



Research paper

Rapid prediction of thermal stress on satellites via domain decomposition-based Hybrid Fourier Neural Operator

Kangrui Zhou ^{a,b,c}, Wei Peng ^{b,c}, Xiaoya Zhang ^{b,c}, Xu Liu ^{b,c}, Wen Yao ^{b,c,*}^a College of Aerospace Science and Engineering, National University of Defense Technology, No. 109, Deya Road, Changsha, 410073, China^b Defense Innovation Institute, Academy of Military Science, No. 53, Fengtai East Street, Beijing, 100071, China^c Intelligent Game and decision Laboratory, Beijing, 100071, China

ARTICLE INFO

Dataset link: <https://github.com/KangruiZhou/HFNO>

Keywords:

Thermal stress
 Fourier Neural Operator
 Domain decomposition
 Surrogate model

ABSTRACT

Rapid thermal stress analysis is crucial for the thermal design of satellites. To overcome the disadvantages of traditional algorithms in terms of efficiency, deep learning methods have been used to tackle these problems. However, using uniform grid-based techniques is challenging when faced with complex geometric shapes. To address this, we introduce the domain decomposition-based Hybrid Fourier Neural Operator (HFNO), a comprehensive framework for learning a multi-scale and end-to-end operator on two-dimensional point clouds. We then propose two decomposition metrics: a stress gradient-based metric for scenarios with prior knowledge of training data, and a mesh density-based metric for scenarios without prior knowledge. Leveraging K-Dimension tree-based domain decomposition optimized via Monte Carlo tree search, we decompose the computational domain into several disjoint rectangular subdomains. In the proposed hybrid framework, a Geometry-aware Fourier Neural Operator (Geo-FNO) is used to deal with subdomains with high-frequency information, while a Non-Uniform Fourier Neural Operator (NU-FNO) is used to deal with subdomains with low-frequency information. This framework effectively combines the advantages of two Fourier Neural Operator variants, overcoming the issue of large prediction errors on the subdomains with high-frequency information and ensuring stable prediction performance across different positions. Furthermore, we introduce a boundary loss term during the training process to enhance continuity across subdomain boundaries. The numerical results demonstrate that our method achieves a superior balance between efficiency and precision, surpassing that of a single algorithm.

1. Introduction

As an important service platform for space missions, satellites play an irreplaceable role in remote sensing, navigation, mapping, communication, and other fields (Montenbruck and Gill, 2000). Satellites work in a harsh environment while in orbit and are periodically subjected to alternating high and low temperatures from outer space, resulting in large temperature gradients. Moreover, satellite components will inevitably generate a lot of heat during work due to the high power density (Chen et al., 2020). The joint effect of the severe external temperature difference and internal heat sources is prone to thermal deformation, thermal buckling, or thermal-induced vibration, posing great challenges to structure stability (Zhengchun et al., 2016), equipment reliability (Boudjemai et al., 2014) and mission accuracy (Liu and Cai, 2022).

Traditionally, the finite difference method, the finite element method, and other numerical methods are used to analyze thermal

stress by solving partial differential equations (PDEs). Traditional algorithms typically exhibit high accuracy, but meeting real-time requirements is challenging in certain cases (Zhao et al., 2021). For example, in the process of satellite structure optimization design, it is difficult to provide a real-time response when considering structural performance requirements such as thermal deformation and thermal stress (Shen et al., 2019). It is urgent and necessary to perform a real-time analysis of the thermomechanical coupling problems of the satellite.

Building a surrogate model is a common method to achieve a compromise between computational accuracy and cost, improving the efficiency of prediction tasks. Traditional surrogate modeling methods including polynomial-based response surface (Goel et al., 2009), support vector machine regression (Clarke et al., 2004), radial basis function (Yao et al., 2012), and Kriging interpolation (Zhang et al., 2019) all face the challenge of “curse of dimensionality”, which means that the high dimension of input and output brings an exponential

* Corresponding author at: Defense Innovation Institute, Academy of Military Science, No. 53, Fengtai East Street, Beijing, 100071, China.
 E-mail address: wendy0782@126.com (W. Yao).

difficulty for modeling. Fortunately, deep learning models are considered to possess advantages in handling high-dimensional data. Maurizi et al. (2022) propose a general method based on graph neural networks (GNN) to predict stress, strain, and deformation fields in material and structural systems. Sharma et al. (2018) take different boundary conditions as input and apply the U-Net to predict the temperature field in a two-dimensional plane. Cao et al. (2022) propose MTA-UNet that adopts the U-Net as its basic architecture and is capable of incorporating physical information and weighting multiple physical tasks, resulting in highly accurate predictions of thermal stress. These surrogate models use supervised learning to directly approximate the input-output relationship and have demonstrated substantial speed improvements when compared to traditional solvers.

Recently, a new class of data-driven models, known as neural operators, has gained success in learning the operator mapping between infinite-dimensional function spaces. Neural operators directly learn the solution operator of PDE in a mesh-invariant manner and are invariant to discretization. Therefore, neural operators may have better potential to solve PDEs. Neural operators can be broadly classified into two types based on the input mesh type: mesh-based operators and mesh-less operators. Mesh-based operators show the advantages of being fast and accurate. In these operators, both the input data and output result are discretized into uniform grids. Famous examples include convolutional neural operators (CNO) (Raonic et al., 2023; Guo et al., 2016; Bhatnagar et al., 2019), Fourier neural operators (FNO) (Li et al., 2020a) and Transformer neural operators (GNOT) (Hao et al., 2023; Cao, 2021). However, in real-world applications with complex geometries and unstructured data, the mesh-less techniques are more flexible and can adapt to these challenging scenes. Existing mesh-less operators such as DeepONet (Lu et al., 2021) and graph neural operator (GNO) (Li et al., 2020b; Anandkumar et al., 2019) allow for arbitrary inputs and queries and can adapt to complex geometric shapes. But they have lost the speed and efficient mesh-based techniques like the Fast Fourier Transform (FFT). Some variants of FNO have demonstrated strong performance in terms of both precision and efficiency when applied to unstructured data. Geometry-aware FNO (Geo-FNO) (Li et al., 2024) deforms irregular input domains into uniform latent grids, allowing it to work on arbitrary geometries. However, the accuracy of the Geometry-aware Fourier Neural Operator (Geo-FNO) remains suboptimal, particularly when handling a wide range of ground truth values. Non-Uniform Fourier Neural Operators (NU-FNO) (Liu et al., 2023) partition the computational domain into several subdomains to mitigate non-uniformity and reduce subsequent interpolation errors. However, the interpolation technique still introduces significant errors in regions with high-gradient stress (characterized by high-frequency information), such as stress concentration areas in thermal stress prediction tasks.

Motivated by the limitations of the vanilla Fourier Neural Operator (FNO), particularly its tendency to filter out high-frequency information during the Fast Fourier Transform (FFT) (Li et al., 2024), and by the increased information loss due to interpolation onto structured grids, this paper proposes a novel domain decomposition-based Hybrid Fourier Neural Operator (HFNO) framework. This framework is applied to the prediction task of thermal stress in satellite motherboards. The domain is adaptively decomposed into subdomains characterized by high-frequency and low-frequency information, based on the coordinates and two alternative features of the point cloud: either mesh density or stress gradient magnitude. In regions with high-frequency information, the Geometry-aware Fourier Neural Operator (Geo-FNO) transforms unstructured meshes into uniform grids, thereby mitigating significant interpolation errors. In low-frequency regions, NU-FNO first interpolates the point cloud onto uniform grids and then efficiently learns the operator via FFT. This highlights the complementary nature of these two operators. This paper aims to achieve fast and accurate predictions of thermal stress with various thermal loads and complex geometric shapes.

Contributions. In summary, the general contributions of this paper include

- (1) We propose an unstructured grid-based thermal stress dataset consisting of temperature and stress data pairs. Local grid refinement is used for finer resolution in regions with high-gradient stress, and the unstructured grid can adapt to complex geometric shapes.
- (2) We propose an adaptive domain decomposition method based on KD trees optimized via Monte Carlo tree search. By developing a novel metric that integrates coordinates with either mesh density or stress gradient magnitude, which is tailored for two distinct scenarios, the domain is adaptively decomposed into several disjoint rectangular regions to better learn the solution operator.
- (3) We propose a novel framework, a Hybrid Fourier Neural Operator (HFNO), for fast stress prediction from irregular temperature fields. HFNO inherits the advantages of Geo-FNO and NU-FNO, which are applied in high-frequency subdomains and low-frequency subdomains respectively. They complement each other and improve both time efficiency and prediction precision.

The rest of the paper is structured as follows. Section 2 presents the background of the related techniques applied in this paper. Section 3 introduces the strategies applied in domain decomposition-based HFNO in detail. Section 4 shows the experimental setup and the performance of our method (HFNO). Section 5 draws a discussion and conclusion.

2. Background

In this section, we expand and explain the background concepts introduced in the HFNO architecture, including the mathematical modeling of thermal stress, Monte Carlo Tree Search and Fourier Neural Operators.

2.1. Mathematical modeling of thermal stress

This paper uses a two-dimensional planar elastomer with holes as a study case. The holes represent screw holes in engineering assembly. It is assumed that the relevant coefficients of the motherboard, such as the linear elasticity coefficient, thermal conductivity coefficient, and coefficient of linear expansion, remain constant regardless of temperature variations.

We consider the domain D with heat loads and specify the temperature function $T(x, y)$ on D . When the temperature changes, the thermal deformations across the domain are distinct under internal and external constraints on elastomers, causing thermal stress. This in turn leads to new additional tension, which affects thermal deformation. The thermal stress components and temperature change $T(x, y)$ satisfy the following equations:

$$\begin{cases} \sigma_{xx} = \frac{E}{1-\mu^2} \left(\frac{\partial u_x}{\partial x} + \mu \frac{\partial u_y}{\partial y} \right) - \frac{E\alpha T}{1-\mu} \\ \sigma_{yy} = \frac{E}{1-\mu^2} \left(\frac{\partial u_y}{\partial y} + \mu \frac{\partial u_x}{\partial x} \right) - \frac{E\alpha T}{1-\mu}, \\ \sigma_{xy} = \sigma_{yx} = \frac{E}{2(1+\mu)} \left(\frac{\partial u_y}{\partial x} + \mu \frac{\partial u_x}{\partial y} \right) \end{cases} \quad (1)$$

where α means the coefficient of linear expansion, E denotes Young's modulus. μ presents Poisson's ratio. In addition, σ_{xx} , σ_{yy} and σ_{xy} means x -directional thermal stress, y -directional thermal stress, and tangential thermal stress respectively. u_x and u_y represent x -directional displacement and y -directional displacement respectively. According to the equilibrium differential equation, the thermal displacement components u_x and u_y satisfy the following equations:

$$\begin{cases} \frac{\partial^2 u_x}{\partial x^2} + \frac{1-\mu}{2} \frac{\partial^2 u_x}{\partial y^2} + \frac{1+\mu}{2} \frac{\partial^2 u_y}{\partial x \partial y} - (1+\mu)\alpha \frac{\partial T}{\partial x} = 0 \\ \frac{\partial^2 u_y}{\partial y^2} + \frac{1-\mu}{2} \frac{\partial^2 u_y}{\partial x^2} + \frac{1+\mu}{2} \frac{\partial^2 u_x}{\partial x \partial y} - (1+\mu)\alpha \frac{\partial T}{\partial y} = 0 \end{cases}, \quad (2)$$

The stress boundary conditions according to displacement satisfy

$$\left\{ \begin{array}{l} l \left(\frac{\partial u_x}{\partial x} + \mu \frac{\partial u_y}{\partial y} \right)_s + m \frac{1-\mu}{2} \left(\frac{\partial u_x}{\partial y} + \mu \frac{\partial u_y}{\partial x} \right)_s \\ = l(1+\mu)\alpha(T)_s \\ m \left(\frac{\partial u_x}{\partial x} + \mu \frac{\partial u_y}{\partial y} \right)_s + l \frac{1-\mu}{2} \left(\frac{\partial u_y}{\partial x} + \mu \frac{\partial u_x}{\partial y} \right)_s \\ = m(1+\mu)\alpha(T)_s \end{array} \right., \quad (3)$$

where l and m can be expressed as the directional cosine of the x -axis and y -axis, respectively (Sitharam and Govindaraju, 2021).

