learning rate 0.001 and constrained to a maximum of 20 iterations per optimization step. The KAN hyperparameters employed in the numerical examples (network depth and width, grid size, and B-spline order) were manually fine-tuned through iterative adjustments

to achieve optimal performance. These settings also reference the foundational configurations reported in [28, 36]. Neural network training was performed on an NVIDIA GeForce RTX 4060 Laptop GPU. Specific network architecture details are provided for each example.

5.1. Cantilever beam analysis across material interfaces

We investigate PIKAN's performance through three cantilever beam bending problems with different material interface geometries, as shown in Fig. 7. Each beam has length $L = 8$ and height $2h = 2$, with the coordinate origin at the bottom-left corner. The straight interface (Fig. 7a) is located at distance $h = 1$ from the bottom edge. The wavy interface (Fig. 7b) consists of two arc segments $\widehat{l}_1$ and $\widehat{l}_2$ that are antisymmetric about the beam's center point $c(4, 1)$. Arc $\widehat{l}_1$ belongs to a circle centered at $(2, -8)$ and arc $\widehat{l}_2$ to a circle centered at $(6, 10)$, both with radius $r = \sqrt{85}$. The stepped interface (Fig. 7c) has geometric parameters $a = 1.2$, $b = 0.8$, $d = 4$, and $e = 0.4$. As shown in Fig. 7, all models are subjected to a shear traction $\bar{T} = 6$ N/mm applied at the right boundary. The left end is fully constrained.

The material parameters are: Material 1 with Young's modulus $E_1 = 8,500$ MPa and Material 2 with $E_2 = 43,000$ MPa, both having Poisson's ratio $\nu = 0.3$. The method uses a KAN structure of $[2, 5, 5, 5, 2]$ with grid size 10, B-spline order 3, and grid range $[0, 1]$, totaling 1050 trainable parameters. The material interfaces are depicted as red dashed lines L_1, L_2, and L_3 in Fig. 7, representing straight, wavy, and stepped configurations for the three cantilever beam models.

Fig. 8 illustrates the distribution of training points used in the analysis of different models. Green points on the right boundary represent natural

(a) Straight interface   (b) Wavy interface

(c) Stepped interface   (d) Homogeneous beam with analysis setup
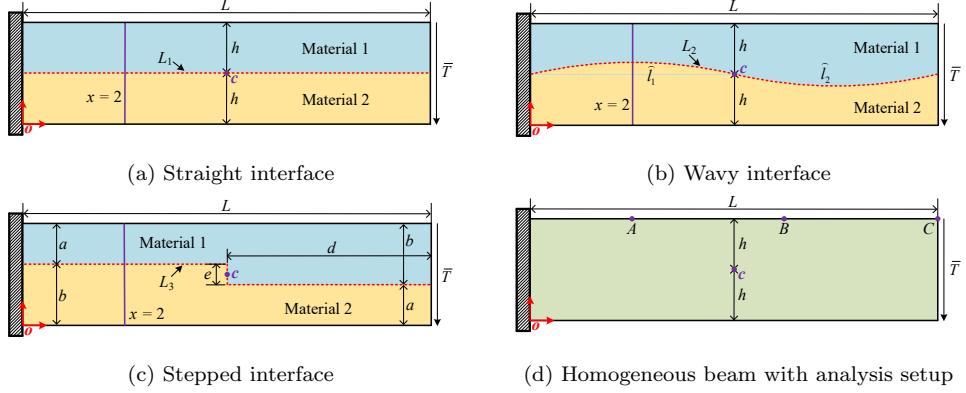
Figure 7: Cantilever beam models with different material interface geometries.

boundary condition points, with 101 points uniformly distributed at spacing $dy = 0.02$ (this boundary point distribution is consistent across all cantilever beam models). For the triangular centroid integration scheme, the interior point counts are: 79,200 points for the straight interface beam (Fig. 8a), 80,278 points for the wavy interface beam (Fig. 8c), and 79,163 points for the stepped interface beam (Fig. 8e). In contrast, the uniform grid schemes shown in Figs. 8b, 8d, and 8f employ identical distribution patterns with 40,501 interior points for each beam configuration.

Fig. 9 shows the evolution of loss functions during training for different numerical integration methods: triangular, Simpson, trapezoidal and Monte Carlo schemes. The triangular scheme uses the triangular point distribution from Fig. 8(a), (c), (e), while Simpson, trapezoidal and Monte Carlo schemes use the uniform distribution from Fig. 8(b), (d), (f). All schemes exhibit convergent behavior, with the triangular integration method generally achieving faster convergence and lower final energy values. Despite different interface geometries, all models demonstrate similar convergence trends with increas-

31

(a) Straight interface - triangular   (b) Straight interface - uniform

(c) Wavy interface - triangular   (d) Wavy interface - uniform

(e) Stepped interface - triangular   (f) Stepped interface - uniform
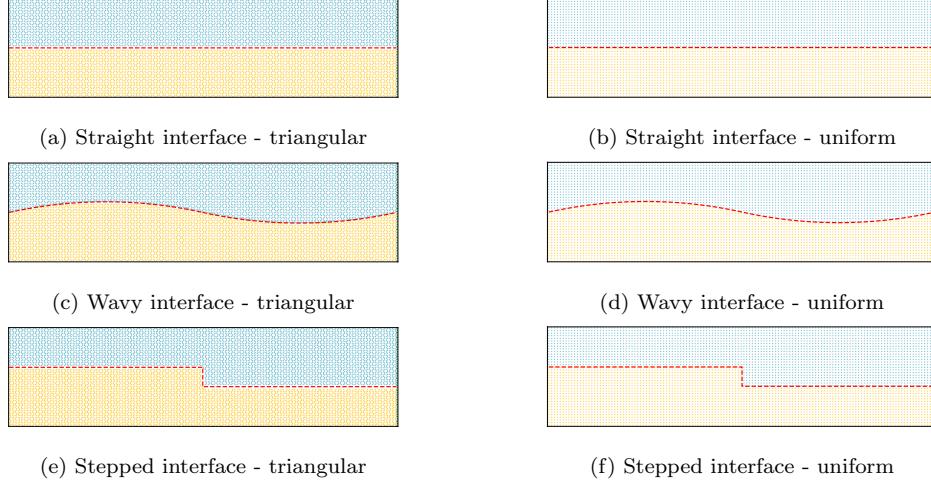
Figure 8: Training point distributions: (a,c,e) triangular scheme; (b,d,f) uniform scheme.

ing training iterations, indicating the robustness of the PIKAN approach across various geometric configurations.
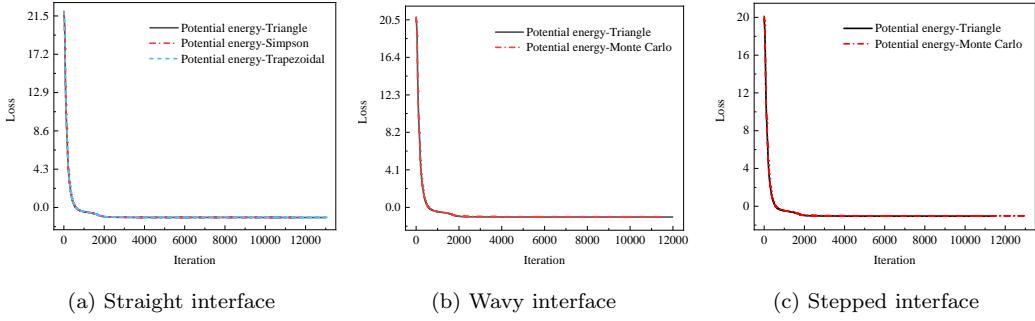

(a) Straight interface   (b) Wavy interface   (c) Stepped interface

Figure 9: Evolution of energy-based loss functions for different interface geometries.

Fig. 10 presents a detailed comparison of displacement fields between PIKAN results (using triangular integration) and FEM reference solutions. PIKAN accurately reproduces both u_x and u_y displacement components across all interface configurations, demonstrating excellent agreement with FEM solutions. The method successfully captures complex displacement

near material interfaces, particularly evident in the wavy and stepped configurations where interface geometry creates significant stress concentrations and discontinuities in material properties.

To further validate PIKAN's computational accuracy, we compare results with the Conservative Energy Neural Network (CENN) [31] method using domain decomposition. CENN applies traditional DEM with separate MLPs for different subdomains, requiring interface continuity conditions that introduce additional penalty terms and hyperparameters in the loss function. The CENN loss function takes the form

$$\mathcal{L}_{\text{CENN}} = \sum_{i=1}^{n} \Psi_{\text{in}}^{i} - \Psi_{\text{ex}} + \beta \sum_{j=1}^{N_{\text{inter}}} \left\| \mathbf{u}_{\text{inter}}^{+}(\mathbf{x}_{\text{inter},j}) - \mathbf{u}_{\text{inter}}^{-}(\mathbf{x}_{\text{inter},j}) \right\|^{2} \qquad (38)$$

