

Article

# Physics-Informed MTA-UNet: Prediction of Thermal Stress and Thermal Deformation of Satellites

Zeyu Cao <sup>1,2</sup>, Wen Yao <sup>1,\*</sup>, Wei Peng <sup>1</sup>, Xiaoya Zhang <sup>1</sup> and Kairui Bao <sup>1</sup>

<sup>1</sup> Defense Innovation Institute, Chinese Academy of Military Science, No. 53, Fengtai East Street, Beijing 100071, China

<sup>2</sup> The No. 92941st Troop of PLA, Huludao 125001, China

\* Correspondence: wendy0782@126.com

**Abstract:** The rapid analysis of thermal stress and deformation plays a pivotal role in the thermal control measures and optimization of the structural design of satellites. For achieving real-time thermal stress and thermal deformation analysis of satellite motherboards, this paper proposes a novel Multi-Task Attention UNet (MTA-UNet) neural network which combines the advantages of both Multi-Task Learning (MTL) and U-Net with an attention mechanism. Furthermore, a physics-informed strategy is used in the training process, where partial differential equations (PDEs) are integrated into the loss functions as residual terms. Finally, an uncertainty-based loss balancing approach is applied to weight different loss functions of multiple training tasks. Experimental results show that the proposed MTA-UNet effectively improves the prediction accuracy of multiple physics tasks compared with Single-Task Learning (STL) models. In addition, the physics-informed method brings less error in the prediction of each task, especially on small data sets.

**Keywords:** physics-informed surrogate model; multi-task learning; U-Net; thermal stress and deformation



**Citation:** Cao, Z.; Yao, W.; Peng, W.; Zhang, X.; Bao, K. Physics-Informed MTA-UNet: Prediction of Thermal Stress and Thermal Deformation of Satellites. *Aerospace* **2022**, *9*, 603. <https://doi.org/10.3390/aerospace9100603>

Academic Editor: Vladimir S. Aslanov

Received: 5 September 2022

Accepted: 11 October 2022

Published: 14 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Satellites are important performers of space missions and play an indispensable role in fields such as communication, remote sensing, navigation, and military reconnaissance [1]. Increasingly complicated aerospace missions pose greater challenges to the stability and reliability of satellites [2]. The components installed on satellite motherboards inevitably generate much heat during operation due to their high power density [3]. The heat then leads to a raft of thermal responses such as thermal deformation, thermal buckling, and thermally induced vibration, affecting the accuracy of the satellite mission [4,5]. A ground-based simulation test and numerical analysis are two traditional methods commonly adopted in engineering to analyze thermo-mechanical coupling problems such as thermal stress and thermal deformation of satellites [6–9]. However, these two methods are usually costly and require a long computational time. As a result, in terms of optimization of the design of thermal control measures and structural layout for satellites, the iterative design by the above two methods is likely to face a significant cost and computational burden or even fail to be completed in a given time [10–12]. It is urgent and necessary to conduct a real-time analysis of the thermo-mechanical coupling problems of the satellite. This paper aims to achieve fast and accurate predictions of thermal stress and thermal deformation of satellite motherboards when given a temperature field.

A common method for the aforementioned engineering requirements is to build a surrogate model, which implements a high-precision prediction quickly. Once iterated in the optimization procedure, the surrogate model could achieve a compromise between computational accuracy and computational cost, improving the efficiency of prediction tasks [12–14]. At present, the mainstream surrogate model methods include polynomial-based response surface [15], support vector machine regression [16], radial basis function [17], and Kriging interpolation [18]. However, when dealing with ultra-high

dimensional regression problems, these traditional surrogate modeling methods all face the challenge of the “curse of dimensionality” (i.e., the difficulty in constructing surrogate models between high-dimensional variables). As artificial intelligence advances, many deep learning-based surrogate model methods have been applied to the prediction of physics-related problems, and neural networks such as fully connected neural networks [19], fully convolutional neural networks [20], and conditional generative adversarial networks [21] have been widely used. In particular, the U-Net, as an advanced neural network, is coming into vogue. Rishi Sharma et al. [22] take different boundary conditions as input and applies the U-Net to predict the temperature field in a two-dimensional plane. Cheng et al. [23] analyze the two-dimensional velocity and pressure fields around arbitrary shapes in laminar flow through the U-Net. In addition, Saurabh Deshpande et al. [24] use the U-Net to predict the large deformation response of hyperelastic bodies under loading. These related works prove the potential of U-Net in ultra-high dimensional regression problems.