In this paper, we assume the boundary conditions are fixed. The solution operator simplifies to $\mathcal{G}^\dagger : T \mapsto \sigma$. The thermal stress σ of the satellite motherboard with $T(x, y)$ can be calculated by the finite element method (FEM) numerically.

2.2. Monte Carlo tree search

Monte Carlo tree search (MCTS) (Coulom, 2007) is a heuristic search algorithm used for decision processes in large combinatorial spaces. It is designed to strike a balance between exploration (of not well-tested actions) and exploitation (of the best actions identified so far) by intelligently exploring the search tree. It performs random sampling through simulations and collecting statistics of actions, enabling more informed decisions in subsequent iterations. This algorithm has been successfully applied in various domains such as combination optimization (Luo et al., 2023) and symbolic regression (Sun et al., 2023), demonstrating its effectiveness. The detailed structure of a basic MCTS algorithm is discussed by explaining the four phases below:

- Selection.** The MCTS agent searches the portion of the tree that has already been represented in the memory. Selection always starts from the root node and selects the next node at each turn according to the given selection policy. This phase terminates when it reaches an expandable node or a leaf node.
- Expansion.** At an expandable node, the MCTS agent expands the search tree by selecting one of its unvisited children and adds to the tree represented in memory. When expansion reaches the terminal state, the current iteration skips directly to backpropagation.
- Simulation.** After expansion, if the current node is non-terminal, the agent performs a complete random simulation until reaching a terminal state and fetches the reward. In this phase, actions are randomly selected. This is the “Monte Carlo” part of the MCTS.
- Backpropagation.** After the simulation, the reward propagates back along the path from the current node to the root. The statistics of nodes are updated.

The selection policy aims to maintain a proper balance between exploration and exploitation. The most common algorithm is the Upper Confidence Bounds applied for Trees (UCT) (Kocsis and Szepesvári, 2006), formulated as:

$$UCT(s, a) = Q(s, a) + c \sqrt{\ln[N(s)] / N(s, a)}, \quad (4)$$

where $Q(s, a)$ is the average reward of taking action a in state s in all simulations performed so far, encouraging the exploitation of the current best child node; $N(s)$ is the number of times state s has been visited in previous iterations and $N(s, a)$ is the number of times action a has been selected at state s . $\sqrt{\ln[N(s)] / N(s, a)}$ consequently promotes the exploration of less-visited child nodes. The constant c controls the balance between exploration and exploitation, which is empirically defined based on the specific problem.

Table 1
Parameter list of dataset.

Parameter	Symbol	Value	Units
Length of plate	L	20	cm
Height of plate	H	10	cm
Radius of the holes	R	0.5	cm
Expansion coefficient	α	$1e-5$	/°C
Young's modulus	E	$50e3$	MPa
Poisson's ratio	ν	0.2	/
Smoothing rate	γ	0.5	/
Mean of GRFs	M	30	/
Std of GRFs	S	60	/

2.3. Fourier neural operators

Fourier Neural Operator (FNO) (Li et al., 2020a) has shown the unprecedented cost-accuracy tradeoff in simulating many PDEs such as the elastic wave equation (Zhang et al., 2023), multiphase flow (Alpak et al., 2023) and weather forecasts (Kurth et al., 2023). It is formulated directly in Fourier space, offering quasi-linear time complexity and state-of-the-art approximation capabilities. The integral operator is constrained to a convolution operation and implemented through a linear transformation in the Fourier domain. Define the Fourier integral operator \mathcal{K} as:

$$(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}(R_\phi \cdot (\mathcal{F}v_t))(x) \quad \forall x \in D, \quad (5)$$

where R_ϕ is the Fourier transform of a periodic function $\kappa : \overline{D} \rightarrow \mathbb{R}^{d_v \times d_v}$ parameterized by $\phi \in \Theta_K$. The Fourier transform \mathcal{F} is defined as:

$$(\mathcal{F}v)(k) = \int_x v(x)\psi(x, k)\mu(x) \approx \sum_{x \in \mathcal{T}} v(x)\psi(x, k), \quad (6)$$

where $\psi(x, k) = e^{2i\pi\langle x, k \rangle} \in L(D)$ is the Fourier basis and \mathcal{T} is the mesh sampled from the distribution μ . When the domain D is assumed to be a periodic, square torus and the mesh \mathcal{T} is uniform, \mathcal{F} can be implemented by the FFT (Anandkumar et al., 2019).

However, the FNO uses FFT as its base layer, which is limited to rectangular domains with uniform grids. To address the challenges posed by the irregular domain with unstructured data, two variants of FNO have been proposed. One is the Geo-FNO (Li et al., 2024), which learns a deformable mesh along with the solution operator on a uniform latent mesh. To transform the function $v(x)$ from the physical domain to the computation domain, the geometric Fourier transform is defined as:

$$\begin{aligned} (\mathcal{F}_a v)(k) &:= \int_{D^c} v^c(\xi)e^{-2i\pi\langle \xi, k \rangle} d\xi \\ &= \int_D v(x)e^{-2i\pi\langle \phi^{-1}(x), k \rangle} \left| \det [\nabla_x \phi^{-1}(x)] \right| dx, \\ &\approx \frac{1}{|\mathcal{T}^i|} \sum_{x \in \mathcal{T}^i} m(x)v(x)e^{-2i\pi\langle \phi^{-1}(x), k \rangle} \end{aligned} \quad (7)$$

where the weight $m(x) = \left| \det [\nabla_x \phi^{-1}(x)] \right| / \rho_a(x)$ and $\rho_a(x)$ is a distribution from which the input mesh \mathcal{T}^i is sampled. To transform the spectral function $\hat{v}(k)$ from the spectral space of the computation domain to the physical domain, the inverse geometric Fourier transform is defined as:

$$(\mathcal{F}_a^{-1}\hat{v})(x) = (\mathcal{F}_a^{-1}\hat{v})(\phi^{-1}(x)) = \sum_k \hat{v}(k)e^{2i\pi\langle \phi^{-1}(x), k \rangle} \quad (8)$$

The other is the NU-FNO (Liu et al., 2023), which decomposes the domain into several subdomains and interpolates onto structured grids for each subdomain. The interpolated subdomain grids are then used to train the underlying uniform grid-based model using FFT, thereby better balancing efficiency and precision compared to meshless operators.

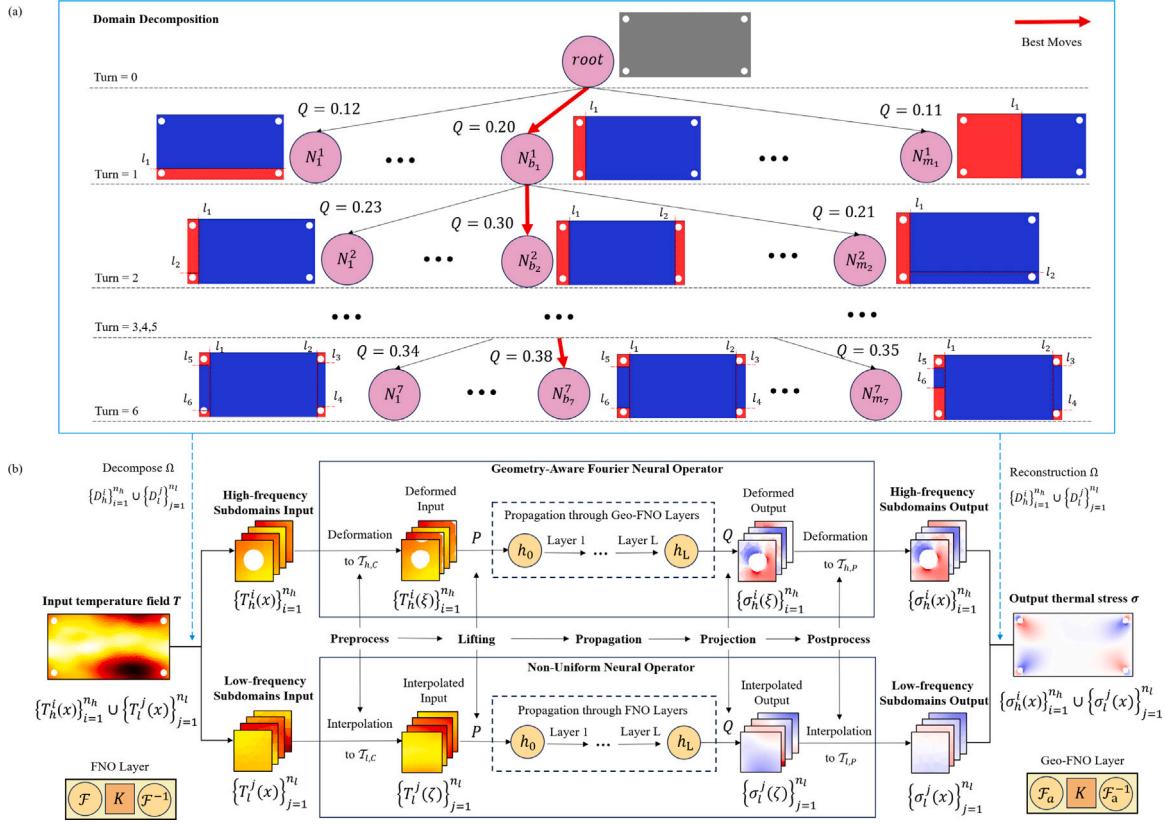


Fig. 1. (a) Domain decomposition: starting from $D^{(0)}$, we simulate the domain decomposition with several hyperplane l_i via MCTS. In the end, we choose the best actions and obtain n sub-point clouds according to the highest reward. **(b) HFNO framework:** 1. preprocess; 2. lifting by P ; 3. propagated through the Geo-FNO layers and FNO layers; 4. projection by Q ; 5. postprocess.

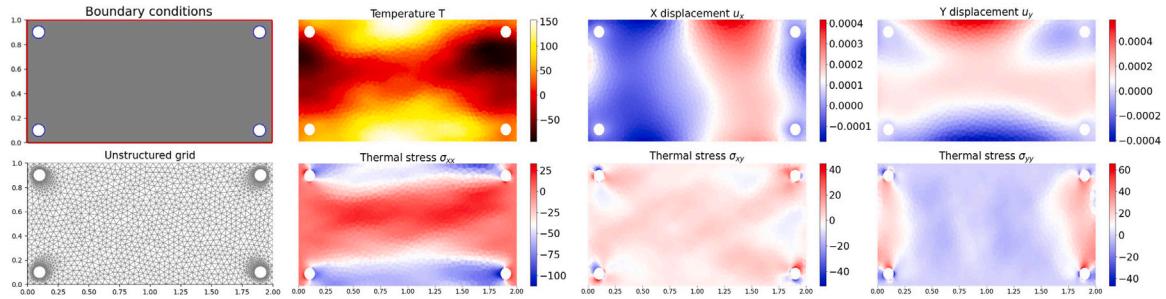


Fig. 2. A sample of our unstructured grid-based dataset, including boundary condition, unstructured grid temperature field and thermal stress. Non-displacement boundary conditions are marked in blue and free boundary conditions are marked in red.

3. Method

In this section, we will first introduce the dataset design in this paper, which can adapt to complex geometric shapes of satellite components. We then present a K-D tree algorithm optimized via Monte Carlo tree search for domain decomposition, followed by the strategies applied in domain decomposition-based Hybrid Fourier Neural Operator. The overview of the proposed framework is shown in Fig. 1.

3.1. Dataset design

In contrast to previous studies (Chen et al., 2020; Cao et al., 2022) that employed structured grids, we adopt unstructured grids in our dataset to flexibly handle complex geometric shapes. The presence of holes leads to sharp thermal stress in their vicinity, resulting in a stress concentration phenomenon. To accurately calculate the thermal stress in these regions, we apply local grid refinement for finer resolution.

The input temperature field is sampled from a Gaussian Random Field (GRF) (Murali, 2011), and the roughness of the temperature field matrix changes when the mean and covariance of GRF are adjusted. The parameters of GRFs are listed in Table 1. These parameters generate temperature loads ranging from -80°C to 145°C , which is representative of both the design phase and the in-orbit phase (Jurkowski et al., 2025; Corpino et al., 2015; guitarist Gilmore, 2002). In this paper, we generate a uniform grid and employ the nearest neighbor interpolation method to determine the temperature at each node on the unstructured grids.