Compared to single material domain energy-based loss functions, Eq. (38) includes additional interface continuity penalty terms, where $\mathbf{u}_{\text{inter}}^{+}$ and $\mathbf{u}_{\text{inter}}^{-}$ are predicted displacement fields from different material subnetworks at the interface, and β is the penalty parameter controlling interface condition enforcement. The penalty parameter is calculated using

$$\beta = -\lambda \cdot \ln \left[\tanh \left(N_{\text{inter}}/N_{\text{dom}} \right) \right] \qquad (39)$$

where $\lambda = 1000$, N_{inter} is the number of interface sample points, and N_{dom} is the total number of domain sample points. For comparison purposes, CENN employs identical MLP architectures for different material domains, each consisting of 5 hidden layers with 20 neurons per layer, yielding a total of 3,564 trainable parameters. The optimization employs L-BFGS with a learning rate of 0.001, tanh activation functions, and the same triangular integration scheme for training sample point generation.
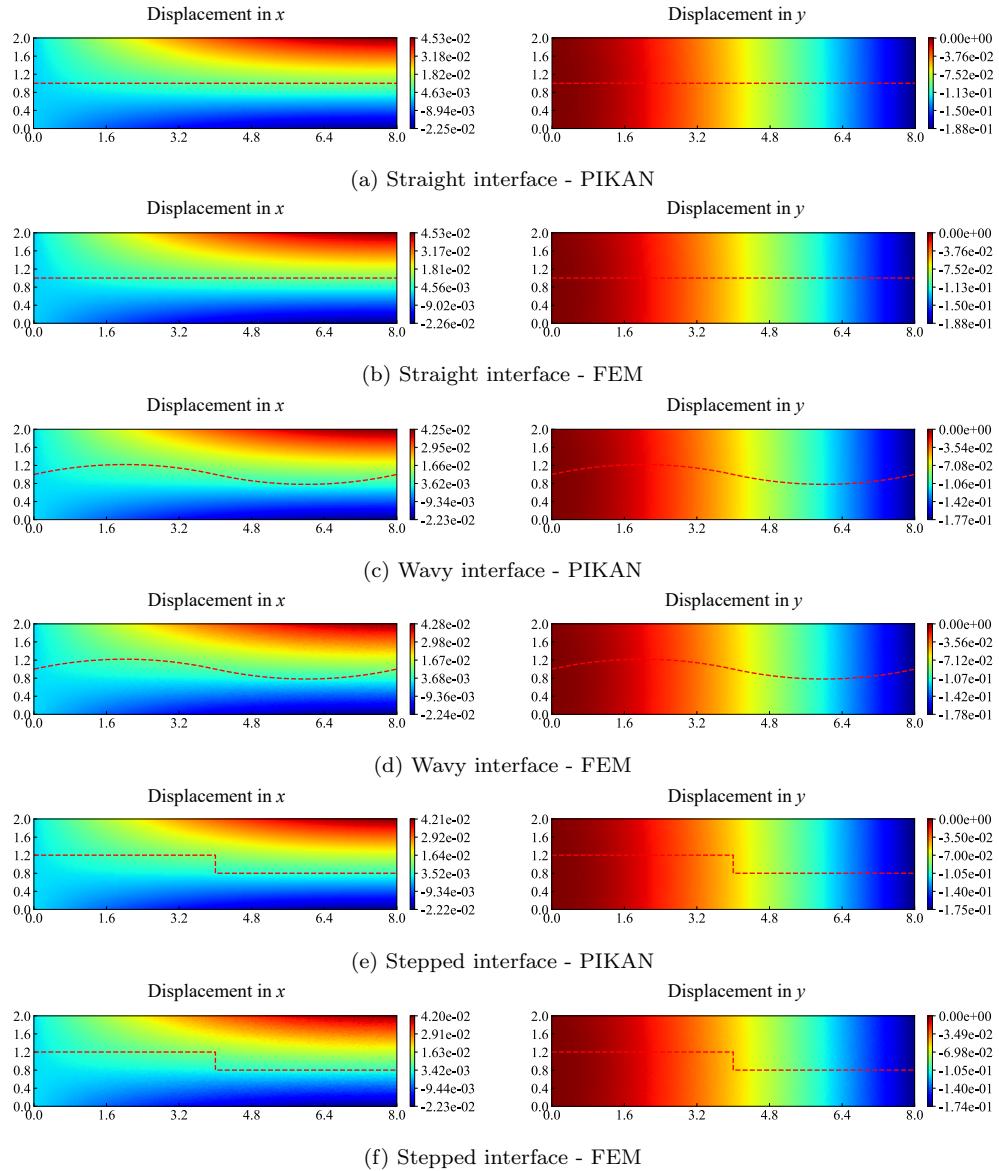
Figure 10: Displacement fields u_x and u_y for different interface geometries: (a), (c), (e) PIKAN predictions; (b), (d), (f) FEM solutions.

34

Fig. 11 compares u_x displacement profiles along the vertical line $x = 6$ for all three cantilever beam models. PIKAN results using different integration schemes are compared with CENN predictions and FEM reference solutions calculated using refined finite element models. The figures demonstrate PIKAN's superior performance across all interface geometries, with results closely matching FEM reference solutions and exhibiting excellent accuracy and stability. Among the integration schemes, triangular, Simpson and trapezoidal methods show the best agreement with FEM solutions, while Monte Carlo integration exhibits slightly reduced precision due to its stochastic nature. In contrast, CENN produces noticeable displacement discontinuities at material interfaces due to its subdomain decomposition approach using independent neural networks for different material regions. Under complex interface geometries (particularly wavy and stepped configurations), CENN exhibits computational instability with significant localized errors, highlighting the inherent challenges of enforcing interface continuity conditions through penalty parameter tuning in domain decomposition methods.
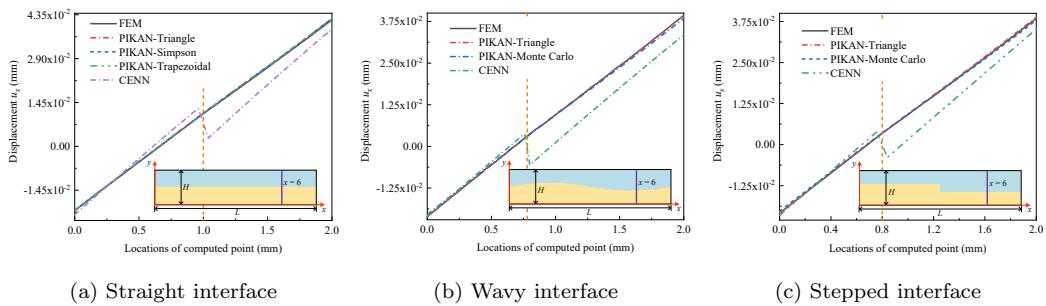

(a) Straight interface   (b) Wavy interface   (c) Stepped interface

Figure 11: Displacement (u_x) at $x = 6$ for different interface geometries: PIKAN vs. CENN and FEM solutions. The orange dashed line denotes the interface position.

We investigate PIKAN's performance in computing von Mises stress for

cantilever beam models by analyzing stress profiles along the vertical line $x = 2$, $0 \leq y \leq 2$ (as shown in Fig. 7). PIKAN results are compared with CENN predictions and FEM reference solutions. Fig. 12 demonstrates that PIKAN predictions exhibit excellent consistency with FEM reference solutions across all interface geometries, accurately capturing stress variations and maintaining smooth stress distributions. As shown in the figure, stress oscillations are observed, which can be attributed to two main factors. First, energy-based methods primarily minimize potential energy through displacement field optimization rather than directly enforcing pointwise stress equilibrium conditions. Second, stress solutions are computed as derivatives of displacement fields combined with constitutive relations, making them highly sensitive to small displacement errors, particularly near boundaries where geometric constraints and material property variations create stress concentrations. Consequently, the numerical accuracy of stress field is inherently lower than that of displacement [37]. Since PIKAN achieves superior displacement accuracy compared to CENN's subdomain approach in multi-material problems, it also yields more accurate stress results. In contrast, CENN exhibits significant stress prediction errors, particularly at material interfaces, due to displacement discontinuities arising from its subdomain decomposition approach. The use of independent neural networks for different material regions results in pronounced stress fluctuations at material boundaries. Under complex interface geometries (wavy and stepped configurations), CENN also demonstrates computational instability in non-interface regions with substantial localized errors, confirming the limitations observed in the displacement analysis.

36

(a) Straight interface   (b) Wavy interface   (c) Stepped interface
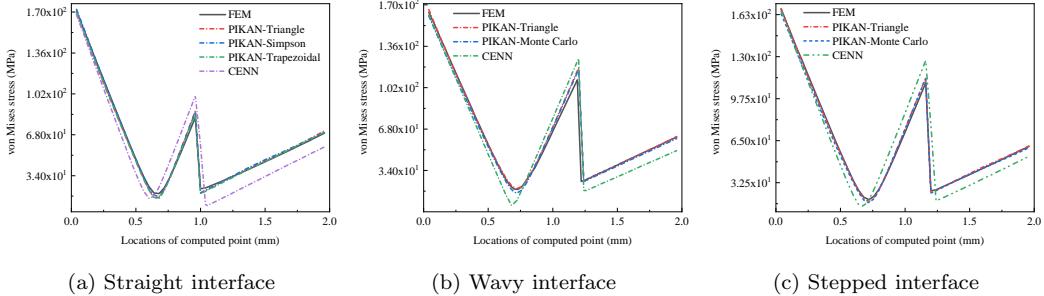
Figure 12: von Mises stress at $x = 2$: PIKAN vs. CENN and FEM solutions.

Hyperparameter selection critically affects PINN computational results. This section investigates how different hyperparameter combinations impact PIKAN prediction accuracy using the straight interface cantilever beam model from Fig. 7a. We analyze the relative L_2 norm error variation for displacement component u_y under different KAN configurations, where each configuration is defined by network depth and width, grid size, and B-spline order. Fig. 13 presents the hyperparameter sensitivity analysis results. Fig. 13a shows the L_2 norm error of u_y with B-spline order fixed at 3 for three different KAN architectures: [2,5,5,2], [2,5,5,5,2], and [2,5,5,5,5,2]. The results indicate that increasing grid size does not produce consistent accuracy improvements across different network depths, suggesting that optimal grid size depends on the specific network architecture. Fig. 13b demonstrates the effect of varying B-spline order on computational accuracy when grid size is fixed at 10. The results show that the optimal order varies with network architecture. These findings highlight the importance of joint optimization of KAN hyperparameters rather than individual parameter tuning.