The focus of the aforementioned deep learning-based surrogate models is to dig a huge number of training data built by finite element or experiments. Nevertheless, these models are driven by data loss only, and they often suffer from high costs of training data or a small number of training data. The physics-informed neural network (PINN) [25] is an effective way to alleviate this problem. In the face of the physics prediction tasks where some existing experience has been accumulated in the real world, the PINN method enables the integration of prior physical information with neural networks, so as to reduce the dependence of neural networks on large amounts of training data. Raissi et al. [25] combine the residual error of PDEs with the loss function of FCN and predict the velocity and pressure fields around a cylinder, while Liu et al. [26] use PINN to solve the forward and inverse problem of the temperature field. In addition, motivated by PINN, the PDE-based loss function can also be explored for surrogate modeling. Bao et al. [27] and Zhao et al. [28] discretize the heat conduction equation by finite difference, making it a loss function.

It is noticed that all the physics prediction models mentioned above are only trained in a Single-Task Learning (STL) manner (i.e., each task is trained separately). When faced with multiple physics prediction tasks, the STL has two drawbacks. On the one hand, the separate training for each sub task takes up much memory space and computational time. On the other hand, if each sub task is modeled separately, the STL tends to ignore the relationship between tasks, such as correlation, conflict, and constraint, making it hard to achieve high precision for the prediction of multiple tasks. Multi-Task Learning (MTL) shows more obvious advantages in dealing with such multiple tasks. In the MTL, multiple tasks share one model, which reduces memory usage and improves inference speed. MTL combines related tasks together and enables the tasks to complement each other by sharing feature information between tasks, which can effectively reduce over-fitting and improve prediction performance [29,30].

Existing MTL is divided into soft parameter sharing and hard parameter sharing. In recent years, there has been a proliferation of research and successful practices in the industrial community on soft parameter sharing, such as the Cross-Stitch [31], PLE [32], ESMM [33] and SNR [34]. In particular, the MMOE [35] proposed by Google has gained great success. It consists of expert networks shared by multiple tasks and task-specific gate networks which can decide what to share. Such a flexible design allows the model to automatically learn how to assign expert parameters based on the relationship of the low-level tasks. The gate networks thus effectively improve the generalization performance of the model and the prediction accuracy across tasks. The MTL has made remarkable achievements in the field of computer vision and has been widely applied in engineering problems such as recommendation systems. However, such MTL models have rarely been used in high-dimensional regression problems with multiple physics tasks.

In this paper, the regression tasks of temperature fields to thermal deformation and thermal stress of satellite motherboards are regarded as mappings between images. To achieve accurate and fast prediction of these multiple physics tasks, three strategies are adopted.

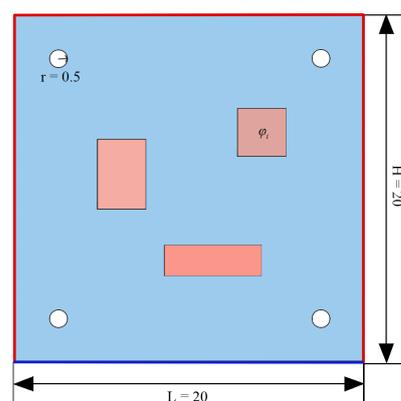
- (1) This paper integrates the advantages of the U-Net and MTL, and proposes the Multi-Task Attention UNet network (MTA-UNet). This network not only shares feature information between high-level layers and low-level layers, but also shares feature information between different tasks. Specifically, it shares the parameters in a targeted way through the attention mechanism. The MTA-UNet effectively reduces the training time of the model and improves the accuracy of prediction compared with STL U-Net.
- (2) A physics-informed approach is applied in training the deep learning-based surrogate model, where the finite difference is applied to discrete thermoelastic and thermal equilibrium equations. The equations are encoded into a loss function to fully exploit the existing physics knowledge.
- (3) Faced with multiple physics tasks, an uncertainty-based loss balancing strategy is adopted to weigh the loss functions of different tasks during the training process. This strategy solves the problem that the training speed and accuracy between different tasks are difficult to balance if trained together, effectively reducing the phenomenon of competition between tasks.