Non-displacement boundary conditions are implemented along the edges of the inner holes. Free boundary conditions are applied to the rectangular boundary of the motherboard. The thermoelastic properties on the motherboard tend to be isotropic, with the parameters consistent with the thermoelastic properties of aluminum. The parameter list of the dataset is shown in Table 1. Then, we implement FEM by

Fenics (Alnæs et al., 2014) to obtain the thermal stress shown in Fig. 2.

3.2. Domain decomposition via K-D tree optimized by Monte Carlo tree search

Following the basic method of NU-FNO (Liu et al., 2023), the domain is decomposed into several disjoint rectangular subdomains. Due to the phenomenon of stress concentration, the domain D exhibits diverse frequency information across different regions. Capturing high-frequency information poses a significant challenge for the Fourier Neural Operator (FNO). To address this issue, we decompose the domain into regions characterized by high-frequency and low-frequency information, expressed as $D_h = \bigcup_{i=1}^{n_h} D_h^i$, $D_l = \bigcup_{j=1}^{n_l} D_l^j$. Domain decomposition aims to classify the given point clouds into several sub-point clouds based on their coordinates and feature values. We utilize the stress gradient magnitude or mesh density as features for domain decomposition in two distinct scenarios. When prior knowledge of the training data is unavailable, the stress gradient magnitude cannot be computed. In such circumstances, the mesh distribution metric serves as a viable alternative for domain decomposition, as it is independent of the training data. Conversely, when prior knowledge of the training data is available, employing the stress gradient magnitude for domain decomposition enables more precise predictions. We elaborate on this distinction in Appendix B. Using the average feature value of the domain as the threshold value, we calculate the average feature value of each subdomain. Based on this threshold, we classify these subdomains into two categories: high-frequency subdomains and low-frequency subdomains. In the following subsection, we employ the Geometry-aware Fourier Neural Operator (Geo-FNO) and the Non-Uniform Fourier Neural Operator (NU-FNO) in the two types of subdomains, respectively, to enhance the precision and stability of the predictions.

Define $\mathcal{M}(D) = \{x^{(i)} = (x_1^{(i)}, \dots, x_d^{(i)})\}_{i=1}^M$ be the point cloud, where d and M represents the dimensionality and the number of points in the point cloud, respectively. The optimization objective of domain decomposition can be formulated as the minimization of the total sum of squared feature values within each subdomain. It can be described as:

$$\min_{D^{(1)}, \dots, D^{(n)}} \sum_{i=1}^n TSS_i, \quad (9)$$

where $\bigcup_{i=1}^n D^{(i)} = D^{(0)}$ consist of n disjoint rectangular subdomains, TSS_i denotes the total sum of squared differences between the feature values and the mean feature value of the points in $D^{(i)}$, the formula is as follows:

$$TSS_i = \sum_{j=1}^{M^{(i)}} (f_i^{(j)} - \bar{f}_i)^2, \quad (10)$$

where $M^{(i)}$ denotes the number of points in $D^{(i)}$, $f_i^{(j)}$ is the feature value of the j th point in $D^{(i)}$, and \bar{f}_i denotes the mean feature value in $D^{(i)}$.

The optimization problem in Eq. (9) is a complex combinatorial optimization problem. Inspired by the K-D tree algorithm, we employ the Monte Carlo tree search (MCTS) to approximate the solution. Each node of the Monte Carlo tree represents a K-D tree and takes an action to transition to its child node. The actions of the MCTS agent, which consists of three components, involve choosing a subdomain denoted as D^* , determining the dimensionality represented by d^* , and splitting D^* using a hyperplane $x_d = b^*$, $x_{\min,d} \leq b^* \leq x_{\max,d}$, where $x_{\min,d}$ and $x_{\max,d}$ present the left and right interval endpoint of the subdomain coordinates in this dimension d^* respectively.

To evaluate the actions at the end of each simulation, we define a final reward based on TSS in $D^{(0)}$ and the sum of TSS_i in all subdomains. It is formulated as

$$reward = 1 - \frac{\sum_{i=1}^n TSS_i}{TSS_0}, \quad (11)$$

where n represents the number of disjoint rectangular regions. This rewarding formulation selects the potential child nodes according to our optimization target. The domain decomposition metric is formulated by minimizing the total sum of squared features across all subdomains. When the objective function reaches its minimum, the resulting subdomains predominantly consist of points that are exclusively either high-frequency or low-frequency. This optimization-driven approach enables effective partitioning of the entire domain into disjoint high-frequency and low-frequency subdomains.

In the process of every epoch of MCTS, we first judge whether the current node is expandable, that is whether the turn of the current node t is equal to the number of sub-regions n minus one. If not, the search is done. The K-D tree of the current node is the result of domain decomposition. If expandable, the agent has a 50% probability to expand the current node and another 50% to go into the best child node based on the UCB principle. Then, the agent simulates according to the random policy, that is randomly chooses the action $a_t \in \mathcal{A}$ until the search of the current epoch is done, where \mathcal{A} is the action space. MCTS updates the information of all nodes from bottom to top. The process of domain decomposition is shown at Fig. 1(a). More details of the algorithm are presented in Algorithm 1.

Algorithm 1: Domain Decomposition via Monte Carlo Tree Search and K-D Tree

```

Input: initial domain  $D^{(0)}$ , the number of subdomains  $n$ , and
       the simulation times  $m$ ;
Output: Optimal subdomains  $S = \bigcup_{k=1}^n D^{(k)}$ ;
1 Initialize:  $s_0 = D^{(0)}$ ,  $r = 0$ ,  $v = 0$ ;
2 for current turn = 0 to  $n - 1$  do
3   for epoch = 1 to  $m$  do
4     if  $s_t$  expandable and  $t < n - 1$  then
5       Selection:
6       if empty( $s_{t+1}$ ) or random() < 0.5 then
7         Expansion: Randomly take a action  $a_{t+1}$ , observe
            $s_{t+1}$ ;
8       else
9         |  $s_{t+1} = \arg \max_{C_t} UCT(r_t, v_t, v_{t+1})$ ;
10      end
11      |  $C_t \leftarrow s_{t+1} \cup C_t$ ;
12    end
13    Simulation:
14    while  $s_{t+1}$  expandable and  $t < n - 1$  do
15      Expansion: Randomly take a action  $a_{t+2}$ , observe
            $s_{t+2}$ ;
16      |  $t \leftarrow t + 1$ 
17    end
18    |  $r = Reward(s_n)$ ;
19    Backpropagate:
20    while  $t \neq$  current turn do
21      |  $r_t \leftarrow r + r_t$ ,  $v_t \leftarrow v_t + 1$ ,  $t \leftarrow t - 1$ ;
22    end
23  end
24  Choose the best state  $s_{t+1}$ 
25  |  $s_{t+1} = \arg \max_{C_t} UCT(r_t, v_t, v_{t+1})$ ;
26  |  $S \leftarrow s_{t+1}$ 
27 end

```

3.3. Hybrid Fourier Neural Operator

Motivated by the limitations of the vanilla Fourier Neural Operator (FNO), we address its inherent shortcomings in handling high-frequency information. Specifically, the vanilla FNO tends to filter out high-frequency information during the Fast Fourier Transform (FFT) (Li et al., 2024). Furthermore, the interpolation onto a structured grid exacerbates the information loss in these regions. In low-frequency

regions, rapid and accurate predictions are achieved by preserving low-frequency information through FFT processing. In the high-frequency subdomains, Geo-FNO learns a deformable grid along with the solution operator. It uses a fully connected network to deform the unstructured grids into adaptive grids and learns the operator in the latent space, which avoids the error that comes from the interpolation on the high-frequency subdomains. While in the low-frequency subdomains, NU-FNO first decomposes the domain and interpolates these sub-point clouds onto uniform grids, where the mesh-based technique based on FFT is employed to learn the solution operator. It is faster and more accurate than learning a mesh deformation. The complementarity enhances the performance of them in both types of subdomains.

Therefore, we apply the Geo-FNO and NU-FNO to learn the solution operator in the high-frequency subdomains and low-frequency subdomains respectively. As shown in Fig. 1(b), the framework can be decomposed into five main steps:

1. **Preprocess.** Initially, in the high-frequency subdomains $\bigcup_{i=1}^{n_h} D_h^i$, we deform the physical grid $T_{h,p}$ into an adaptive grid $T_{h,C}$ in the computational domain by a fully connected network. In the low-frequency regions $\bigcup_{i=1}^{n_l} D_l^i$, the input point cloud $\{T_l^i(x)\}_{i=1}^{n_l}$ are interpolated onto the input subdomain grids $T_{l,C}$.
2. **Lifting.** The temperature embeddings $\{T_h^j(\xi)\}_{j=1}^{n_h}$ in the adaptive grid $T_{h,C}$ are also transformed into hidden embeddings by a fully connected layer P . The temperature embeddings $\{T_l^i(\zeta)\}_{i=1}^{n_l}$ in subdomain grids $T_{l,C}$ are also transformed into hidden embeddings via P .
3. **Propagation.** These generated hidden embeddings in $T_{h,C}$ and $T_{l,C}$ are then propagated through the hidden layers by geometric Fourier transform pairs F_a , F_a^{-1} and Fourier transform pairs F , F^{-1} respectively.
4. **Projection.** Post this operation, the hidden embeddings in $T_{h,C}$ and $T_{l,C}$ are transformed into the output subdomain grids $T_{h,p}$ and $T_{l,p}$ respectively, facilitated by a fully connected layers Q .
5. **Postprocess.** Lastly, the computational grid $T_{h,C}$ are transformed to the physical grids $T_{h,p}$ and its embedding $\{\sigma_h^j(x)\}_{j=1}^{n_h}$ is obtained. The stress embeddings $\{\sigma_l^i(\zeta)\}_{i=1}^{n_l}$ in $T_{l,C}$ is interpolated back onto the output point cloud $\{\sigma_l^i(x)\}_{i=1}^{n_l}$. They constitute the final prediction of thermal stress $\sigma = \{\sigma_l^i(x)\}_{i=1}^{n_l} \bigcup \{\sigma_h^j(x)\}_{j=1}^{n_h}$.

Notably, the continuity issue represents a critical challenge in domain decomposition methods. To address this, we introduce a boundary loss term to enhance the continuity of the solution. The formulation of the boundary loss is as follows:

$$\mathcal{L}_{b,h} = \lambda \frac{1}{N} \sum_{j=1}^N (\text{GeoFNO}(T, x_i) - \sigma_i)^2, \quad (12)$$

$$\mathcal{L}_{b,l} = \lambda \frac{1}{N} \sum_{j=1}^N (\text{NUFNO}(T, x_i) - \sigma_i)^2, \quad (13)$$

where λ is the weight of the boundary loss in the total loss function, N denotes the number of sample points on the boundary, x_i represents the coordinates of the sample points, and T is the temperature load. GeoFNO and NUFNO map the temperature loads to thermal stress at the specified points. $\mathcal{L}_{b,h}$ and $\mathcal{L}_{b,l}$ denote the loss functions for the high-frequency and low-frequency regions, respectively. By incorporating the boundary loss, we aim to mitigate discontinuities across subdomains. To optimize the loss during the training process, we employ two distinct optimizers for the high-frequency and low-frequency regions. The loss functions are formulated as follows:

$$\mathcal{L}_{\text{high}} = \mathcal{L}_{\text{GEO}} + \mathcal{L}_{b,h}, \quad (14)$$

$$\mathcal{L}_{\text{low}} = \mathcal{L}_{\text{NUFO}} + \mathcal{L}_{b,l}, \quad (15)$$

where the loss functions \mathcal{L}_{GEO} and $\mathcal{L}_{\text{NUFO}}$ denote the L2 loss in the high-frequency and low-frequency subdomains, respectively. Additionally, $\mathcal{L}_{\text{high}}$ and \mathcal{L}_{low} are used to train the Geo-FNO and NU-FNO models, respectively.

4. Experiments

4.1. Experimental setup

We conducted thermal stress prediction experiments on three cases with different geometric domains, as shown in Appendix Fig. A.1, to demonstrate the generalization capability of the HFNO. Each case is a two-dimensional satellite motherboard with several holes. Case 1 involves the centrosymmetric distribution of holes at the corners of the rectangular motherboard. Case 2 features five holes distributed both at the boundary and the center of the motherboard. Case 3 involves five holes distributed asymmetrically.