The analysis reveals complex relationships between KAN hyperparameters and computational accuracy. While smaller grid sizes and B-spline
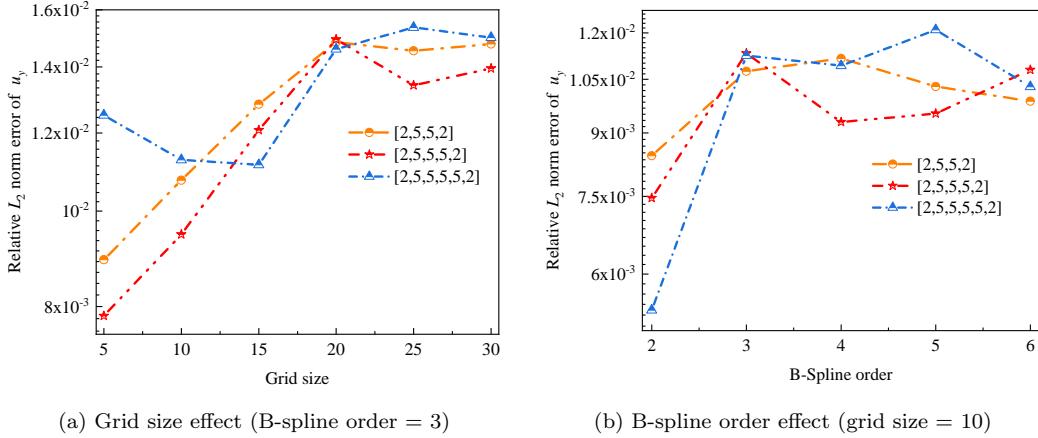
(a) Grid size effect (B-spline order = 3)

(b) B-spline order effect (grid size = 10)

Figure 13: Relative L_2 norm error of u_y for different KAN hyperparameter combinations.

orders often yield better performance in these specific examples, the optimal hyperparameter combination varies significantly with network architecture and problem complexity. This sensitivity occurs because non-optimal hyperparameter selections can lead to either underfitting (insufficient model capacity to capture solution complexity) or overfitting (excessive model complexity leading to poor generalization), consistent with findings reported in machine learning literature [28]. The non-monotonic relationship between hyperparameters and accuracy highlights a fundamental challenge in KAN optimization: larger grid sizes or higher B-spline orders do not automatically improve performance. Instead, these parameters must be carefully balanced with network depth and width to achieve optimal results. Currently, effective methods for global optimization of KAN hyperparameter combinations are lacking due to the vast search space and extensive computational requirements for systematic exploration. Future work should focus on developing efficient hyperparameter optimization strategies specifically tailored for

physics-informed KAN applications.

Finally, we compare PIKAN (with grid size 20 and spline order $k = 3$) with DEM for the problem shown in Fig. 7d. The displacement u_x can be obtained analytically from [38]

$$u_x = \frac{T(y - h)}{6EI_z} \left((6L - 3x)x + (2 + \nu) \left(y^2 - 2hy \right) \right) \tag{40}$$

where $T = 12$ N is the applied traction, Young's modulus $E = 15,000$ MPa, Poisson's ratio $\nu = 0.3$, and $I_z = (2h)^3/12$ is the rectangular cross-sectional moment of inertia. The geometric dimensions are consistent with those described previously. Table 2 presents the computational results and efficiency comparison between PIKAN and DEM. Specifically, the displacement u_x is compared at three points A, B, and C (shown in Fig. 7d). Using trained neural networks to predict on computational domains with identical sample point density, PIKAN achieves significantly superior accuracy compared to DEM with fewer trainable parameters and similar prediction efficiency, closely matching the analytical solution. It should be noted that PIKAN exhibits lower training efficiency, primarily due to the higher computational complexity of KAN's learnable B-spline activation functions compared to MLPs' fixed activation functions. During forward propagation, each KAN neuron dynamically computes spline basis functions and weighted summations. During backpropagation, gradients with respect to both spline coefficients and inputs must be calculated. Although requiring fewer parameters, each parameter incurs higher computational costs. Additionally, current code implementations have not been fully optimized.

Table 2: Comparison of PIKAN and DEM results with analytical solution

| Methods | Displacement u_x (mm) | | | Relative error | | | Neural Network Architecture | Number of Parameters | Time (s) | |
	A	B	C	A	B	C			Training/1000 iter.	Prediction
Analytical	0.0168	0.033	0.0384	-	-	-	-	-	-	-
PIKAN	0.01681	0.03302	0.03859	0.06%	0.061%	0.495%	[2,5,5,5,2]	1750	334.82	0.242
DEM	0.01596	0.03245	0.03736	5%	1.667%	2.708%	[2,30,30,30,30,2]	2942	39.83	0.254

5.2. Heterogeneous plate with central hole

We apply PIKAN to analyze a multi-material square plate with a central hole under tensile loading, as shown in Fig. 14(a). Due to symmetry, only a quarter-plate model with dimensions $L \times L$, where $L = 21$, is analyzed. The coordinate origin o is located at the bottom-left corner of the quarter model. As illustrated in Fig. 14(b), the central circular hole has radius $R_1 = 5$ with traction-free boundaries. The material interface (indicated by the red dashed line) is a circular boundary with radius $R_2 = 13$, where material 1 occupies the inner region and material 2 occupies the outer region. A uniform tensile load $\bar{T} = 120$ N/mm is applied to the right boundary of the quarter model. Symmetry boundary conditions are imposed along the left and bottom edges. The heterogeneous plate consists of two materials: Material 1 with Young's modulus $E_1 = 10,000$ MPa and Poisson's ratio $\nu_1 = 0.3$, and Material 2 with $E_2 = 1,000$ MPa and $\nu_2 = 0.4$.

The KAN architecture for this model is configured as $[2, 5, 5, 5, 2]$ with grid size 20, B-spline order $k = 3$, and grid range $[0, 1]$. The neural network contains 1,750 trainable parameters. Fig. 15 shows the training point distributions for different integration schemes applied to the multi-material plate model. Purple points represent sample points in Material 1 (inner region), maroon points represent sample points in Material 2 (outer region), and green
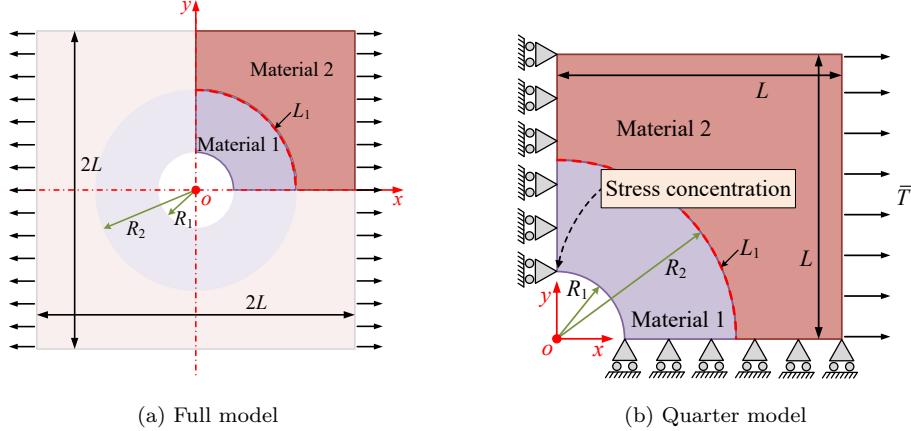
(a) Full model   (b) Quarter model

Figure 14: Heterogeneous plate with central hole model.

points along the loaded boundaries represent natural boundary conditions, totaling 211 points. The triangular integration scheme uses 22,786 interior points in Material 1 and 61,624 interior points in Material 2. For uniform grid-based integration schemes (Monte Carlo and Delaunay), Material 1 contains 11,574 interior points and Material 2 contains 31,121 interior points. This difference in point density allows for investigating the effect of sampling resolution on solution accuracy. For comparison, results from CENN and DEM are also presented. In CENN, each material domain employs a separate MLP with 6 hidden layers of 30 neurons each, totaling 4,802 trainable parameters. DEM uses a single network without subdomain partitioning, requiring a more complex MLP architecture with 7 hidden layers of 40 neurons each, totaling 10,042 trainable parameters. The training sample points follow the triangular distribution pattern shown in Fig. 15(a).

Fig. 16 demonstrates the evolution of energy-based loss functions during training for the heterogeneous plate model. As the neural network undergoes

(a) Triangular integration scheme   (b) Uniform grid integration scheme
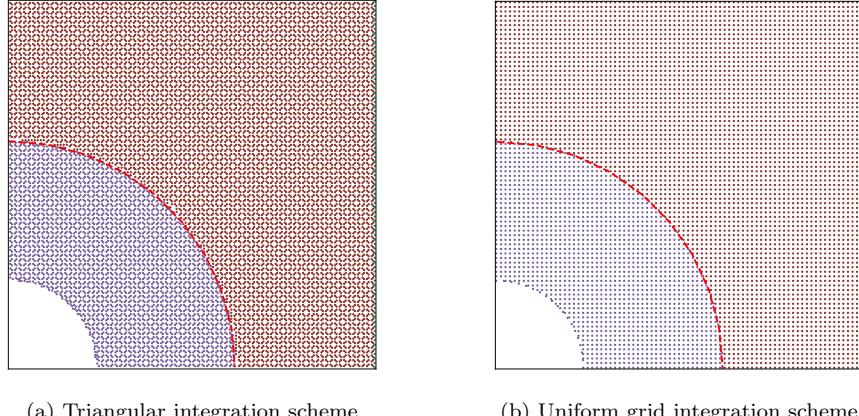
Figure 15: Sample point distribution for heterogeneous plate model.

iterative optimization, the total potential energy gradually converges to its minimum value, validating the principle of minimum potential energy. All numerical integration schemes exhibit similar convergence behavior, but the Delaunay scheme exhibits marginally weaker convergence in the loss compared to the other two schemes. Fig. 17 presents a detailed comparison between PIKAN predictions (using triangular integration) and FEM reference solutions for displacement magnitude u_{mag} (calculated according to Eq. (41)).

$$u_{\mathrm{mag}} = \sqrt{u_x^2 + u_y^2} \tag{41}$$

The results demonstrate excellent agreement between PIKAN and FEM solutions, with PIKAN successfully capturing the stress concentration around the central hole and the smooth transition across the material interface. The displacement patterns clearly show the influence of material property differences, with higher displacements occurring in the softer outer material (Material 2). Fig. 18 shows the pointwise absolute error contours for displacement components u_x and u_y compared to FEM reference solutions. The

42

error distribution reveals that PIKAN maintains high accuracy throughout most of the domain, with maximum errors typically below 9.57×10^{-3}.
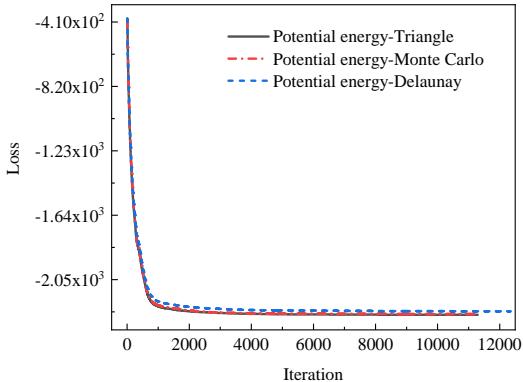

Figure 16: Loss function evolution during training for heterogeneous plate model.
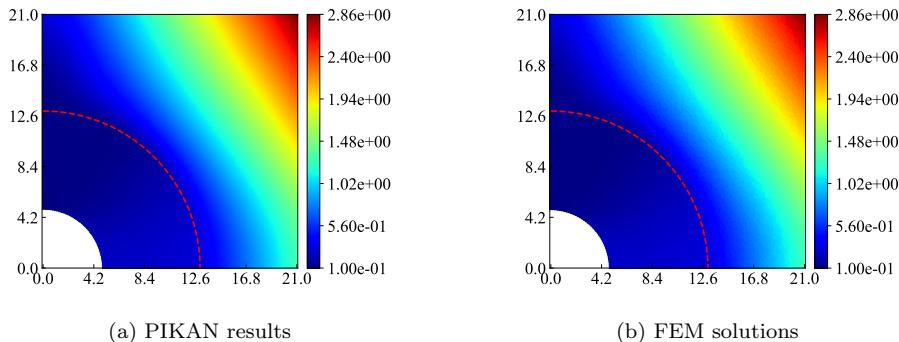

(a) PIKAN results

(b) FEM solutions

Figure 17: Displacement u_{mag} for heterogeneous plate model: PIKAN vs. FEM.