The rest of the paper is structured as follows. Section 2 presents the specific definition of the mathematical model. Section 3 introduces the strategies applied in the construction of deep learning-based MTA-UNet in detail. Section 4 shows the training steps and the experimental results. Section 5 is the conclusion.

## 2. Mathematical Modeling of Thermal Stress and Deformation Prediction

In this paper, as previously defined in [3,11], a two-dimensional satellite motherboard with partial openings is used as a study case. The objective is to realize a rapid prediction of thermal stress and thermal deformation.

The satellite motherboard with four circular holes is shown in Figure 1, and the holes represent screw holes in engineering assembly. There are  $n$  electronic components installed on the motherboard, which generate a large amount of heat during operation and can be regarded as internal heat sources. It is assumed that the temperature of the satellite motherboard changes  $T$  under the joint action of the internal heat sources and external environment. Due to the thermal expansion and contraction, the elastomer tends to undergo thermal deformation, and the deformation trend is restricted to a certain extent because of the external constraints and mutual constraints between various parts of the body. With thermal stress generated inside the elastomer, the thermal stress in turn leads to new additional tension and affects the thermal deformation.



**Figure 1.** Schematic diagram of a two-dimensional satellite motherboard.

In this case, non-displacement boundary conditions are adopted at the edges of the holes. The thermoelastic properties on the motherboard tend to be isotropic, with the parameters consistent with the thermoelastic properties of aluminum. The linear elasticity coefficient of the motherboard is considered to be around the reference temperature  $T_0 = 273$  K and is assumed not to change with temperature. Non-stress and adiabatic

boundary conditions are adopted on the outer boundary of the motherboard. The side length of the rectangular computational domain is set to  $L = H = 20$  cm. The radius of the holes is  $r = 0.5$  cm, and the thermal conductivity coefficient within the domain is  $k = 1$  W/(m × k). The coefficient of linear expansion is  $\alpha = 1 \times 10^{-5}/^\circ\text{C}$ , Young's modulus is  $E = 50 \times 10^3$  MP, and Poisson's ratio is  $\mu = 0.2$ .

The thermoelasticity motherboard is a two-dimensional planar elastomer and its thermal stress components and temperature satisfy the following equations:

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

where  $\sigma_{xx}$  means the x-directional thermal stress,  $\sigma_{yy}$  means the y-directional thermal stress, and  $\sigma_{xy}$  denotes the tangential directional thermal stress. In addition,  $u_x$  and  $u_y$  represent x-directional thermal displacement and y-directional thermal displacement, respectively. According to the differential equation of thermal equilibrium, the  $u_x$  and  $u_y$  satisfy the following set of equations:

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

and the boundary conditions further satisfy:

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

Assuming that the direction of the outer normal to the elastic plane is  $\mathbf{N}$ , then  $l$  and  $m$  can be expressed as the directional cosine of  $\mathbf{N}$  with the  $x$  axis and  $y$  axis, respectively.