Three datasets are generated by the FEM method on the three cases respectively, and each dataset consists of 6000 samples. each sample includes the temperature and thermal stress data pairs for all nodes in the grids. The first 5000 samples are allocated as train sets, while the remaining 1000 samples are designated as test sets. Subsequently, HFNO is trained using the training sets consisting of 5000 samples and is then evaluated on the test set. We take the tangential thermal stress σ_{xy} as our prediction task in this experiment.

We use the L2 loss function to train the model. The prediction performance of the model is measured by MAE and s_E , which present the mean value and standard deviation of the absolute errors respectively. They defined as:

$$MAE = \frac{1}{N} \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^M |r_{i,j}| \quad (16)$$

$$s_E = \sqrt{\frac{1}{N} \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^M (|r_{i,j}| - MAE)^2} \quad (17)$$

where $r_{i,j}$ presents the residual error of the j th point in i th sample of test sets. N presents the size of test sets and M is the number of nodes in a sample.

We apply z-score normalization to the temperature load and stress, transforming data with different distributions into a uniform scale. Additionally, we employ max-min normalization to scale the coordinate data within the interval $[-1, 1]$. Both normalization techniques are utilized to minimize the impact of features with larger numerical ranges on the network's learning process, ensuring that all features contribute equally to the model. This preprocessing step enables the neural network to learn the mapping from the temperature field to the stress field more effectively, while enhancing the model's generalization capability.

4.2. Result

Assuming prior knowledge of the gradient magnitude of the training data, we employ the metric based on stress gradient magnitude for domain decomposition, as illustrated in Column 1 of Fig. 3. This includes cases with complex geometric shapes, where the MCTS optimization process requires 108s, 429s, and 544s for the three cases, respectively. This demonstrates the generality of our approach and its suitability for the multiple structures of satellite motherboards. HFNO is trained over 200 epochs, with a total training time of less than 1 h for all three cases using a single Tesla T4 GPU.

After domain decomposition, Geo-FNO and NU-FNO are applied to the high-frequency subdomains and low-frequency subdomains. Table 2 shows the metrics and the time cost on the entire domain, high-frequency subdomains and low-frequency subdomains, respectively, of three cases. It demonstrates that the mean absolute error (MAE) is below 0.8 Pa across all three cases for the entire domain, with the standard deviation of the error (s_E) under 0.9 Pa. The training time is approximately 1 h. Additionally, in the high-frequency subdomains, the MAE is 0.1517 Pa higher, and the s_E is 0.3341 Pa higher on average compared to the low-frequency subdomains. This discrepancy arises

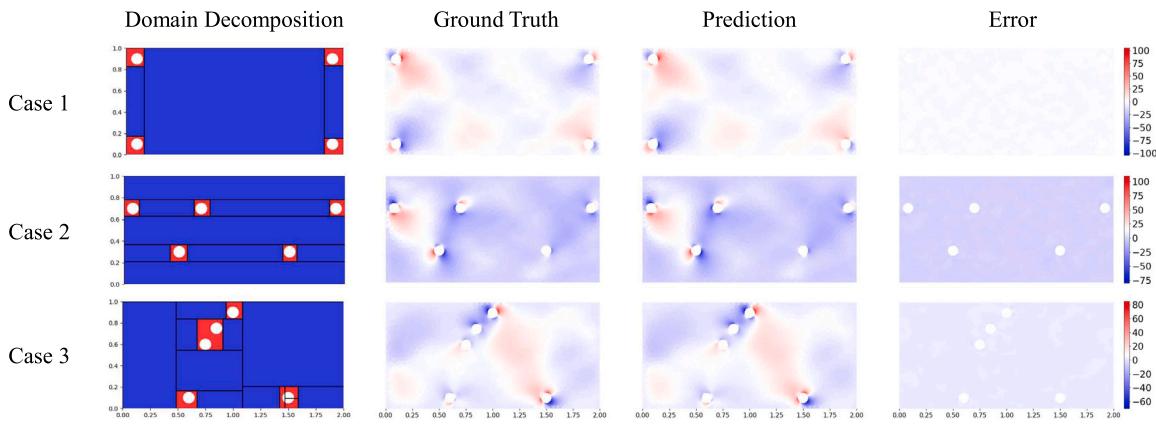


Fig. 3. The results of the domain decomposition (column 1), the ground truth (column 2), the prediction (column 3) and error (column 4) of HFNO on the three cases. These high-frequency and low-frequency subdomains are marked in red and blue respectively.

Table 2
Experimental results of three cases.

Case	Grid size	Subdomain	Metrics		Training time	
			MAE	s_E	per Epoch	per Run
Case 1	2128	High-frequency	0.7088	0.6569	6.7 s	53.3 min
		Low-frequency	0.5662	0.7229	6.2 s	
		Total	0.6167	0.7611	16.0 s	
Case 2	2883	High-frequency	0.8447	0.7852	7.6 s	57.7 min
		Low-frequency	0.7289	0.8296	6.4 s	
		Total	0.7667	0.8172	17.3 s	
Case 3	2699	High-frequency	0.8648	0.8529	7.2 s	55.3 min
		Low-frequency	0.6197	0.5615	6.3 s	
		Total	0.7076	0.6906	16.6s	

due to the inherent challenge of capturing high-frequency information associated with high-gradient stress using the Fourier Neural Operator.

A sample with an enlarged error heat map of each high-frequency and low-frequency subdomain is shown in Fig. 4. Subdomains 1-4 present the error of high-gradient subdomains and Subdomains 5-7 are in low-gradient subdomains of Case 1. Although there are significant variations in thermal stress magnitudes among these subdomains, the performance of HFNO is excellent and stable. Besides, in the low-frequency subdomains, the junction with the high-frequency subdomains exhibits larger errors than other positions. This verifies the shortcomings of NU-FNO, which exhibits poor performance in regions with high-frequency information (high gradient stress) because of its interpolation. The metrics of each subdomain are shown in Table 3. Despite the significant difference in the number of nodes in Subdomain 5, HFNO consistently demonstrates steady performance in both types of subdomains. Subdomains 1-4, representing high-frequency regions, exhibit an MAE of 0.7090 Pa and s_E of 0.4317 Pa. In contrast, Subdomains 5-7, corresponding to low-frequency regions, demonstrate an MAE of 0.5796 Pa and s_E of 0.5670 Pa. The MAE in high-frequency subdomains is 0.1253 Pa higher, while s_E is 0.1353 Pa lower compared to low-frequency subdomains.

4.3. Effects of the domain decomposition strategy

We compared the proposed HFNO with four methods: GCN (Kipf and Welling, 2017), GEN (Alet et al., 2019), Geo-FNO (Li et al., 2024) and NU-FNO (Liu et al., 2023) on Case 3. The quantitative results shown in Table 4 indicate that the HFNO outperforms the comparison methods, which achieves best precision with MAE of 0.7076 Pa and best stability s_E of 0.6906 Pa. It demonstrates its superior performance compared to the GCN and GEN. Leveraging the domain decomposition technique, we implement Geo-FNO and NU-FNO in these subdomains based on their features. Therefore, HFNO inherits the advantages of

Table 3
Experimental results of each subdomains.

Type of subdomain	Subdomain number	Number of nodes	Metrics	
			MAE	s_E
High-frequency	Subdomain 1	175	0.6860	0.3850
	Subdomain 2	181	0.7125	0.4355
	Subdomain 3	181	0.6990	0.4059
	Subdomain 4	170	0.7385	0.5003
Low-frequency	Subdomain 5	1127	0.6040	0.6290
	Subdomain 6	154	0.5766	0.5606
	Subdomain 7	140	0.5582	0.5115

both Geo-FNO and NU-FNO. Specifically, it transforms unstructured meshes into uniform grids in the latent space, significantly reducing interpolation errors in high-frequency subdomains compared to NU-FNO, which exhibits an MAE of 0.7866 Pa and s_E of 0.9474 Pa. Additionally, HFNO achieves faster training, requiring 16.6s per epoch through the use of FFT in low-frequency subdomains, compared to Geo-FNO, which takes 23.8s per training epoch. Furthermore, HFNO demonstrates superior effectiveness by decomposing the domain based on both coordinates and gradients of the point cloud, unlike NU-FNO, which relies solely on coordinates. This is evidenced by HFNO's improved performance in low-frequency subdomains, with an MAE of 0.6197 Pa and s_E of 0.5615 Pa, outperforming NU-FNO's MAE of 0.6266 Pa and s_E of 0.5779 Pa. The slower training speed in high-frequency subdomains is attributed to the adoption of Geo-FNO.

The error distributions across these methods are illustrated in Fig. 5. HFNO demonstrates smaller errors in both high-frequency and low-frequency subdomains. Unlike Geo-FNO, which exhibits a uniform error distribution across the entire domain, errors in other methods are concentrated in high-frequency regions (around the holes). The underperformance of NU-FNO in high-frequency subdomains is primarily due to its inefficiency in handling linear interpolation. HFNO

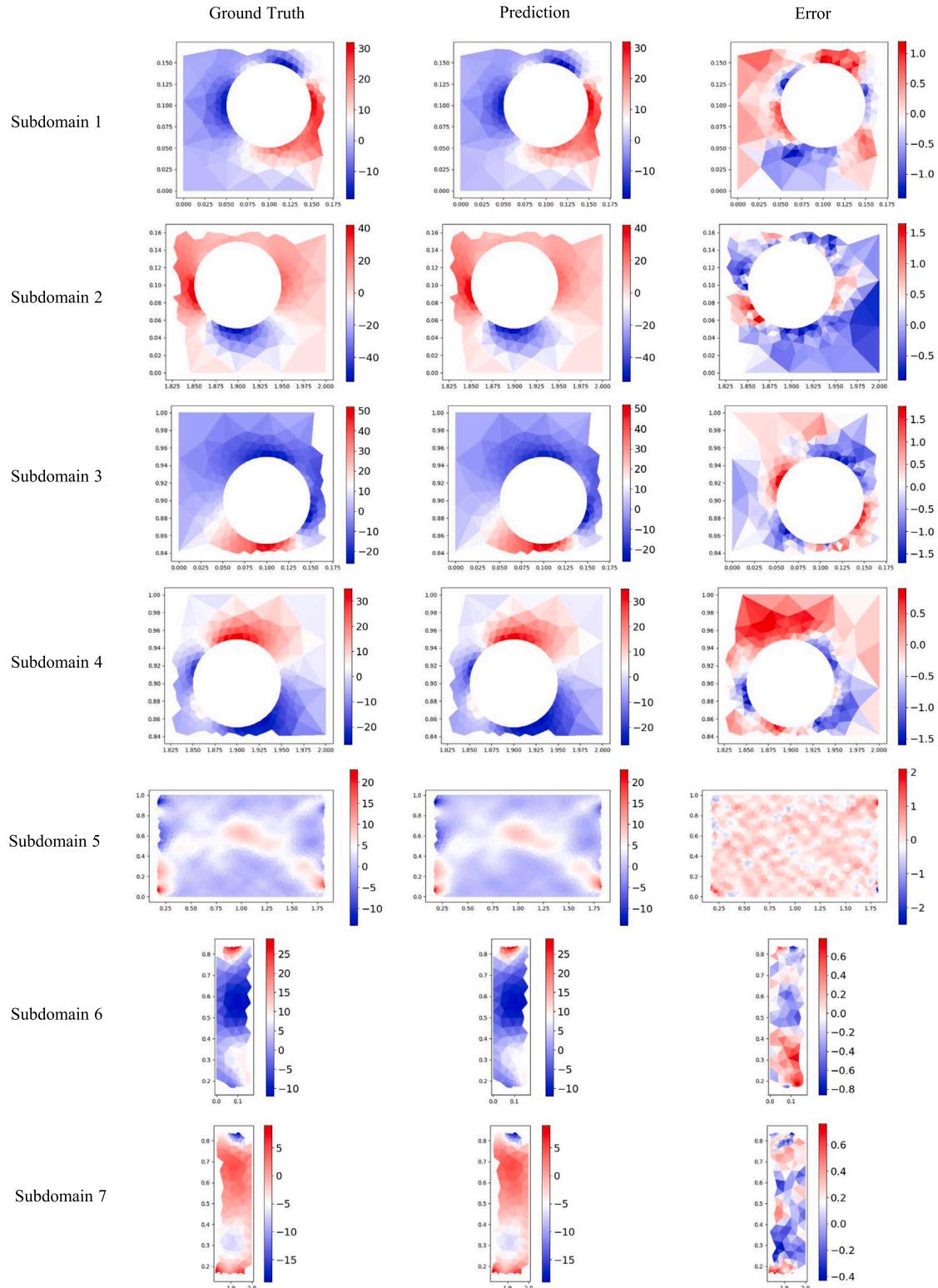


Fig. 4. The ground truth (column 1), the prediction (column 2) and the error (column 3) in each subdomain of Case 1.