Fig. 19 compares displacement components u_x and u_y at the circular material interface. The interface analysis reveals the effectiveness of PIKAN in handling material property discontinuities. Additionally, to examine solution behavior across the entire domain, displacement profiles along the diagonal line $\theta = 45°$ (from the hole edge to the outer boundary, indicated by the red line) are analyzed.

43

(a) u_x absolute error (b) u_y absolute error
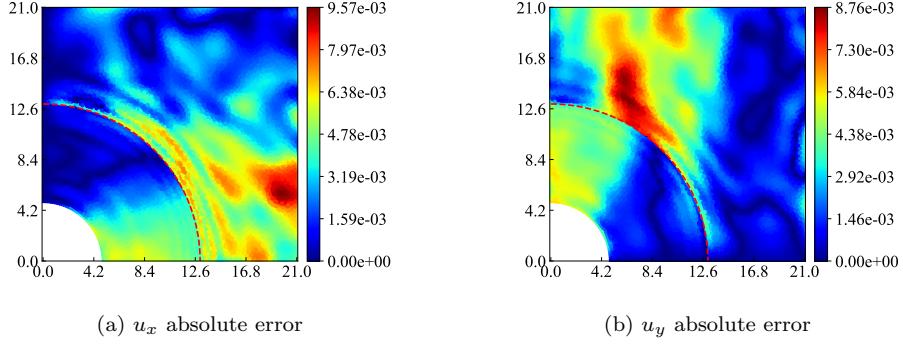
Figure 18: Absolute error contours for displacement fields in heterogeneous plate model.

Fig. 20 shows the comparison of displacement magnitude u_{mag} along this diagonal line. The results demonstrate that triangular integration and Monte Carlo integration schemes produce solutions that agree well with the FEM reference, maintaining accuracy across both material regions and the interface transition. However, the Delaunay-based integration scheme demonstrates slightly inferior performance near the central hole, likely due to suboptimal control area distribution around the circular boundary, indicating that integration scheme performance is sensitive to local geometric complexity. In contrast, CENN and DEM predictions show significant deviations from the reference solution throughout the domain, failing to accurately capture the displacement field distribution patterns. This demonstrates the inherent challenges of traditional DEM and domain decomposition approaches that rely on penalty terms for interface continuity, particularly for problems with circular interfaces where geometric complexity compounds convergence difficulties.
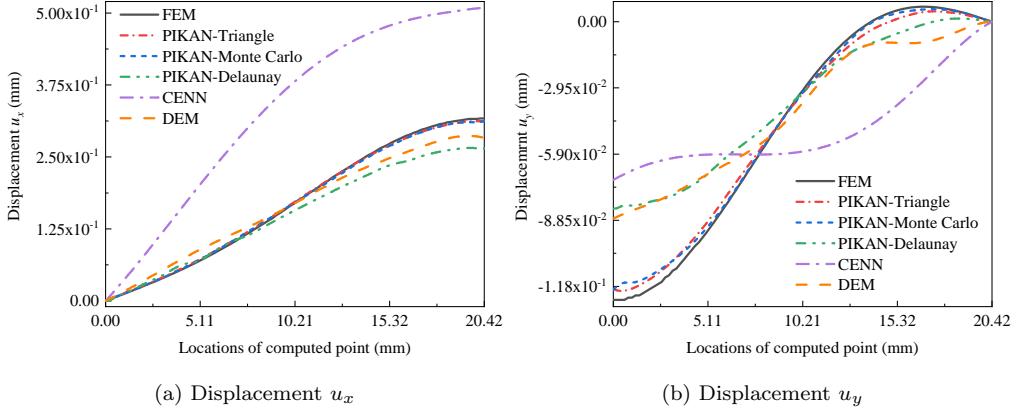
44

(a) Displacement u_x

(b) Displacement u_y

Figure 19: Displacement fields at the material interface for the heterogeneous plate model.
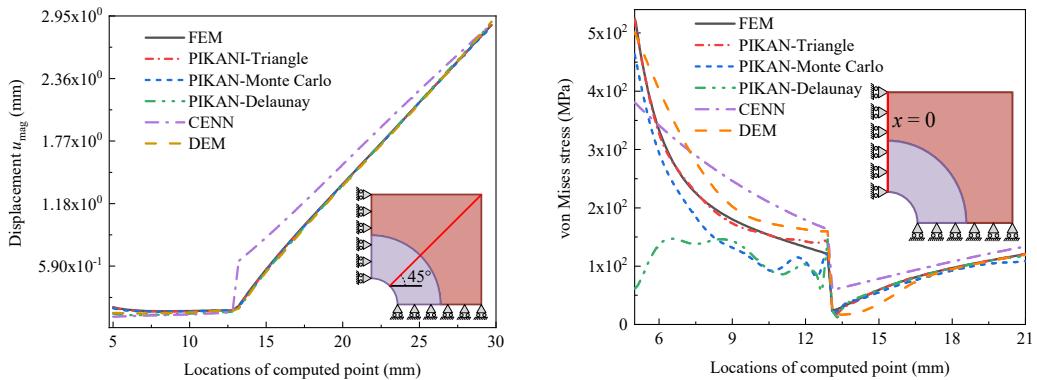

Figure 20: u_{mag} along diagonal line.

Figure 21: Stress along the line $x = 0$.

Next, we evaluate PIKAN's performance by comparing stress results along the vertical line $x = 0$ (from $y = 5$ to $y = 21$) between PIKAN and FEM solutions. This analysis line extends from the hole boundary through both material regions to the outer boundary, providing insight into stress prediction accuracy across material interfaces and high-gradient regions. The results are presented in Fig. 21. The analysis reveals that triangular integration achieves the highest accuracy, closely converging to the FEM reference

45

solution throughout the domain. In contrast, Monte Carlo and Delaunay integration schemes exhibit poor prediction performance in the Material 1 domain (inner region with higher stiffness), where stress concentrations are more pronounced, but show better agreement with the reference solution in the Material 2 domain (outer region with lower stiffness). This differential performance suggests that stress calculations are particularly sensitive to integration scheme choice in regions with steep stress gradients and high material property contrasts. CENN and DEM predictions follow the general trend of the reference solution but with substantially larger errors, highlighting the challenges of maintaining stress accuracy in traditional DEM and domain decomposition approaches that rely on penalty terms for interface continuity.

5.3. PIKAN application to DBC substrate structure

PIKAN is applied to analyze DBC (Direct Bonding Copper) substrates featuring multi-material domains. DBC substrates are widely used in power electronics, high-power modules, and aerospace applications due to their excellent thermal conductivity and mechanical strength. Material property differences and thermal/mechanical stresses during manufacturing or operation cause these substrates to warp, affecting assembly performance and compromising equipment functionality and reliability [5].

A simplified DBC substrate warping problem is modeled as shown in Fig. 22(a). The substrate has total length $L = 8$ with an Al_2O_3 ceramic core of thickness $b = 0.48$ sandwiched between upper and lower copper layers, each with thickness $a = 0.26$. The substrate experiences equal and opposite bending moments $M = 100$ N·mm at both ends, creating pure bending

46

deformation. Due to symmetry about the midplane, only the right half of the model requires analysis, as shown in Fig. 22(b). The coordinate origin is located at the center of the substrate cross-section. Symmetry boundary conditions are applied at the left boundary, while the bending moment is applied as an equivalent distributed force at the right end. The equivalent force distribution, illustrated in Fig. 22(c), is given by

$$f_M(y) = \frac{12M}{h^3}y = 1200y \quad \text{for } y \in (-0.5, 0.5) \tag{42}$$

where $h = 1.0$ is the total substrate thickness. This linear distribution ensures that the resultant force is zero and the moment equals the applied bending moment M. The DBC substrate consists of copper layers with Young's modulus $E_{\text{Cu}} = 128,000$ MPa and Poisson's ratio $\nu_{\text{Cu}} = 0.34$, and an alumina ceramic core with $E_{\text{Al}_2\text{O}_3} = 270,000$ MPa and $\nu_{\text{Al}_2\text{O}_3} = 0.28$.
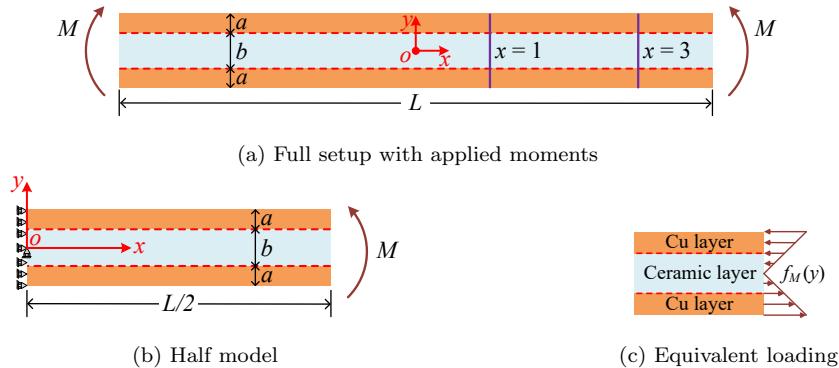

(a) Full setup with applied moments

(b) Half model

(c) Equivalent loading

Figure 22: DBC substrate bending problem.