Based on the above set of equations and the corresponding parameters, the thermal stress components and the thermal displacement components can be calculated when given an arbitrary temperature field  $T$ . However, the numerical methods are usually costly and time-consuming. In this paper, a deep learning-based surrogate model is constructed to quickly predict five physics components of satellite motherboard,  $u_x$ ,  $u_y$ ,  $\sigma_{xx}$ ,  $\sigma_{yy}$ , and  $\sigma_{xy}$ . The five multiple physics prediction tasks are divided into three multiple sub tasks,  $u_x$  and  $\sigma_{xx}$  in task 1,  $u_y$  and  $\sigma_{yy}$  in task 2, and  $\sigma_{xy}$  in task 3. Three sub tasks are trained together and the temperature field  $T$  is the input of the model.

### 3. Method

This section details the technical strategies proposed during the construction of the deep learning surrogate model. Firstly, on account of multiple physics tasks, to reduce the training cost and improve the prediction performance of the model, we design a neural network structure MTA-UNet in Section 3.1. Subsequently, in Section 3.2, a physics-informed approach is applied to reduce the sample size of training data. Lastly, Section 3.3 adopts an uncertainty-based loss balancing strategy to weight the loss function of multiple tasks, adjusting the training speed and accuracy of multiple tasks. Figure 2 shows the frame diagram of the main technical strategies.

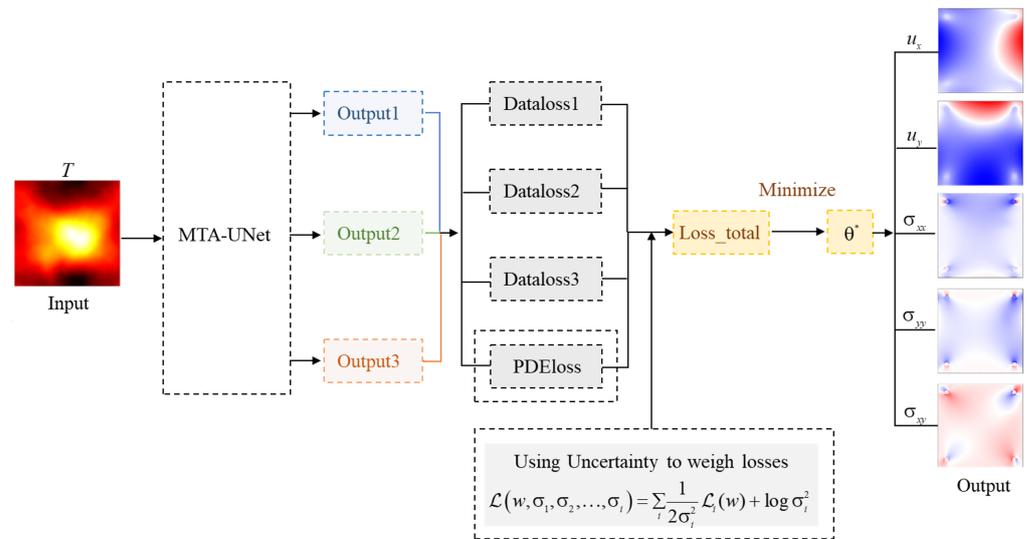


Figure 2. The frame diagram of the main technical strategy.

3.1. MTA-Unet Network Structure

This section describes the MTA-UNet in detail, and Figure 3 shows the actual structure. This model takes the U-Net as the basic architecture, and the improvement is the sharing method between different tasks. In MTA-UNet, multiple sub tasks are trained together and can share feature parameters with other tasks. Different tasks share the coding layer and each of them has its specific decoding layer. In addition, when the features of different layers are concatenated, feature selection is carried out through the Attention Gate (AG). The AG distinguishes which shared features are highly correlated with specific tasks and extracts the relevant shared features from the shared coding layer. The MTA-UNet combines the merits of U-Net and MTL, having the advantage of a “purposeful” combination of feature parameters from different layers and physics tasks. This design fully learns information from the training sets, reducing the training time of the model and memory space of the system and also improving the prediction performance of the model. This section is divided into the architecture of the MTA-UNet (Section 3.1.1), the sharing mechanism of the MTA-UNet (Section 3.1.2), and the Attention Gate of the MTA-UNet (Section 3.1.3).