Table 4
Comparison of different methods.

Method	Learnable parameters	Subdomain	Metrics		Training time	
			MAE	s_E	per Epoch	per Run
GCN	857089	High-frequency	2.6121	5.556	304.5 s	1015.0 min
		Low-frequency	2.3768	4.9891		
		Total	2.7325	5.0129		
GEN	858114	High-frequency	1.4466	1.6012	321.9 s	1073.0 min
		Low-frequency	1.3989	1.1021		
		Total	1.5332	1.3667		
Geo-FNO	850717	High-frequency	0.9238	0.8647	23.8 s	79.3 min
		Low-frequency	1.0604	0.9858		
		Total	1.0166	0.9458		
NU-FNO	819044	High-frequency	1.1888	1.6618	7.5 s	25.0 min
		Low-frequency	0.6266	0.5779		
		Total	0.7866	0.9474		
HFNO (ours)	807738	High-frequency	0.8648	0.8529	16.6 s	55.3 min
		Low-frequency	0.6197	0.5615		
		Total	0.7076	0.6906		

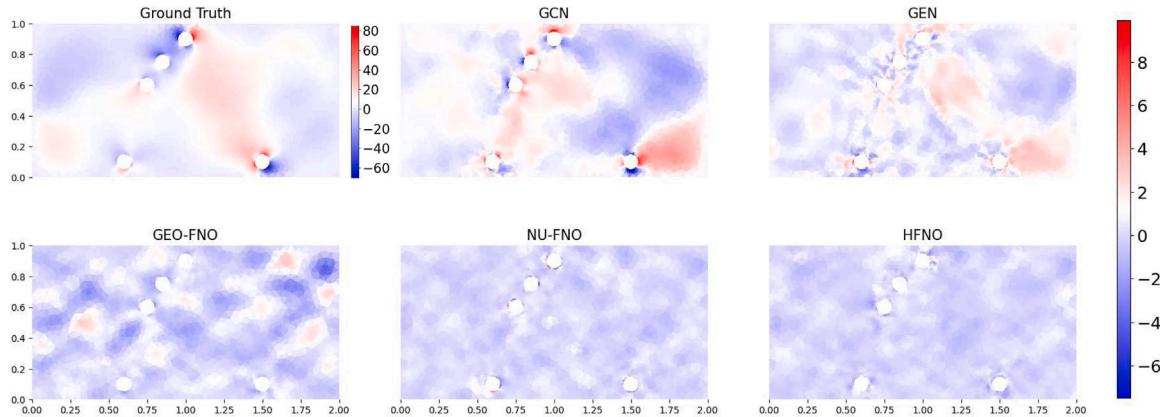


Fig. 5. A sample includes the ground truth and the error heat map of GCN, GEN, Geo-FNO, NU-FNO and HFNO methods.

effectively addresses these limitations, producing a smoother error distribution compared to other methods. Moreover, the error range in both high-frequency and low-frequency subdomains is significantly reduced, achieving much smaller magnitudes than those of single models.

5. Conclusion and discussion

In this paper, we propose a novel framework, coined as Hybrid Fourier Neural Operator (HFNO), to achieve rapid and precise prediction of stress thermal on satellites. Firstly, we first design an unstructured grid-based dataset, which can adapt to the complex geometric shapes of satellite components. Then, we propose an adaptive domain decomposition algorithm to decompose the computational domain into several disjoint rectangular regions. They are classified into high-frequency and low-frequency subdomains based on a comprehensive consideration of their coordinates and two alternative features of the point cloud: either mesh density or stress gradient magnitude, depending on the availability of prior knowledge from the training data. This classification establishes the foundation for predictions within subdomains characterized by distinct frequency information. Next, in the high-frequency subdomains, We employ Geometry-aware Fourier Neural Operator (Geo-FNO) to transform the unstructured mesh into uniform grids in the latent space, thereby mitigating high interpolation errors. Conversely, for the low-frequency subdomains, we have tailored NU-FNO to facilitate rapid and accurate learning of the solution operator. This emphasizes the complementary nature of both Geo-FNO

and NU-FNO. The temperature loads and corresponding coordinates are responsible for the thermal stress that is used in the prediction process. Additionally, we define a boundary loss function that quantifies the discrepancy between the predictions of the two surrogate models, thereby significantly enhancing continuity across subdomains. Finally, we conduct experiments in three cases with various geometric shapes and temperature loads to prove the universality and superiority of HFNO. It achieves a better trade-off between efficiency and precision of thermal stress.

Despite HFNO achieving a better trade-off between time efficiency and precision, further research is still required. Initially, the current work focuses on square components with circular holes. Extending the approach to more complex geometries and nonlinear problems would be valuable. Furthermore, the domain decomposition remains fixed during deployment. A flexible decomposition strategy could broaden the application scope, particularly in scenarios where regions with high-frequency information undergo significant variations across different samples. Additionally, we apply a soft constraint to the predictions at the boundaries where subdomains intersect, enhancing the continuity across these regions. It is worth exploring strategies to further ensure continuity maintenance within the domain decomposition framework. The application of the method proposed in this paper is not limited to the rapid prediction of thermal stress. It can be widely used in various fields, including the prediction of flow field, temperature field and stress field.

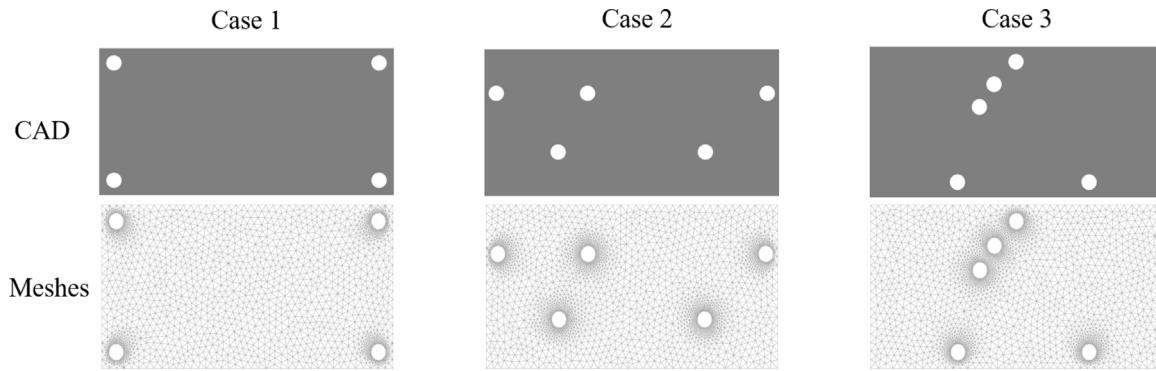


Fig. A.1. Three cases in the experiment that contain the CAD model (row 1) and the unstructured grids (row 2).

CRediT authorship contribution statement

Kangrui Zhou: Writing – original draft, Validation, Software, Methodology, Formal analysis. **Wei Peng:** Writing – review & editing, Supervision, Resources. **Xiaoya Zhang:** Writing – review & editing, Supervision. **Xu Liu:** Writing – review & editing. **Wen Yao:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 92371206).

Appendix A. Experimental details

In this part, We provide supplementary descriptions of the experimental environment, data generation and hyperparameters.

Experimental environment. Our codes are written in Python and the deep learning part is based on PyTorch library. All the models are trained on 1 NVIDIA Tesla T4 15 GB GPU.

Data Generation. We generate a temperature field as the input by Gaussian Random Fields (GRFs) and set the boundary condition based on the experiment case. Then, we use the FEniCS library to solve the thermal stress by the finite element method. The three cases used in the experiment is shown in Fig. A.1.

Hyperparameters. The optimizer is Adam with a learning rate of 10^{-3} and weight decay of 10^{-4} . The learning rate scheduler is StepLR with step size 100 epochs and decay rate 0.5. The hyperparameters of each baseline are listed below:

- **GCN, GEN** The learning rate scheduler is ReduceLROnPlateau with patience $\lfloor \text{epochs}/20 \rfloor$. Both of them have 8 fully connected layers with $k = 256$ as encoder and decoder and a graph convolutional layer with message passing steps 10. GEN has an additional weight in the graph edges. They are trained for 1400 epochs with batch size 20.
- **Geo-FNO** It has 5 2D Fourier layers with modes 12 and width 22 and a deformation network consisting of 7 fully connected layers. It is trained for 200 epochs with batch size 20.
- **NU-FNO** It has 4 2D Fourier layers with modes 8 and width 28. The number of subdomains is 16. It is trained for 200 epochs with batch size 20.
- **HFNO** It has 9 2D Fourier layers with modes 8 and width 20. The other parameters are the same as the corresponding baselines, except for the number of subdomains which is 7, 13 and 15 in three cases respectively. It is trained for 200 epochs with batch size 20.

Appendix B. Comparison of decomposition metric

We propose two features for domain decomposition: the stress gradient magnitude and mesh density, for two scenarios. Mesh density partitions the domain independently of the training data, making it potentially extensible to a broad range of applications. In scenarios where prior knowledge of the training data is unavailable, the stress gradient cannot be computed. In such cases, the mesh distribution metric serves as a viable alternative for domain decomposition. Conversely, when prior knowledge of the training data is available, employing the gradient magnitude for domain decomposition enables more precise predictions. We elaborate on this distinction as follows.

Employing a gradient-based metric leads to more precise predictions, as it enables more accurate identification of high-frequency and low-frequency regions compared to mesh distribution, despite the strong correlation between the two. The gradient distribution is more concentrated around the hole, resulting in a more refined decomposition for high-frequency regions. This ultimately enhances prediction precision. To demonstrate visually, we apply the Kernel Density Estimation method (Liu et al., 2023) to compute the density of these points on the mesh, which measures the mesh distribution. The high-frequency regions are predominantly localized around the hole, as shown in rows 1 and 2 of Fig. B.1. We then conduct a comparative experiment to illustrate the advantage of our metric. For fair comparison, we substitute the stress gradient magnitude with mesh density as the domain decomposition feature, while keeping all other parameters unchanged.

The domain decomposition results of the two features are illustrated in rows 3 and 4 of Fig. B.1. The gradient-based metric distinguishes high-frequency regions with greater precision compared to the density-based metric, which often encompasses a substantial number of low-frequency areas within regions classified as high-frequency. This inclusion complicates the stress gradient distribution and increases the difficulty of learning within high-frequency zones, consequently leading to larger prediction errors in these subdomains. The comparison of prediction error is shown in Table B.1. By employing domain decomposition based on the gradient, we achieve an average mean absolute error (MAE) that is reduced by more than 0.0563 Pa compared to the density-based metric, with an average standard deviation of absolute error (s_E) less than 0.0324 Pa. This indicates greater stability in the predictions. In low-frequency regions, the density-based metric exhibits an MAE that is 0.0512 Pa higher compared to the gradient-based metric, while s_E values are 0.1611 Pa lower than those of the gradient-based metric. This is because the low-frequency subdomains derived from the density-based metric exhibit lower frequency variations compared to those derived from the gradient-based metric.

The metric based on mesh density, which is independent of the training data, offers the potential for a more flexible division strategy during deployment. However, the method for handling inputs of varying sizes across different decompositions remains an area that requires further study.