A KAN with architecture $[2, 5, 5, 5, 2]$ is employed to predict the displacement field (u_x, u_y) with grid size 10, B-spline order $k = 2$, and grid range $[-1, 1]$. The neural network contains 980 trainable parameters. Fig. 23 shows the training point distribution schemes for the DBC substrate model. Due

to the layered structure of the DBC substrate, sample points are distributed across three material regions: the upper copper layer, the ceramic core, and the lower copper layer. In the triangular distribution scheme, the two copper layers contain a total of 166,400 sample points (83,200 points per layer) and the ceramic core contains 150,400 sample points. In the uniform grid distribution scheme, the copper layers contain a total of 84,906 sample points (42,453 points per layer) and the ceramic core contains 76,095 sample points. For both distribution schemes, the right-end natural boundary condition is discretized using 201 uniformly distributed sample points to accurately represent the linear force distribution given by Eq. (42).

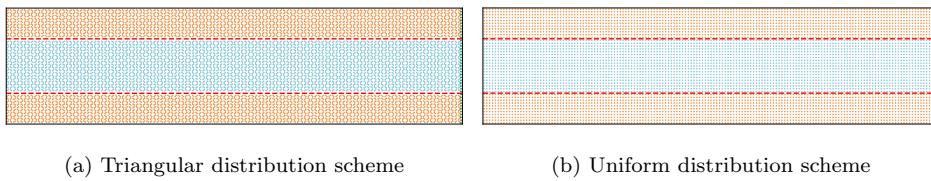(a) Triangular distribution scheme   (b) Uniform distribution scheme

Figure 23: Training point distribution schemes for DBC substrate model.

Fig. 24 illustrates the evolution of the potential energy loss function for the DBC substrate warpage model. Fig. 25 presents a comprehensive comparison of displacement field solutions between PIKAN predictions (using simpson integration) and FEM reference solutions for the DBC substrate bending problem. The results demonstrate excellent agreement between PIKAN and FEM across all displacement components. The horizontal displacement u_x (Figs. 25a and 25b) shows the expected symmetric distribution about the neutral surface, with the upper and lower surfaces deflecting in opposite directions. For the vertical displacement u_y (Figs. 25c and 25d), both methods capture the characteristic bending deformation pattern with maximum dis-

placements occurring at the loaded end. The displacement magnitude u_{mag} (Figs. 25e and 25f) provides a comprehensive view of the overall deformation. Fig. 26 quantifies the prediction accuracy through absolute error contours for both displacement components. The error analysis reveals that PIKAN maintains high accuracy throughout the domain, with maximum absolute errors typically below 6.6×10^{-4} for both u_x and u_y components. The error distribution shows slightly higher values near the material interfaces and the loaded boundary, which is expected due to the discontinuities in material properties and the complex stress distributions in these regions. Overall, the error magnitudes are several orders smaller than the displacement values, confirming the excellent accuracy of the PIKAN approach for multi-material structural analysis.
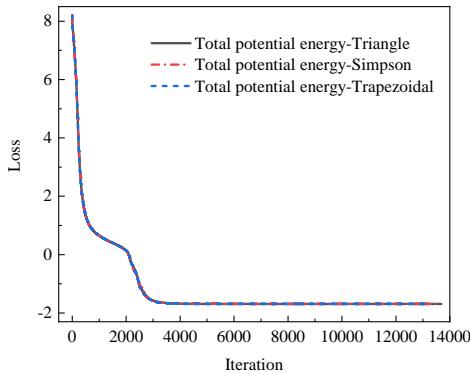

Figure 24: Evolution of loss functions during training for the DBC substrate model.

To further demonstrate the effectiveness of PIKAN for multi-material problems, we analyze displacement distributions at critical locations within the DBC substrate. The material interfaces L_1 and L_2 represent the upper and lower boundaries between the copper layers and the ceramic core, respectively. These interfaces are critical for understanding stress transfer and po-
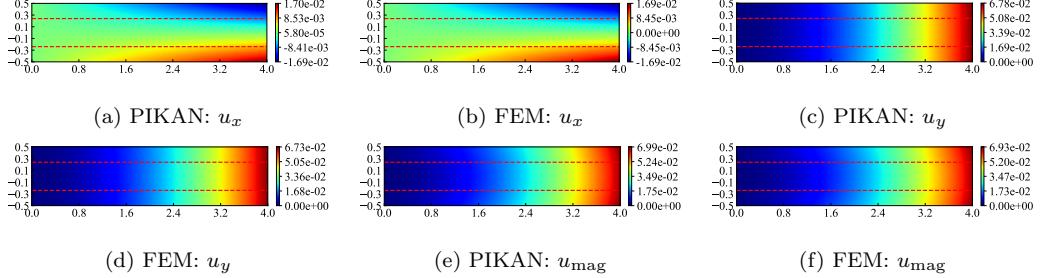
(a) PIKAN: u_x (b) FEM: u_x (c) PIKAN: u_y

(d) FEM: u_y (e) PIKAN: u_{mag} (f) FEM: u_{mag}

Figure 25: Displacement field solutions for DBC substrate model: PIKAN vs. FEM.

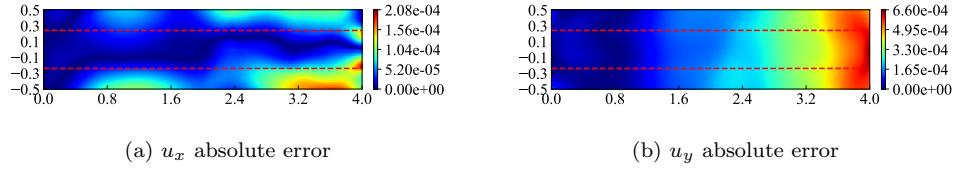(a) u_x absolute error   (b) u_y absolute error

Figure 26: Absolute error contours for displacement fields in DBC substrate model.

tential delamination sites in DBC substrates. Fig. 27 presents displacement distributions at different locations. Fig. 27(a) shows the horizontal displacement u_x along the upper material interface L_1 (copper-ceramic boundary), where the displacement varies linearly due to the pure bending deformation. Fig. 27(b) presents the displacement magnitude u_{mag} along the lower material interface L_2, Fig. 27(c) shows the vertical displacement u_y along the neutral surface (line $y = 0$, where $0 \leq x \leq 4$) within the ceramic layer. The results demonstrate that PIKAN solutions achieve excellent agreement with FEM reference solutions across all analyzed locations. All numerical integration schemes exhibit comparable accuracy levels, indicating the robustness of the PIKAN approach. The smooth displacement distributions at material interfaces confirm that PIKAN effectively handles multi-material problems without requiring explicit interface continuity constraints.

50

(a) u_x along upper interface L_1 (b) u_{mag} along lower interface L_2 (c) u_y along neutral axis $(y = 0)$
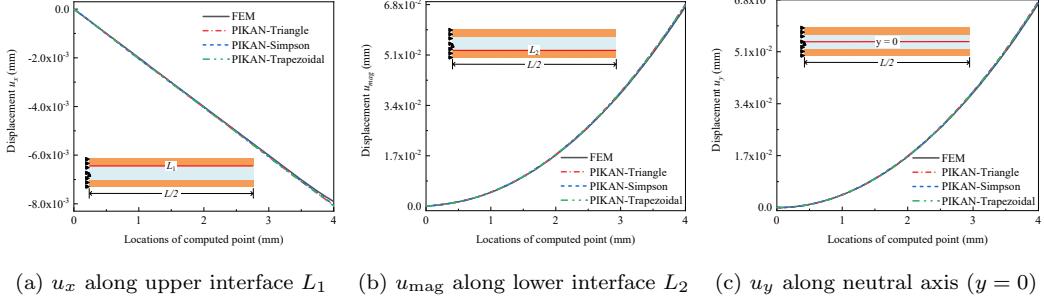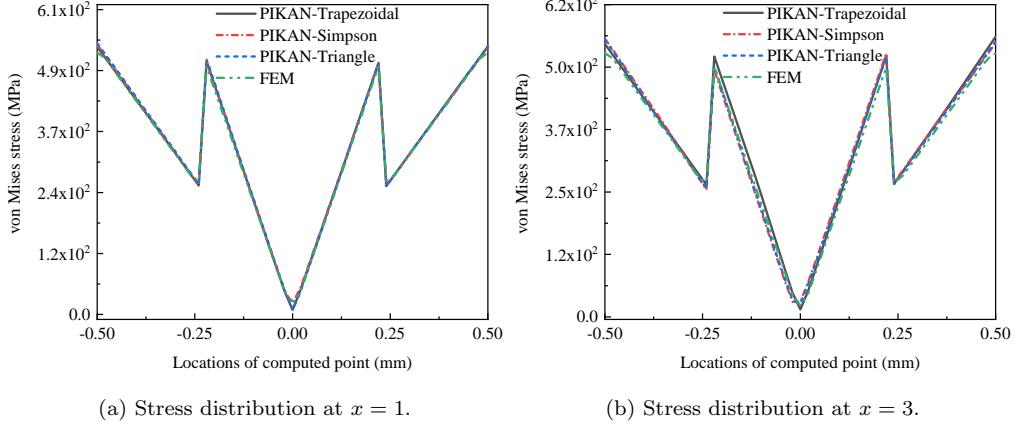
Figure 27: Displacement distributions at critical locations in DBC substrate.

Figs. 28(a) and (b) show stress distributions along lines $x = 1$ and $x = 3$ (where $-0.5 \leq y \leq 0.5$, as shown in Fig. 22) in the DBC substrate. These lines capture stress variations in the interior region and near the loaded end. PIKAN stress results agree well with FEM reference solutions at both locations. The stress distributions show typical multi-material behavior with stress jumps at material interfaces $(y = \pm 0.24)$ due to different elastic moduli. Significant stress variations occur in the stiffer ceramic core, while copper layers show lower stress levels. Among these integration schemes, all computational results demonstrate similar accuracy and converge well to the FEM solutions. This indicates that PIKAN can effectively capture stress variations near material interfaces even without employing higher-order polynomial approximation methods (such as Simpson's rule), confirming the effectiveness and robustness of PIKAN for solving multi-material problems.

5.4. PIKAN application to TGV-Cu structure

Glass-substrate interposer packaging enables high-density interconnection of multi-chip systems through Through Glass Via (TGV) technology. TGV structures provide mechanical support and electrical interconnection via elec-

51

(a) Stress distribution at $x = 1$.

(b) Stress distribution at $x = 3$.

Figure 28: Stress distributions along vertical lines in the DBC substrate.