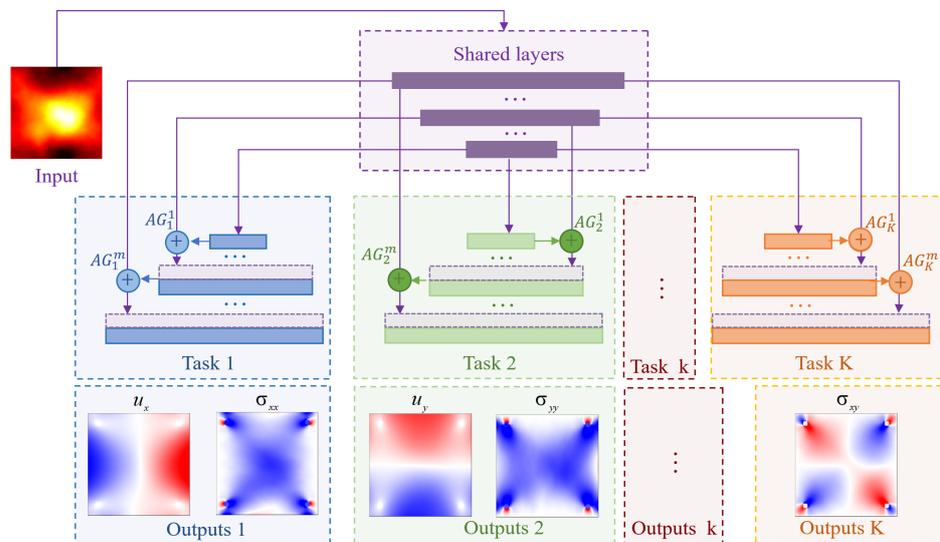
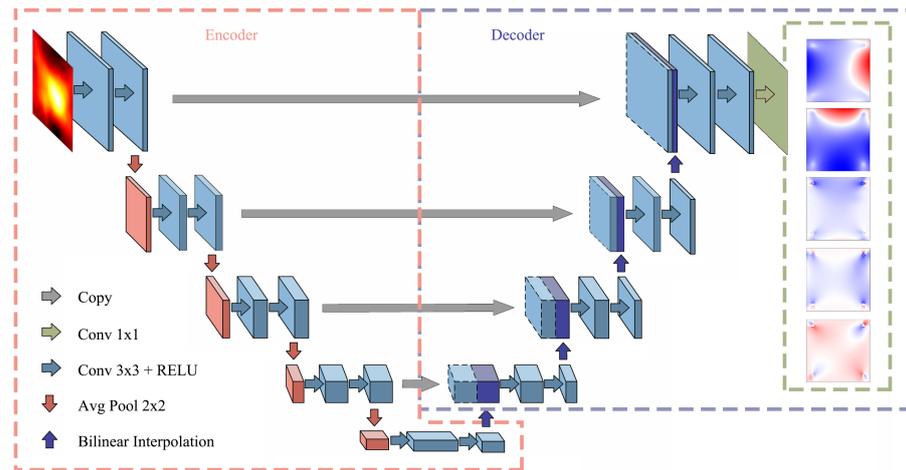


Figure 3. Structure of MTA-UNet.

### 3.1.1. Architecture of the Mta-Unet Network

To fully share feature information of different layers, MTA-UNet adopts the U-Net [36] as its basic architecture. In recent years, the U-Net has exhibited good performance when dealing with various ultra-high dimensional regression problems. According to Figure 4, U-Net has a U-shaped symmetric structure that combines feature parameters of different layers. Owing to this characteristic, the U-Net network is suitable for image-to-image regression tasks as it takes multi-scale feature fusion into account and increases the amount of information.