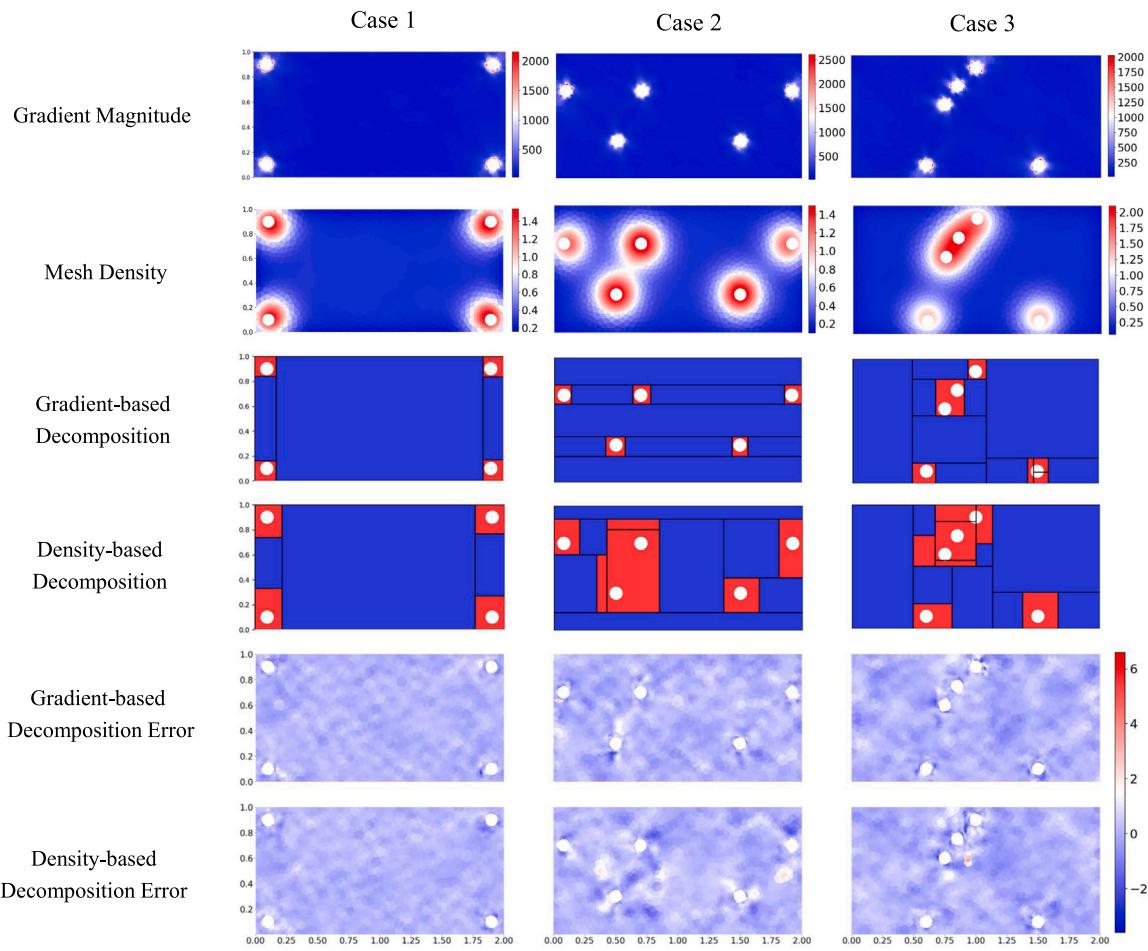


Fig. B.1. Comparison of the value, decomposition result and a sample of error distribution between gradient-based metric and density-based metric.

Table B.1
Comparison of the result between the gradient-based metric and density-based metric.

Decomposition metric	Case	Subdomain	Metrics	
			MAE	s_E
Gradient-based	Case1	High-frequency	0.7088	0.6569
		Low-frequency	0.5662	0.7229
		Total	0.6167	0.7611
	Case2	High-frequency	0.8447	0.7852
		Low-frequency	0.7289	0.8296
		Total	0.7667	0.8172
	Case3	High-frequency	0.8648	0.8529
		Low-frequency	0.6197	0.5615
		Total	0.7076	0.6906
Density-based	Case1	High-frequency	0.7547	0.6999
		Low-frequency	0.5785	0.5446
		Total	0.6663	0.7805
	Case2	High-frequency	0.9904	0.8860
		Low-frequency	0.5781	0.5365
		Total	0.8377	0.8818
	Case3	High-frequency	0.8620	0.7775
		Low-frequency	0.6026	0.5495
		Total	0.7558	0.7037

Appendix C. Ablation study on the number of subdomain

An ablation study on the number of subdomains is conducted using the decomposition metric based on the stress gradient magnitude, and its result is shown in [Table C.1](#). The error decreases as the number of subdomains increases in most cases. This improvement is attributed

to the successful decomposition of the two types of subdomains. By employing the Geometry-aware Fourier Neural Operator (Geo-FNO) in the high-frequency subdomains, significant interpolation errors are mitigated, resulting in enhanced precision and stability. However, an excessive number of subdomains can reduce prediction precision because NU-FNO requires padding each subdomain to a uniform grid of the same shape. This results in some subdomains containing significantly fewer points than others, leading to suboptimal learning performance.

The significant difference in errors when transitioning from 7 to 10 regions in cases 2 and 3 is attributed to incorrect domain decomposition of high-frequency and low-frequency regions. The decomposition results for 7 and 10 regions are illustrated in [Fig. C.1](#). For 7 subdomains in Case 2, the two high-frequency subdomains at the bottom are misclassified as low-frequency. This misclassification leads to the application of NU-FNO in these subdomains, resulting in substantial interpolation errors. In contrast, for 10 subdomains in Case 2, these high-frequency regions are correctly identified, avoiding the significant interpolation errors. The corresponding error distributions are illustrated in [Fig. C.2](#). For 7 subdomains in Case 3, a considerable number of low-frequency points are incorrectly classified into high-frequency subdomains, leading to a larger mean absolute error (MAE) compared to the 10-subdomain case. However, the impact of this misclassification is less severe than in the first situation. This phenomenon does not occur in Case 1 because the geometry is simpler compared to cases 2 and 3. Using 7 subdomains is sufficient to accurately distinguish all high-frequency regions.

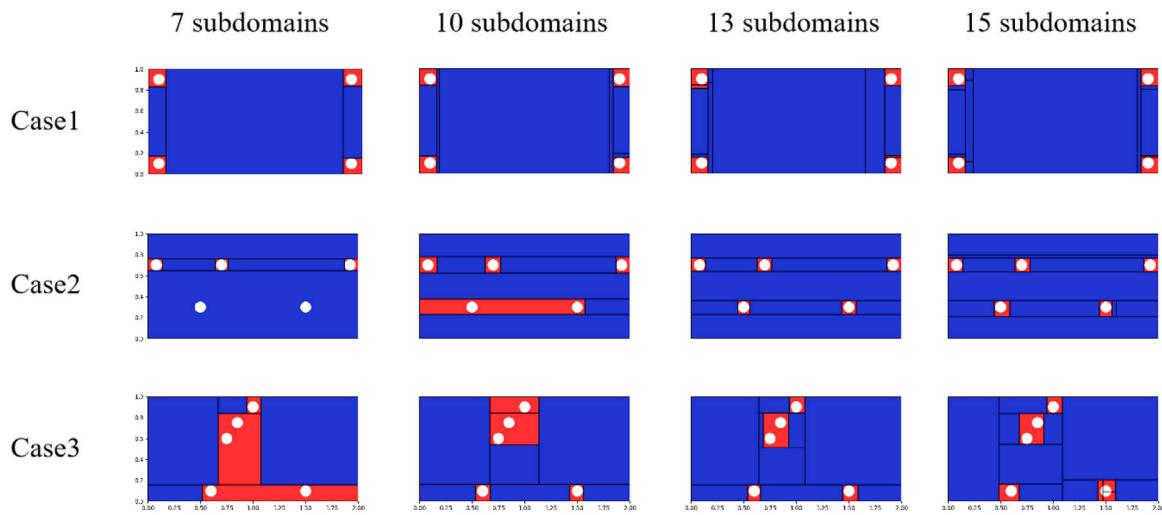


Fig. C.1. The domain decomposition scheme on different subdomains.

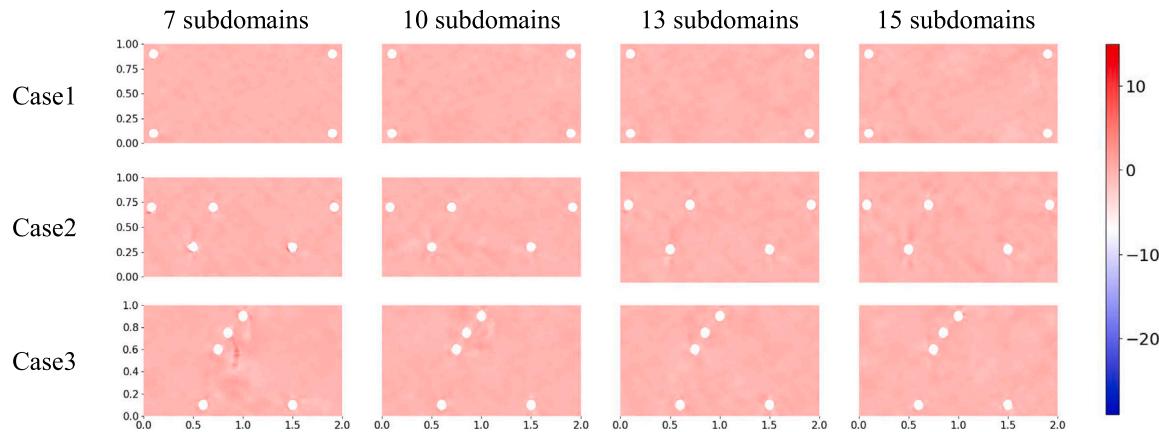


Fig. C.2. A sample of error distribution across varying numbers of subdomains.

Table C.1
Results of an ablation study on the number of subdomains.

Case	Metrics	Number of subdomains			
		7	10	13	15
Case 1	MAE	0.6167	0.6042	0.6236	0.6414
	s_E	0.7611	0.5702	0.6180	0.5881
Case 2	MAE	0.9886	0.7506	0.7667	0.7854
	s_E	1.9154	0.4723	0.7725	0.7607
Case 3	MAE	0.7741	0.7106	0.6841	0.7076
	s_E	0.8257	0.7307	0.6690	0.6906

Appendix D. Ablation study on subdomains applied two models

We select the average thermal stress gradient magnitude as the threshold because the distribution of the stress gradient magnitude serves as a key discriminator between two types of subdomains in the motherboard with screw holes in the engineering assembly, as illustrated in Fig. D.1. Subdomains with low mean stress gradient magnitude are predominantly concentrated on the left side due to stress concentration (Crawford and Martin, 2020). Utilizing the average thermal stress gradient magnitude enables the distinction of regions with relatively high-frequency stress. We conduct an ablation study to validate the optimality of our choice of subdomains and corresponding models by applying only Geo-FNO or NU-FNO to these selected sub-domains. The results, as shown in Table D.1, demonstrate that both

the mean absolute error (MAE) and the standard deviation of absolute error (s_E) are lower in our proposed approach. Specifically, in the high-frequency subdomain, applying Geo-FNO reduces the MAE by an average of 0.5702 Pa and s_E by an average of 0.4781 Pa compared to NU-FNO. Conversely, in the low-frequency subdomains, applying NU-FNO reduces the MAE by 0.0870 Pa and s_E by 0.4367 Pa compared to Geo-FNO. The experimental results statistically validate that our choice of subdomains and corresponding models is optimal, enabling Geo-FNO and NU-FNO to complement each other through domain decomposition.