(a) Single TGV-Cu structure

(b) Interface roughness type 1 ($R_1 = 0.1$)

(c) Interface roughness type 2 ($R_2 = 0.05$)
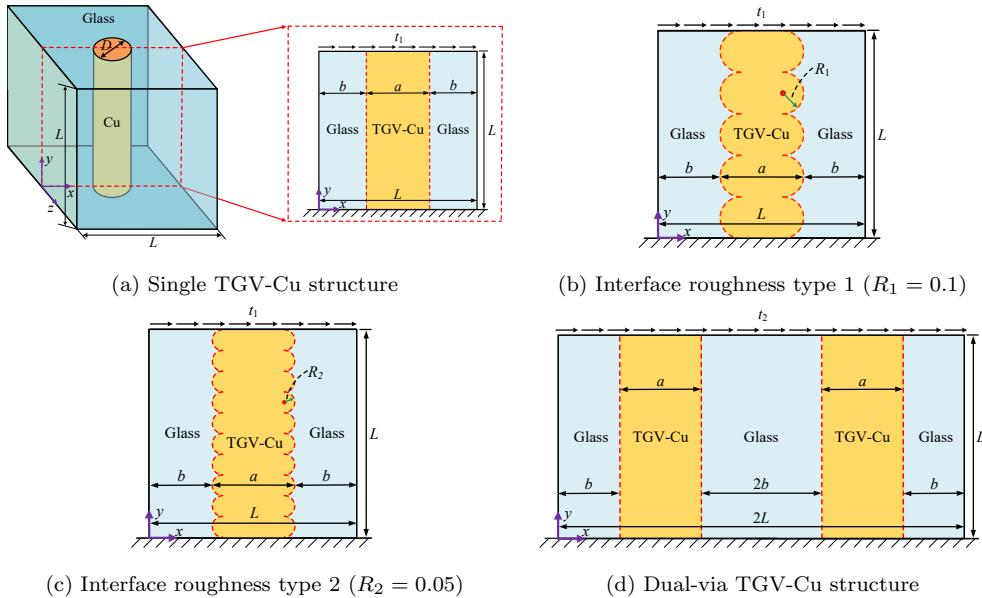
(d) Dual-via TGV-Cu structure

Figure 29: TGV-Cu structure models for CMP analysis: (a) baseline single via; (b,c) single via with interface roughness variations; (d) dual-via configuration.

troplated copper pillars fabricated through electroplating processes. After electroplating, residual copper layers on the glass surface must be removed through chemical mechanical polishing (CMP) to achieve smooth interfaces and proper electrical isolation [39]. PIKAN is employed to analyze TGV-Cu structures during the CMP process, as shown in Fig. 29. Four model configurations are investigated: (1) single TGV-Cu structure (Fig. 29(a)); (2) and (3) TGV-Cu structures with different interface roughness patterns (Figs. 29(b) and (c)); and (4) dual-via TGV-Cu structure (Fig. 29(d)).

The geometric dimensions are $L = 1$, $a = 0.4$, and $b = 0.3$. Interface roughness is modeled as semicircular protrusions with radii $R_1 = 0.1$ and $R_2 = 0.05$ at the glass-copper interface. The dual-via configuration features two copper vias with center-to-center spacing of 0.6. Boundary conditions simulate the CMP process: the bottom surface is fixed, while horizontal forces is applied at the top surface. The applied forces are $t_1 = 200$ N/mm for single-via structures and $t_2 = 250$ N/mm for the dual-via structure. Material properties are listed in Table 3.

Table 3: Material parameters for TGV-Cu models

Material	Young's Modulus E (MPa)	Poisson's Ratio ν
Glass substrate	77,000	0.24
Cu (electroplated)	150,000	0.3

The KAN architecture for this problem maintains consistency with previous examples: $[2, 5, 5, 5, 2]$ structure with grid size 20, B-spline order 3, and 1,750 trainable parameters. Fig. 30 shows the training sample point distribution schemes for all TGV-Cu structure models. For triangular distribution,

the domain point counts are: single structure - 79,200 points; roughness type 1 - 80,094 points; roughness type 2 - 80,064 points; and dual-via structure - 158,400 points. For uniform grid integration schemes, the distributions are: single structure - 40,401 points; both roughness models - 40,761 points each; and dual-via structure - 80,601 points. Natural boundary condition training points are distributed as 201 points for single TGV-Cu structures and 401 points for the dual-via structure, corresponding to the different boundary lengths requiring load application.

(a) Single TGV: triangular   (b) Single TGV: uniform   (c) Roughness 1: triangular

(d) Roughness 1: uniform   (e) Roughness 2: triangular   (f) Roughness 2: uniform
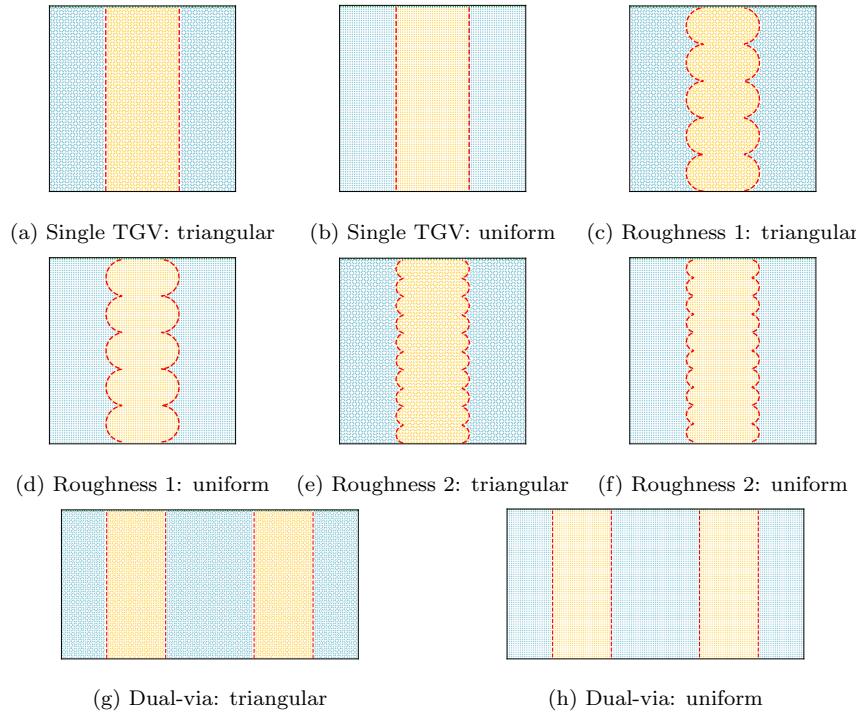
(g) Dual-via: triangular   (h) Dual-via: uniform

Figure 30: Training point distribution schemes for TGV-Cu structure models: (a-f) single via configurations with different integration schemes; (g-h) dual-via configurations.

Fig. 31 shows the evolution of loss functions for different TGV-Cu structure models during neural network training. Despite varying geometric com-

plexities and structural arrangements, all models exhibit similar convergence behavior with consistent energy reduction trends. The convergence patterns demonstrate PIKAN's robustness across different geometric configurations. All integration schemes achieve stable convergence, with triangular integration typically showing slightly faster convergence.

(a) Single TGV-Cu structure

(b) Interface roughness type 1

(c) Interface roughness type 2
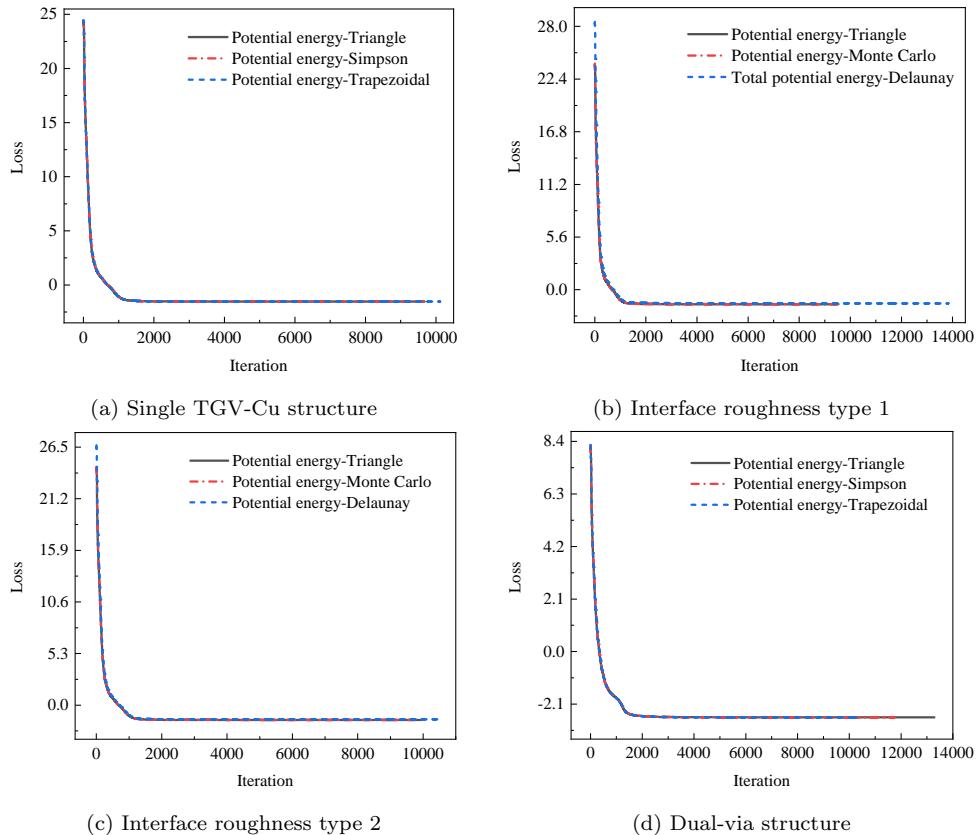
(d) Dual-via structure

Figure 31: Loss function evolution for different TGV-Cu structure configurations.