**Figure 4.** Structure of U-Net.

The MTA-UNet network built in this paper fully retains the advantages of the U-Net network. The first half of MTA-UNet is a classical downsampling process, which is the feature extraction part similar to the coding process, capable of learning multi-scale features of an image through pooling and convolution operations. The second half is an upsampling process similar to the decoding part, capable of restoring the size of the image layer by layer through deconvolution operations. These feature maps are effectively used in subsequent calculations through skip connections.

In MTA-UNet, the downsampling layers are shared layers and the upsampling layers are exclusive to multiple tasks, respectively. The MTA-UNet consists of two repetitive applications of  $3 \times 3$  convolutions (padding = 1), and each convolution is followed by a BatchNorm2d normalized activation function layer and a  $2 \times 2$  max-pooling operation with a stride of 2. We double the number of feature channels in each step of the shared downsampling and, in contrast, halve the number of feature channels in each step of the exclusive upsampling. After the feature selection through the AG, feature parameters of upsampling are skip-connected to downsampling with the corresponding scale of the feature map. In the last layer, each component feature vector is mapped to its corresponding class by a  $1 \times 1$  convolution.

In terms of model training, AdamDelta [37] is selected as the optimizer and the MTA-UNet model is implemented by PyTorch 1.8. The training batch size is set to 16.

### 3.1.2. Feature Sharing Mechanism of Mta-Unet

To enable the sharing of feature parameters across multiple physics tasks, the MTA-UNet uses a partially shared coding layer to enable simultaneous learning of multiple tasks. During the downsampling process, multiple tasks share the coding layer, each having an independent decoding layer. This information concatenation operation of shared and specific feature parameters realizes the deep sharing of multi-task feature parameters.

For the task  $k$  in the given  $K$  tasks,  $k = 1, 2, \dots, K$ . The model consists of the shared downsampling function  $f$ , the exclusive upsampling  $g_k$ , and  $k$  task networks  $h_k$ . The shared

layer follows the input layer, and the task networks are built upon the output of the shared bottom. The  $y_k$  is the output of the task  $k$  and follows the corresponding task-specific tower. Assuming that for the task  $k$ , the  $m$ th downsampling layer of the model is  $f^m$ , and its corresponding upsampling layer is  $g_k^m$ , the output of the task  $k$  is:

$$y_k = h_k(\text{SkipConnect}(AG_k^m(f^m, g_k^m), g_k^m)) \quad (4)$$

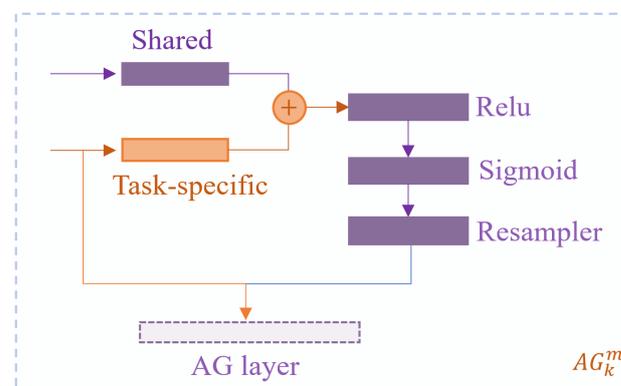
where  $AG_k^m$  denotes the AG operation for the  $m$ th layer of the task  $k$ .

### 3.1.3. Attention Gate of MTA-Unet

Inspired by CNN-based MTL models such as MMOE [35] and PLE [32], the AG operation is added to the MTA-UNet model to realize the selection of shared features. In this way, shared features that are highly relevant to specific tasks are extracted from the shared network layer.