Appendix E. Ablation study on the boundary loss

The continuity issue indeed represents a critical challenge in domain decomposition methods. To address this, we introduce a boundary loss term to enhance the continuity of the solution. To evaluate the effectiveness of this approach, we conduct an ablation study on the inclusion of the boundary loss, as detailed in Table E.1. The mean absolute error (MAE) on the boundary is defined as the difference between the predictions of the two surrogate models, while the standard deviation of this difference is denoted as s_E . The results demonstrate that the MAE on the boundary is below 1.1 Pa, and s_E is below 1 Pa, while the overall MAE across the entire domain is below 0.8 Pa. By incorporating the boundary loss, we observe a trade-off: the overall MAE increases by 0.0817 Pa, and s_E increases by 0.4414 Pa. However, this approach significantly improves boundary performance, reducing the MAE on the

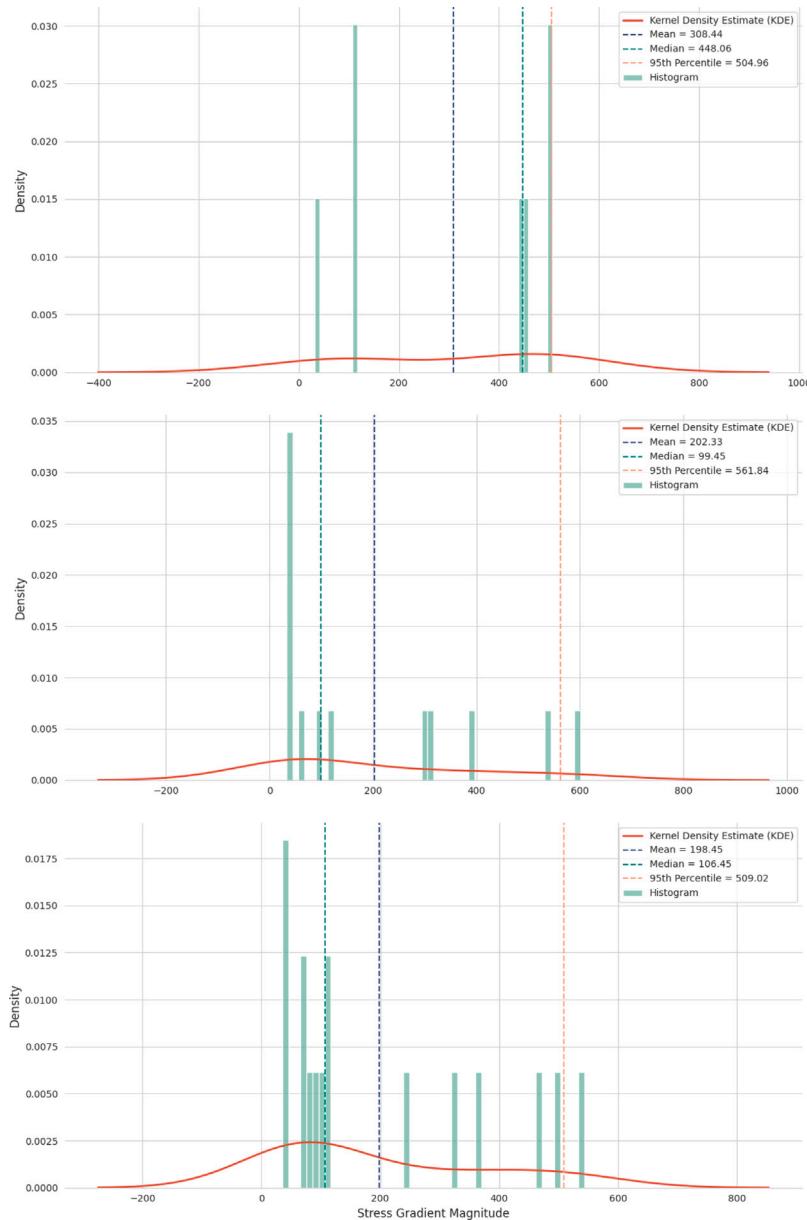


Fig. D.1. The distribution of the mean stress gradient magnitude across subdomains for Case 1 (top), Case 2 (middle), and Case 3 (bottom).

Table D.1
Ablation study on subdomains applied Geo-FNO or NU-FNO.

Subdomain	Case	Method	Metrics	
			MAE	s_E
High-frequency	Case 1	Geo-FNO	0.6678	0.3510
		NU-FNO	1.0738	0.9780
	Case 2	Geo-FNO	0.7973	0.5051
		NU-FNO	1.0543	0.9631
	Case 3	Geo-FNO	0.8265	0.6286
		NU-FNO	1.0475	0.9779
Low-frequency	Case 1	Geo-FNO	0.5638	0.7217
		NU-FNO	0.4443	0.1613
	Case 2	Geo-FNO	0.7200	0.8247
		NU-FNO	0.5967	0.3177
	Case 3	Geo-FNO	0.6197	0.5615
		NU-FNO	0.6015	0.3047

boundary by 93.38% and s_E by 93.63% compared to the baseline model without boundary constraints.

Appendix F. Scalability experiment

We utilize the model trained on a temperature range of -85 °C to 145 °C as the baseline and deploy it to both in-distribution and out-of-distribution datasets. The in-distribution dataset spans a temperature range of 0 °C to 40 °C, while the out-of-distribution dataset spans a temperature range of 150 °C to 200 °C. The results are presented in Table F.1, where the mean absolute error (MAE) is 0.6617 Pa on average, representing a 5.06% reduction compared to the baseline. The standard error (s_E) is reduced by 0.6978 Pa on average, corresponding to a 7.74% reduction compared to the baseline. Table F.2 presents the results for the out-of-distribution data, where the MAE is 0.8421 Pa on average, indicating a 12.68% increase compared to the baseline. The s_E is increased by 0.8423 Pa on average, reflecting an 11.37% increase compared to the baseline.

Table E.1
Ablation study on the boundary loss.

Case	Boundary loss	Subdomain	Metrics	
			MAE	s_E
Case1	+ BL	Boundary	0.8602	0.7036
		Total	0.6167	0.7611
	-BL	Boundary	6.7228	10.3854
		Total	0.5186	0.2354
Case2	+ BL	Boundary	1.0378	0.9233
		Total	0.7667	0.8172
	-BL	Boundary	7.3122	11.0933
		Total	0.6535	0.3177
Case3	+ BL	Boundary	0.8989	0.7800
		Total	0.7076	0.6906
	-BL	Boundary	13.9735	26.3438
		Total	0.6737	0.4197

Table F.1
The result on the In-distribution data.

Case	Subdomain	Metrics	
		MAE	s_E
Case1	High-frequency	0.6582	0.5955
	Low-frequency	0.5527	0.7091
	Total	0.5901	0.6729
Case2	High-frequency	0.7671	0.7013
	Low-frequency	0.7076	0.8124
	Total	0.7270	0.7784
Case3	High-frequency	0.7894	0.7758
	Low-frequency	0.6002	0.5419
	Total	0.6680	0.6422

Table F.2
The result on the Out-of-distribution data.

Case	Subdomain	Metrics	
		MAE	s_E
Case1	High-frequency	0.8466	0.8781
	Low-frequency	0.7378	0.8418
	Total	0.7603	0.8598
Case2	High-frequency	0.8431	0.8162
	Low-frequency	0.8756	0.9541
	Total	0.8450	0.8829
Case3	High-frequency	0.8500	0.8488
	Low-frequency	0.7809	0.7168
	Total	0.7510	0.7843

Appendix G. Sensitive analysis

We have conducted a comprehensive sensitivity analysis of the parameter list (Table 1) as described below.

• **Sensitivity analysis of parameters to influence on stress:** We sample 2000 sets of parameters in the parameter list (Table 1) to generate the temperature loads, then solve the thermal stress by the finite element method. We performed a comprehensive correlation analysis of multiple parameters, including the smooth rate (γ), the standard deviation of Gaussian Random Fields (S), the mean of GRFs (M), the Young's modulus of the material (E), the Poisson's ratio (μ), and the expansion coefficient (α), as shown in Fig. G.1. The figure illustrates the correlation coefficients of these parameters with the maximum stress ($\sigma_{xy,max}$), minimum stress ($\sigma_{xy,min}$), and the mean of stress ($\hat{\sigma}_{xy}$).

– **Maximum Stress ($\sigma_{xy,max}$):** The parameter α exhibits the strongest positive correlation with $\sigma_{xy,max}$, with a correlation coefficient of 0.26. The parameter E also shows a significant positive correlation, with a coefficient of 0.22. Other

Table G.1
The sensitive analysis of parameters influencing error.

Parameters	Value	Metrics	
		MAE	s_E
Expansion coefficient	9.0×10^{-6}	0.5518	0.6207
	1.0×10^{-5}	0.6167	0.7611
	1.1×10^{-5}	0.6816	0.7905
Young's modulus	4.5×10^4	0.5506	0.6205
	5.0×10^4	0.6167	0.7611
	5.5×10^4	0.6840	0.7933
Poisson's ratio	1.8×10^{-1}	0.6002	0.6846
	2.0×10^{-1}	0.6167	0.7611
	2.2×10^{-1}	0.6332	0.7249
Length of plate	1.8×10^1	0.6826	0.9380
	2.0×10^1	0.6167	0.7611
	2.2×10^1	0.6720	1.0462
Height of plate	9.0×10^0	0.5957	0.6581
	1.0×10^1	0.6167	0.7611
	1.1×10^1	0.6245	0.7034
Radius of the holes	4.5×10^{-1}	0.6673	0.7665
	5.0×10^{-1}	0.6167	0.7611
	5.5×10^{-1}	0.6032	0.6654

parameters (γ , S , M , μ) demonstrate weaker or negligible correlations.

- **Minimum Stress ($\sigma_{xy,min}$):** The parameter α has the strongest negative correlation with $\sigma_{xy,min}$, with a coefficient of -0.25. The parameter E shows a moderate negative correlation, with a coefficient of -0.23. Other parameters (γ , S , M , μ) exhibit weaker or negligible correlations.
- **Stress Variation ($\hat{\sigma}_{xy}$):** All parameters exhibit weak correlations with $\hat{\sigma}_{xy}$, with correlation coefficients ranging from -0.04 to 0.03. Among these, the parameter M demonstrates the largest negative correlation (-0.036), while the parameter E shows the highest positive correlation (0.26). In contrast, the parameters S , α , and μ exhibit negligible influence.

- **Sensitivity analysis of parameters to influence on error:** Furthermore, we have performed a systematic sensitivity analysis to evaluate the sensitivity of our method with respect to key parameters, as detailed below.

We employ a single-variable perturbation method to evaluate the sensitivity of our method. The baseline parameters are set as follows: the expansion coefficient is $1.0 \times 10^{-5} / ^\circ\text{C}$, Young's modulus is 5.0×10^4 MPa, Poisson's ratio is 0.2, the length of the plate is 20 cm, the height of the plate is 10 cm, and the radius of the holes is 0.5 cm. To assess the sensitivity of our method, we introduce a 10% perturbation to each parameter individually and analyze the resulting performance, as detailed in Table G.1. These parameters are used to generate the dataset via the finite element method. Subsequently, we train our Hybrid Fourier Neural Operator (HFNO) model on this dataset and evaluate its performance. The average change in mean absolute error (MAE) is 0.0425 Pa, corresponding to a change rate of 6.89%. The maximum change in MAE is 0.0673 Pa, with a change rate of 10.91% observed when Young's modulus is increased to 5.5×10^4 MPa, indicating the highest sensitivity among the tested parameters.

Data availability

Data and source code are available at: <https://github.com/KangruiZhou/HFNO>.