Fig. 32 compares PIKAN results (using triangular integration) with FEM solutions for the single-via TGV-Cu structure with smooth interfaces. The displacement field distribution patterns predicted by PIKAN show excellent

agreement with FEM solutions across u_x and u_y. PIKAN successfully captures the deformation patterns in both glass and copper materials. Fig. 33 presents the comparison between PIKAN results (using trapezoidal integration) and FEM solutions for the dual-via TGV-Cu structure. Despite the increased geometric complexity with multiple material interfaces and via interactions, PIKAN maintains high accuracy in predicting displacement fields. The method effectively captures the coupling effects between adjacent vias and the displacement redistribution in the multi-via configuration.
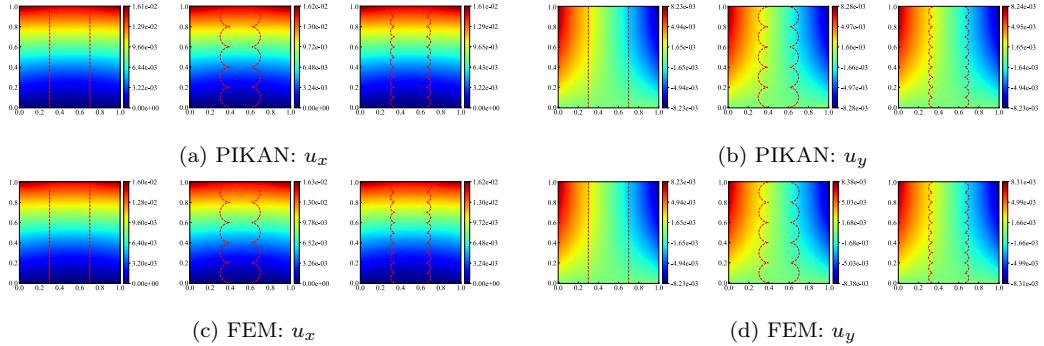

(a) PIKAN: u_x (b) PIKAN: u_y

(c) FEM: u_x (d) FEM: u_y

Figure 32: Displacement comparison for single-via TGV-Cu: PIKAN vs. FEM solutions.

To evaluate PIKAN's performance at critical material interfaces, displacement field u_x values are extracted along the glass-copper interfaces for all TGV-Cu configurations and compared with FEM reference solutions, as shown in Fig. 34. These interface analyses are particularly important for understanding stress transfer mechanisms and potential failure locations in TGV structures. The comparison results demonstrate that PIKAN maintains excellent agreement with reference solutions across all geometric configurations, from simple smooth interfaces to complex roughness patterns and multi-via arrangements. Triangular, Monte Carlo, Simpson, and trapezoidal

56

(a) PIKAN: u_x (b) PIKAN: u_y
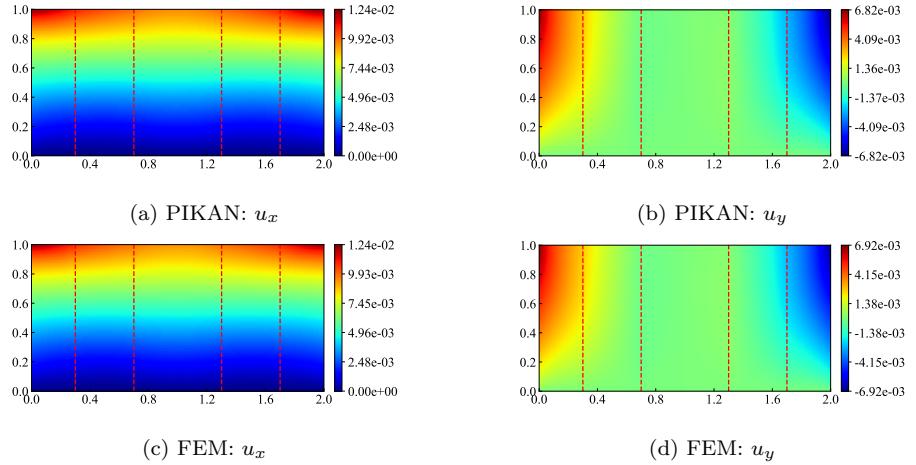
(c) FEM: u_x (d) FEM: u_y

Figure 33: Displacement field for dual-via TGV-Cu structure: PIKAN vs. FEM solutions.

integration schemes all achieve high accuracy and converge well to reference solutions. However, Delaunay integration exhibits some discrepancy from reference solutions, particularly near curved interface regions, consistent with observations from the heterogeneous plate example. The robust performance across different interface geometries confirms PIKAN's effectiveness for complex multi-material structural analysis without requiring specialized interface treatments or domain decomposition approaches.

Fig. 35 presents stress distributions along strategically selected analysis lines to evaluate PIKAN's accuracy in capturing stress concentrations and material interface effects. Figs. 35(a) and (b) show stress distributions along vertical lines at $x = 0.5$ (within the copper via) and $x = 0.9$ (within the glass substrate). These locations capture stress variations in regions of high stress concentration (near the via) and far-field behavior (in the glass matrix). Figs. 35(c) and (d) present stress comparisons along the horizontal line $y = 0.05$, which crosses the glass-copper interface near the bottom boundary for
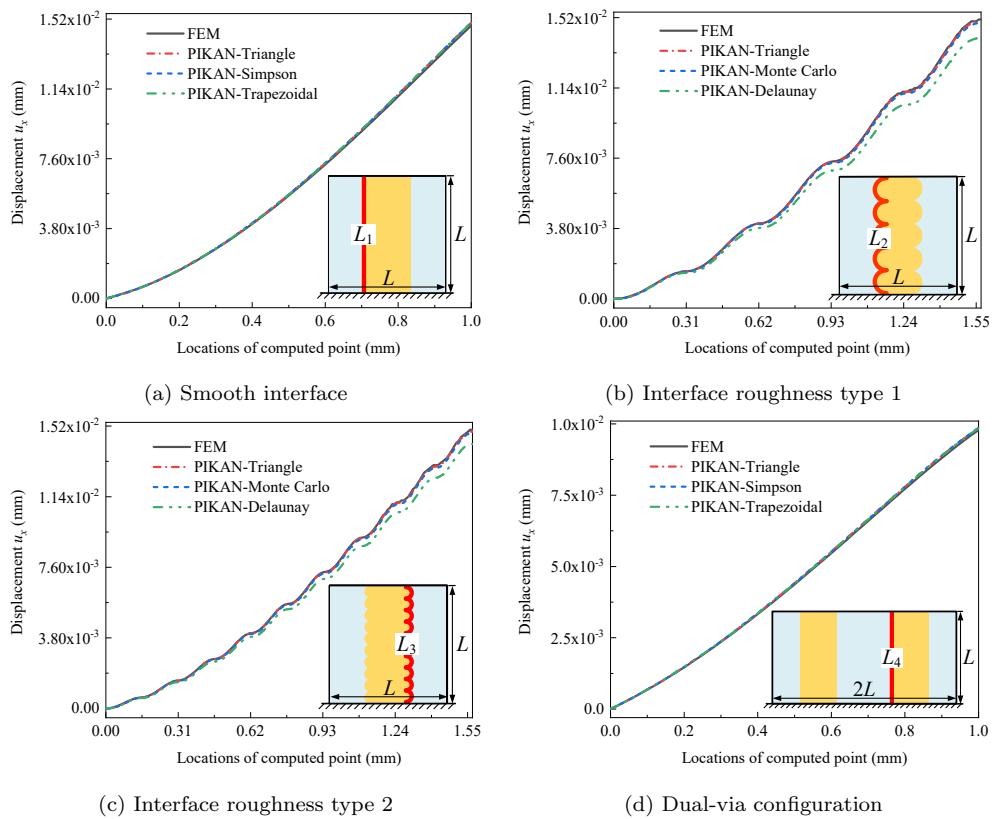
57

(a) Smooth interface

(b) Interface roughness type 1

(c) Interface roughness type 2

(d) Dual-via configuration

Figure 34: Displacement u_x along interfaces for different TGV-Cu configurations.

both single-via and dual-via configurations. This analysis line is critical for understanding stress transfer across material interfaces and potential stress concentration effects.

(a) Copper via region: $x = 0.5$

(b) Glass substrate region: $x = 0.9$

(c) Single-via interface: $y = 0.05$
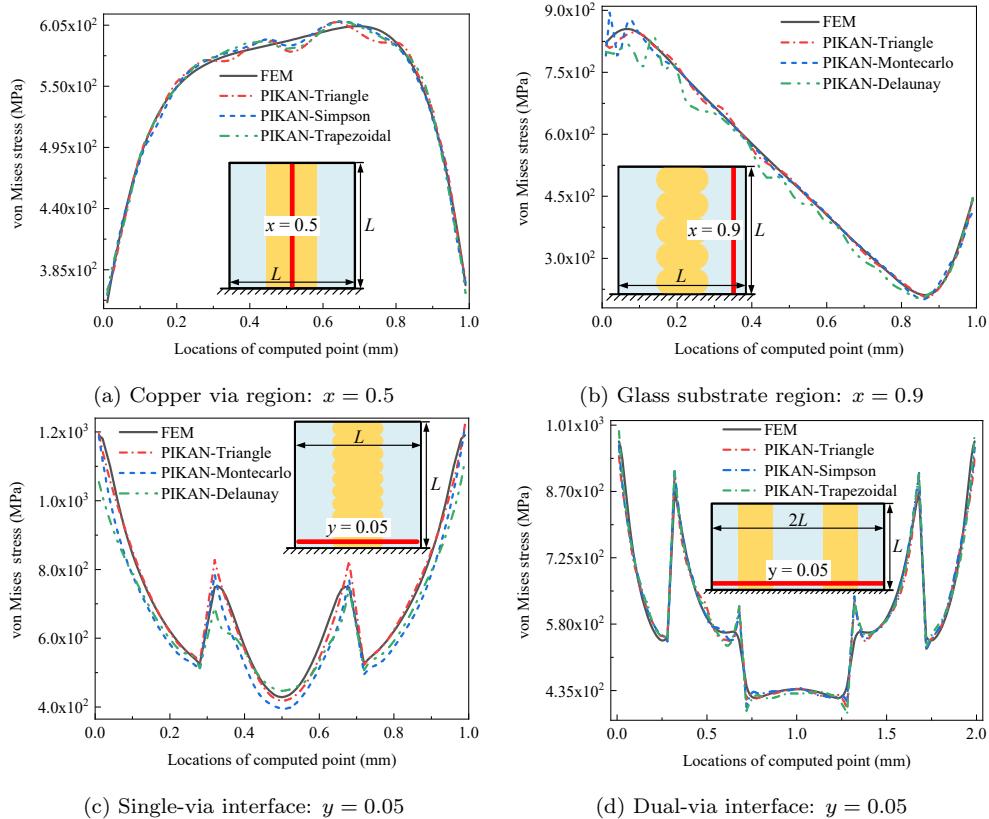
(d) Dual-via interface: $y = 0.05$

Figure 35: von Mises stress distributions at critical locations in TGV-Cu structures.