Figure 5 illustrates the AG operation of task  $k$ , where a series of operations such as Relu, Sigmoid, and resampler are performed to evaluate the similarity between  $f^m$  and  $g_k^m$ . The  $m$ th upsampling layer  $f^m$  and its corresponding downsampling layer  $g_k^m$  are input of  $AG_k^m$ . Since the underlay network  $g_k^m$  is generally more accurate, when the similarity between  $f^m$  and  $g_k^m$  is calculated, the features in  $f^m$  with higher similarity with  $g_k^m$  are given higher weights, equals chosen of feature parameters of  $f^m$ . The selection is reflected in the calculation of the weight coefficient, and the operation is presented as:

$$AG_k^m(f^m, g_k^m) = \sum_{i=1}^{L_x} \text{Similarity}(f^m, g_k^m) \times f^m \quad (5)$$



**Figure 5.** Attention Gate operation.

### 3.2. A Physics-Informed Training Strategy

To make full use of the prior physical laws among the physical quantities being predicted, reducing the size of the training sample, a physics-informed strategy is used in the training process. The existing physics knowledge (Equations (1)–(3)) is discretized by the Finite Difference Method (FDM) and integrated into a loss function to construct a physics-informed surrogate model. The FDM is a numerical solution method that expresses basic equations and boundary conditions in the form of function approximation.

It was assumed that an arbitrary two-dimensional elastomer is divided into a uniform grid as shown in Figure 6, in which the intersections of the lines are the nodes.

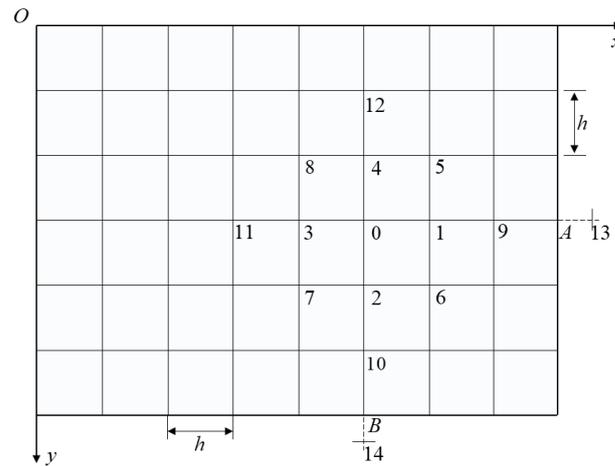


Figure 6. Grid diagram of finite differences.

Let  $f = f(x, y)$  be a continuous physical quantity, which can be expressed as a stress component, displacement component, temperature, etc. Assuming that the grid spacing is sufficiently small, then  $|x - x_0| = h$ . The central difference formulas of the first and second derivatives of  $x$  at node 0 can be obtained:

$$\left(\frac{\partial f}{\partial x}\right)_0 \approx \frac{f_1 - f_3}{2h}, \left(\frac{\partial^2 f}{\partial x^2}\right)_0 \approx \frac{f_1 + f_3 - 2f_0}{h^2} \tag{6}$$

Similarly, the central difference formulas for the first and second derivatives at node 0 in the  $y$  direction are:

$$\left(\frac{\partial f}{\partial y}\right)_0 \approx \frac{f_2 - f_4}{2h}, \left(\frac{\partial^2 f}{\partial y^2}\right)_0 \approx \frac{f_2 + f_4 - 2f_0}{h^2} \tag{7}$$

According to the above equations, the central difference formulas for mixed second derivatives are:

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_0 = \left[\frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y}\right)\right]_0 = \frac{\left(\frac{\partial f}{\partial y}\right)_1 - \left(\frac{\partial f}{\partial y}\right)_3}{2h} \approx \frac{1}{4h^2} [(f_6 + f_8) - (f_5 + f_7)] \tag{8}$$

For boundary points where central difference cannot be used, forward difference or backward difference are used for discretization:

$$\left(\frac{\partial f}{\partial x}\right)_0 \approx \frac{-3f_0 + 4f_1 - f_9}{2h} \approx \frac{3f_0 - 4f_3 + f_{11}}{2h}, \tag{9}$$

$$\left(\frac{\partial f}{\partial y}\right)_0 \approx \frac{-3f_0 + 4f_2 - f_{10}}{2h} \approx \frac{3f_0 - 4f_4 + f_{12}}{2h} \tag{10}$$