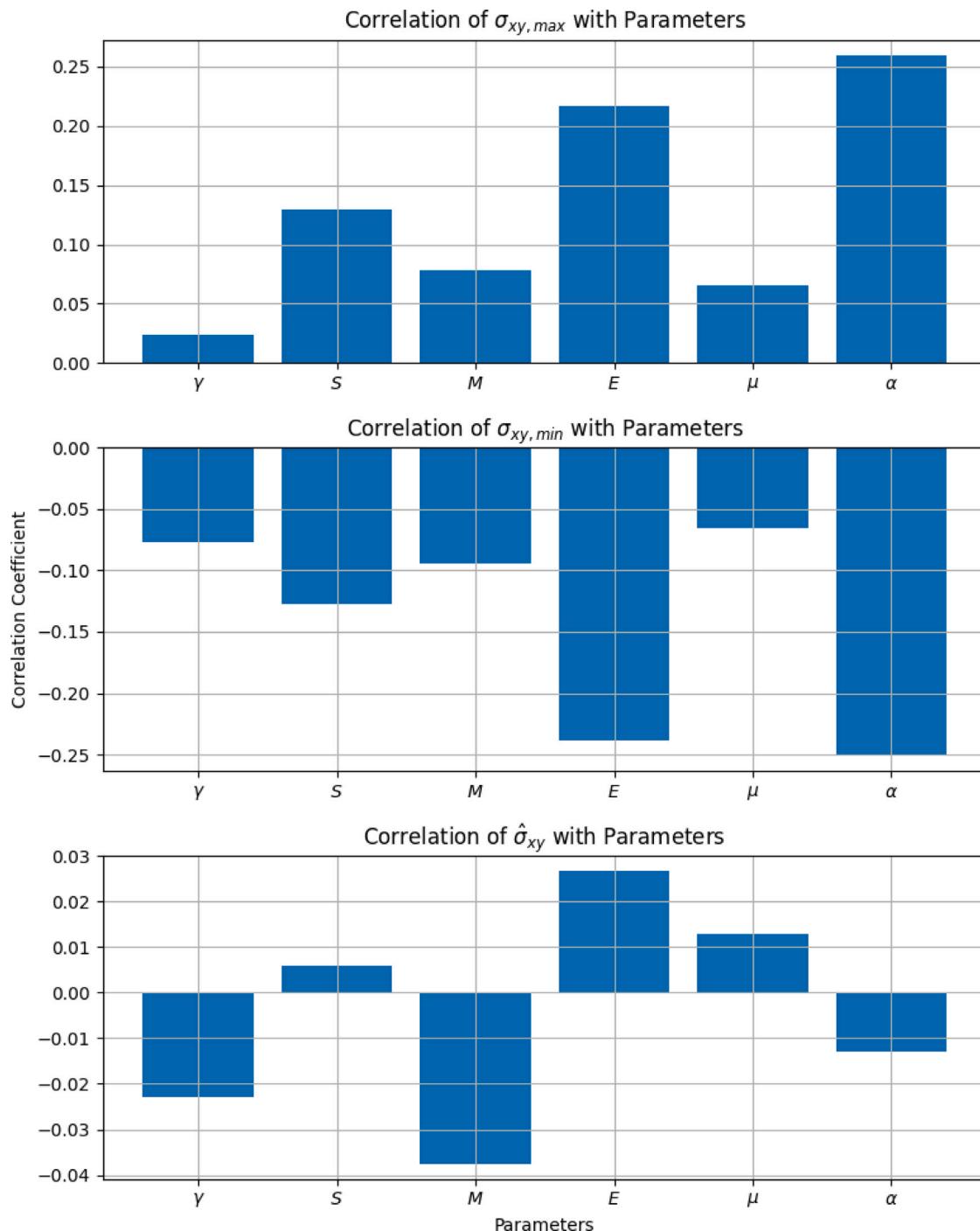


Fig. G.1. Sensitivity analysis of parameters influencing stress, including Gaussian Random Fields (GRFs) and physical properties: the smoothing rate (γ), the standard deviation of GRFs (S), the mean of GRFs (M), the Young's modulus of the material (E), the Poisson's ratio (μ), and the expansion coefficient (α).

References

- Alet, Ferran, Jeewajee, Adarsh Keshav, Villalonga, Maria Bauza, Rodriguez, Alberto, Lozano-Perez, Tomas, Kaelbling, Leslie, 2019. Graph element networks: adaptive, structured computation and memory. In: Chaudhuri, Kamalika, Salakhutdinov, Ruslan (Eds.), Proceedings of the 36th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 97, PMLR, pp. 212–222.
- Alnæs, Martin S., Logg, Anders, Ølgaard, Kristian B., Rognes, Marie E., Wells, Garth N., 2014. Unified form language: A domain-specific language for weak formulations of partial differential equations. ACM Trans. Math. Software 40 (2).
- Alpak, Faruk O., Vamaraju, Janaki, Jennings, James W., Pawar, Suraj, Devarkota, Pandu, Hohl, Detlef, 2023. Augmenting deep residual surrogates with Fourier neural operators for rapid two-phase flow and transport simulations. SPE J. 28 (6), 2982–3003.
- Anandkumar, Anima, Azizzadenesheli, Kamyar, Bhattacharya, Kaushik, Kovachki, Nikola, Li, Zongyi, Liu, Burigede, Stuart, Andrew, 2019. Neural operator: Graph kernel network for partial differential equations. In: ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations.
- Bhatnagar, Saakaar, Afshar, Yaser, Pan, Shaowu, Duraisamy, Karthik, Kaushik, Shailendra, 2019. Prediction of aerodynamic flow fields using convolutional neural networks. Comput. Mech. 64 (2), 525–545.
- Boudjemai, A., Mankour, A., Salem, H., Amri, R., Hocine, R., Chouchaoui, B., 2014. Inserts thermal coupling analysis in hexagonal honeycomb plates used for satellite structural design. Appl. Therm. Eng. 67 (1), 352–361.
- Cao, Shuhao, 2021. Choose a transformer: Fourier or Galerkin. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J. Wortman (Eds.), Advances in Neural Information Processing Systems. vol. 34, Curran Associates, Inc., pp. 24924–24940.

- Cao, Zeyu, Yao, Wen, Peng, Wei, Zhang, Xiaoya, Bao, Kairui, 2022. Physics-informed MTA-UNet: Prediction of thermal stress and thermal deformation of satellites. *Aerospace* 9 (10).
- Chen, Xiaoqian, Chen, Xianqi, Zhou, Weien, Zhang, Jun, Yao, Wen, 2020. The heat source layout optimization using deep learning surrogate modeling. *Struct. Multidiscip. Optim.* 62 (6), 3127–3148.
- Clarke, Stella M., Griebsch, Jan H., Simpson, Timothy W., 2004. Analysis of support vector regression for approximation of complex engineering analyses. *J. Mech. Des.* 127 (6), 1077–1087.
- Corpino, S., Caldera, M., Nichele, F., Masoero, M., Viola, N., 2015. Thermal design and analysis of a nanosatellite in low earth orbit. *Acta Astronaut.* 115, 247–261.
- Coulom, Rémi, 2007. Efficient selectivity and backup operators in Monte-Carlo tree search. In: van den Herik, H. Jaap, Ciancarini, Paolo, Donkers, H. H. L. M. (Jeroen) (Eds.), *Computers and Games*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 72–83.
- Crawford, Roy J., Martin, Peter J., 2020. Chapter 2 - Mechanical behaviour of plastics. In: Crawford, Roy J., Martin, Peter J. (Eds.), *Plastics Engineering* (Fourth Edition), Fourth Edition Butterworth-Heinemann, pp. 59–194.
- guitarist Gilmore, David, 2002. *Spacecraft thermal control handbook*, volume I: Fundamental technologies.
- Goel, Tushar, Hafkta, Raphael T., Shyy, Wei, 2009. Comparing error estimation measures for polynomial and kriging approximation of noise-free functions. *Struct. Multidiscip. Optim.* 38 (5), 429–442.
- Guo, Xiaoxiao, Li, Wei, Iorio, Francesco, 2016. Convolutional neural networks for steady flow approximation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, Association for Computing Machinery, New York, NY, USA, pp. 481–490.
- Hao, Zhongkai, Wang, Zhengyi, Su, Hang, Ying, Chengyang, Dong, Yinpeng, Liu, Songming, Cheng, Ze, Song, Jian, Zhu, Jun, 2023. GNOT: A general neural operator transformer for operator learning. In: Krause, Andreas, Brunskill, Emma, Cho, Kyunghyun, Engelhardt, Barbara, Sabato, Sivan, Scarlett, Jonathan (Eds.), *Proceedings of the 40th International Conference on Machine Learning*. In: *Proceedings of Machine Learning Research*, vol. 202, PMLR, pp. 12556–12569.
- Jurkowski, Artur, Klimanek, Adam, Sladek, Slawomir, 2025. Numerical and experimental study of thermal stabilization system for satellite electronics with integrated phase-change capacitor. *Appl. Therm. Eng.* 258, 124645.
- Kipf, Thomas N., Welling, Max, 2017. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*.
- Kocsis, Levente, Szepesvári, Csaba, 2006. Bandit based Monte-Carlo planning. In: Fürnkranz, Johannes, Scheffer, Tobias, Spiliopoulou, Myra (Eds.), *Machine Learning: ECML 2006*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 282–293.
- Kurth, Thorsten, Subramanian, Shashank, Harrington, Peter, Pathak, Jaideep, Marandi, Morteza, Hall, David, Miele, Andrea, Kashinath, Karthik, Anandkumar, Anima, 2023. FourCastNet: Accelerating global high-resolution weather forecasting using adaptive Fourier neural operators. In: *Proceedings of the Platform for Advanced Scientific Computing Conference. PASC '23*, Association for Computing Machinery, New York, NY, USA.
- Li, Zongyi, Huang, Daniel Zhengyu, Liu, Burigede, Anandkumar, Anima, 2024. Fourier neural operator with learned deformations for PDEs on general geometries. *J. Mach. Learn. Res.* 24 (1).
- Li, Zongyi, Kovachki, Nikola, Azizzadenesheli, Kamyar, Liu, Burigede, Bhattacharya, Kaushik, Stuart, Andrew, Anandkumar, Animashree, 2020a. Fourier neural operator for parametric partial differential equations.
- Li, Zongyi, Kovachki, Nikola, Azizzadenesheli, Kamyar, Liu, Burigede, Bhattacharya, Kaushik, Stuart, Andrew, Anandkumar, Animashree, 2020b. Multipole graph neural operator for parametric partial differential equations.
- Liu, Xiang, Cai, Guoping, 2022. Thermal analysis and rigid-flexible coupling dynamics of a satellite with membrane antenna. *Int. J. Aerosp. Eng.* 2022, 3256825.
- Liu, Songming, Hao, Zhongkai, Ying, Chengyang, Su, Hang, Cheng, Ze, Zhu, Jun, 2023. NUNO: A general framework for learning parametric PDEs with non-uniform data. In: *International Conference on Machine Learning*.
- Lu, Lu, Jin, Pengzhan, Pang, Guofei, Zhang, Zhongqiang, Karniadakis, George Em, 2021. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* 3 (3), 218–229.
- Luo, Guiyang, Wang, Yantao, Zhang, Hui, Yuan, Quan, Li, Jinglin, 2023. AlphRoute: large-scale coordinated route planning via Monte Carlo tree search. *AAAI'23/IAAI'23/EAAI'23*, AAAI Press.
- Maurizi, Marco, Gao, Chao, Berto, Filippo, 2022. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Sci. Rep.* 12 (1), 21834.
- Montenbruck, Oliver, Gill, Eberhard, 2000. *Satellite Orbits: Models, Methods and Applications*, first ed..
- Murali, Haran, 2011. Gaussian random field models for spatial data.
- Raonic, Bogdan, Molinaro, Roberto, Ryck, Tim De, Rohner, Tobias, Bartolucci, Francesca, Alaiafari, Rima, Mishra, Siddhartha, de Bezenac, Emmanuel, 2023. Convolutional neural operators for robust and accurate learning of PDEs. In: *Thirty-Seventh Conference on Neural Information Processing Systems*.
- Sharma, Rishi, Farimani, Amir Barati, Gomes, Joe, Eastman, Peter, Pande, Vijay, 2018. Weakly-supervised deep learning of heat transport via physics informed loss.
- Shen, Zhenxing, Li, Huijian, Liu, Xiaoning, Hu, Gengkai, 2019. Thermal-structural dynamic analysis of a satellite antenna with the cable-network and hoop-truss supports. *Journal of Thermal Stresses* 42, 1–18.
- Sitharam, T.G., Govindaraju, L., 2021. *Theory of Elasticity*, first ed. Springer Singapore.
- Sun, Fangzheng, Liu, Yang, Wang, Jian-Xun, Sun, Hao, 2023. Symbolic physics learner: Discovering governing equations via Monte Carlo tree search. In: *The Eleventh International Conference on Learning Representations*.
- Yao, Wen, Chen, Xiaoqian, Zhao, Yong, van Tooren, Michel, 2012. Concurrent subspace width optimization method for RBF neural network modeling. *IEEE Trans. Neural Netw. Learn. Syst.* 23 (2), 247–259.
- Zhang, Tianze, Trad, Daniel, Innanen, Kristopher, 2023. Learning to solve the elastic wave equation with Fourier neural operators. *Geophysics* 88 (3), T101 – T119.
- Zhang, Yi, Yao, Wen, Ye, Siyu, Chen, Xiaoqian, 2019. A regularization method for constructing trend function in kriging model. *Struct. Multidiscip. Optim.* 59 (4), 1221–1239.
- Zhao, Xiaoyu, Gong, Zhiqiang, Zhang, Jun, Yao, Wen, Chen, Xiaoqian, 2021. A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout. *Struct. Multidiscip. Optim.* 64 (4), 2287–2306.
- Zhengchun, Du, Mengrui, Zhu, Zhiguo, Wang, Jianguo, Yang, 2016. Design and application of composite platform with extreme low thermal deformation for satellite. *Compos. Struct.* 152, 693–703.