The results demonstrate clear performance differences among integration schemes. Triangular integration achieves the highest accuracy due to its adaptive mesh density that better resolves stress gradients near geometric discontinuities. Although Simpson's integration method has higher theoretical precision than the trapezoidal rule, PIKAN results using both methods achieve similar accuracy levels when the grid resolution is sufficiently high.

Monte Carlo integration performs moderately well but shows some scatter typical of stochastic methods. Delaunay integration exhibits the poorest stability, particularly near curved interfaces and material boundaries, consistent with previous observations about its suboptimal control area distribution.

6. Conclusion

This study proposes PIKAN, a novel method combining KAN with the DEM for analyzing multi-material elasticity problems in electronic packaging structures. KAN networks inherently possess piecewise function characteristics that naturally accommodate material property discontinuities, while DEM provides superior computational efficiency and accuracy compared to strong-form PINNs. The PIKAN method effectively integrates these advantages to achieve accurate analysis of complex multi-material structures. Unlike traditional approaches that introduce penalty terms to enforce essential boundary conditions, PIKAN ensures solution validity by constructing admissible displacement fields that automatically satisfy essential boundary conditions. The method requires only a single KAN network to approximate displacement fields across the entire computational domain, eliminating the subdomain partitioning required in conventional multi-material analysis. This approach avoids hyperparameter tuning for interface continuity conditions, significantly reducing training complexity while achieving superior accuracy compared to domain decomposition methods.

Numerical validation demonstrates PIKAN's accuracy and robustness across diverse multi-material configurations. Comparative analysis with traditional CENN and DEM approaches confirms PIKAN's superior perfor-

mance, particularly for complex interface geometries where penalty-based methods struggle with convergence. The study reveals that numerical integration strategy significantly affects solution accuracy. Additionally, PIKAN requires significantly fewer trainable parameters compared to MLP-based methods. However, fully exploiting its potential requires overcoming training efficiency challenges, which depends on future development of optimization algorithms specifically designed for KAN architectures. Furthermore, systematic hyperparameter tuning strategies tailored for PIKAN could further enhance its performance.

In conclusion, this research successfully extends PINNs to heterogeneous material problems, demonstrating significant potential for electronic packaging structure reliability analysis and broader multi-material engineering applications. Source codes are available at https://github.com/yanpeng-gong/PIKAN-MultiMateria.

Acknowledgments

References

[1] S. Zhang, X. Xu, T. Lin, P. He, Recent advances in nano-materials for packaging of electronic devices, Journal of Materials Science: Materials

in Electronics 30 (2019) 13855–13868.

[2] W. M. Moreau, Semiconductor lithography: principles, practices, and materials, Springer Science & Business Media, 2012.

[3] Y. K. Kim, S.-m. Lee, D.-s. Hwang, S. Jang, Analyses on the large size PBGA packaging reliability under random vibrations for space applications, Microelectronics Reliability 109 (2020) 113654.

[4] G. Hao, L. Zhou, H. Ren, L. Ran, B. Xie, Study on thermal buffering effect of phase change material on press-pack IGBT, International Journal of Heat and Mass Transfer 154 (2020) 119584.

[5] H. Yu, Y. Guo, Y. Gong, F. Qin, Thermal analysis of electronic packaging structure using isogeometric boundary element method, Engineering Analysis with Boundary Elements 128 (2021) 195–202.

[6] Y. Gong, Y. He, H. Hu, X. Zhuang, F. Qin, H. Xu, T. Rabczuk, A coupled finite element–boundary element method for transient elastic dynamic analysis of electronic packaging structures, Engineering Structures 326 (2025) 119500.

[7] F. Qin, Q. He, Y. Gong, T. An, P. Chen, Y. Dai, The application of FEM-BEM coupling method for steady 2D heat transfer problems with multi-scale structure, Engineering Analysis with Boundary Elements 137 (2022) 78–90.

[8] Y. P. Zhang, Finite-difference time-domain analysis of integrated ceramic ball grid array package antenna for highly integrated wireless

transceivers, IEEE Transactions on Antennas and Propagation 52 (2) (2004) 435–442.

[9] C. Liu, K. Aygün, A. E. Yılmaz, A parallel FFT-accelerated layered-medium integral-equation solver for electronic packages, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 33 (2) (2020) e2684.

[10] Y. Gong, F. Qin, C. Dong, J. Trevelyan, An isogeometric boundary element method for heat transfer problems of multiscale structures in electronic packaging with arbitrary heat sources, Applied Mathematical Modelling 109 (2022) 161–185.

[11] S. Feng, Y. Guo, G. Królczyk, X. Han, A. Incecik, Z. Li, An engineered solution to multi-physics of insulated gate bipolar transistor module considering electrical-thermal-mechanical coupling effect, Advances in Engineering Software 175 (2023) 103365.

[12] F. Qin, Q. He, Y. Gong, C. Hou, H. Cheng, T. An, Y. Dai, P. Chen, An automatic finite element method-boundary element method coupling method for elastic-plastic problems of multiscale structures in electronic packaging, Journal of Electronic Packaging 145 (2) (2023) 021003.

[13] Y. Gong, Y. Kou, Q. Yue, X. Zhuang, F. Qin, Q. Wang, T. Rabczuk, The application of thermomechanically coupled phase-field models in electronic packaging interconnect structures, International Communications in Heat and Mass Transfer 159 (2024) 108033.

[14] Y. Gong, Y. Kou, Q. Yue, X. Zhuang, N. Valizadeh, F. Qin, Q. Wang, T. Rabczuk, A phase-field study on thermo-mechanical coupled damage evolution and failure mechanisms of sintered silver interconnections, Engineering Fracture Mechanics 320 (2025) 111039.

[15] S. Kang, J. H. Lee, S. M. Kim, J. Lim, A.-Y. Park, S. Han, J.-Y. Song, S.-I. Kim, Machine learning-based solution for thermomechanical analysis of mmic packaging, Advanced Materials Technologies 8 (5) (2023) 2201479.

[16] P. Zaghari, S. S. Sinha, D. C. Hopkins, J. E. Ryu, Co-design and ML-based optimization of Through-Via in Silicon and Glass interposers for electronic packaging applications, IEEE Transactions on Components, Packaging and Manufacturing Technology (2025).

[17] C.-F. Yu, J.-W. Peng, C.-C. Hsiao, C.-H. Wang, W.-C. Lo, Development of GUI-driven AI deep learning platform for predicting warpage behavior of Fan-Out Wafer-Level packaging, Micromachines 16 (3) (2025) 342.

[18] M. Hou, Z. Ou, J. Long, S. Ding, G. Wen, Y. Chen, X. Chen, Machine learning enables accurate wire loop profile prediction for advanced microelectronics packaging, Journal of Manufacturing Processes 84 (2022) 394–402.

[19] L. Yuan, H. S. Park, E. Lejeune, Towards out of distribution generalization for problems in mechanics, Computer Methods in Applied Mechanics and Engineering 400 (2022) 115569.

[20] T. Rabczuk, K.-J. Bathe (Eds.), Machine Learning in Modeling and Simulation, 1st Edition, Computational Methods in Engineering & the Sciences, Springer Cham, 2023.

[21] M. Raissi, P. Perdikaris, 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 378 (2019) 686–707.

[22] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, Journal of Scientific Computing 92 (3) (2022) 88.

[23] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112790.

[24] W. E, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, Communications in Mathematics and Statistics 6 (1) (2018) 1–12.

[25] P. Yao, J. Yang, Y. Zhang, X. Fan, H. Chen, J. Yang, J. Wu, Physics-based nested-ANN approach for fan-out wafer-level package reliability prediction, in: 2022 IEEE 72nd Electronic Components and Technology Conference (ECTC), IEEE, 2022, pp. 1827–1833.

[26] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366.

[27] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proceedings of the National Academy of Sciences 115 (34) (2018) 8505–8510.

[28] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, M. Tegmark, Kan: Kolmogorov-arnold networks, arXiv preprint arXiv:2404.19756 (2024).

[29] P. Thakolkaran, Y. Guo, S. Saini, M. Peirlinck, B. Alheit, S. Kumar, Can kan cans? input-convex kolmogorov-arnold networks (kans) as hyperelastic constitutive artificial neural networks (cans), Computer Methods in Applied Mechanics and Engineering 443 (2025) 118089.

[30] Y. Diao, J. Yang, Y. Zhang, D. Zhang, Y. Du, Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology, Computer Methods in Applied Mechanics and Engineering 413 (2023) 116120.

[31] Y. Wang, J. Sun, W. Li, Z. Lu, Y. Liu, CENN: Conservative energy method based on neural networks with subdomains for solving variational problems involving heterogeneous and complex geometries, Computer Methods in Applied Mechanics and Engineering 400 (2022) 115491.

[32] H. Sheng, C. Yang, PFNN: A penalty-free neural network method for

solving a class of second-order boundary-value problems on complex geometries, Journal of Computational Physics 428 (2021) 110085.

[33] S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, Journal of Computational Physics 449 (2022) 110768.

[34] W. Li, M. Z. Bazant, J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, Computer Methods in Applied Mechanics and Engineering 383 (2021) 113933.

[35] N. Sukumar, A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114333.

[36] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, Y. Liu, Kolmogorov-Arnold-Informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on kolmogorov-arnold networks, Computer Methods in Applied Mechanics and Engineering 433 (2025) 117518.

[37] J. Bai, H. Jeong, C. P. Batuwatta-Gamage, S. Xiao, Q. Wang, C. M. Rathnayaka, L. Alzubaidi, G.-R. Liu, Y. Gu, An introduction to programming physics-informed neural network-based computational solid mechanics, International Journal of Computational Methods 20 (10) (2023) 2350013.

[38] S. P. Timoshenko, J. N. Goodier, Theory of Elasticity. 3rd ed., New York: McGraw-Hill, 1970.

[39] J. P. Gambinoa, S. A. Adderly, J. U. Knickerbocker, An overview of through-silicon-via technology and manufacturing challenges, Microelectronic Engineering 135 (2015) 73–106.