To deal with the prediction tasks of thermal stress and thermal displacement, a neural network  $\hat{Y}(x, \theta)$  is constructed, where  $\theta = \{W^\ell, b^\ell\}_{1 \leq \ell \leq D}$ , and  $W^\ell$  is the weight matrix,  $b^\ell$  is the deviation vector in the neural network, and the physics-informed loss function of the neural network is defined as:

$$\mathcal{L}_{\text{PINN}}(\theta) = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{pde}} \tag{11}$$

where  $\mathcal{L}_{\text{data}}$  is the data loss obtained from labeled data with the loss function of Mean Square Error (MSE), and the data loss of the task  $k$  is defined as:

$$\mathcal{L}_{\text{data}_k} = \frac{1}{|\mathcal{P}_f|} \sum_{(x_k, y_k) \in \mathcal{P}_f} (Y_k - \hat{Y}_k)^2 \quad (12)$$

where  $\mathcal{P}_f$  denotes the size of the data set,  $\hat{Y}_k$  represents the prediction of the model, and  $Y_k$  is the ground truth.

The aforementioned finite difference equations (Equations (6)–(10)) are used to discretize the thermoelasticity and the thermal equilibrium equations, with the central difference method for interior points and a forward/backward difference method for boundary points. The discretized PDEs are encoded into a loss function  $\mathcal{L}_{\text{pde}}$  ( $\mathcal{L}_{\text{pde}}$  denotes the physical losses of  $K$  tasks), which is defined as:

$$\mathcal{L}_{\text{pde}} = \frac{1}{|\mathcal{P}_f|} \sum_{(x,y) \in \mathcal{P}_f} \left( \mathcal{L}_{\sigma_{xx\_pde}} + \mathcal{L}_{\sigma_{yy\_pde}} + \mathcal{L}_{\sigma_{xy\_pde}} + \mathcal{L}_{u_{x\_pde}} + \mathcal{L}_{u_{y\_pde}} \right) \quad (13)$$

where  $\mathcal{L}_{u_{x\_pde}}$ ,  $\mathcal{L}_{u_{y\_pde}}$ ,  $\mathcal{L}_{\sigma_{xx\_pde}}$ ,  $\mathcal{L}_{\sigma_{xy\_pde}}$ ,  $\mathcal{L}_{\sigma_{yy\_pde}}$  mean the PDE loss of Equations (1) and (2), respectively. The boundary loss  $\mathcal{L}_{bc}$  is added to the physical loss function matrix  $\mathcal{L}_{\text{pde}}$  in the form of the matrix mask. A convolution operation can also realize the physics-informed strategy.

By training the neural network and minimizing the loss function until the best parameter  $\theta^*$  is found, we construct a physics-informed surrogate model of the thermal deformation and thermal stress of satellite motherboards.

### 3.3. Uncertainty-Based Multi-Task Loss Balancing Strategy

The training goal of the MTA-UNet model is to minimize the difference between the prediction results and the labels. For MTL, however, errors between multiple task features cause competition between multiple tasks. To overcome this seesaw phenomenon, it is an integral part of MTL to balance the training speed and accuracy of multiple tasks through some loss balancing strategies.

Since the loss functions of different tasks tend to have different magnitudes, this paper first normalized their inputs. Subsequently, to solve the problem of competition between different tasks, the most direct approach is to use the weight coefficient  $w_i$  to adjust the proportions of the loss functions of different tasks, as shown in the following equation.

$$\text{Loss} = \sum_k^K w_k \times \text{Loss}_k \quad (14)$$

However, this loss balancing method of fixed weights may not be applicable in some cases as different tasks are very sensitive to the setting of  $w_i$ , and different settings of  $w_i$  vary greatly in terms of performance.

We would like to use a dynamic weight method to balance the loss functions of different tasks. The mainstream methods include GradNorm [38], DWA [39], DTP [40], and Uncertainty [41,42], among which GradNorm consumes a longer computational time as it requires gradient computation, while DWA and DTP need additional weighting operations. This paper adopts the uncertainty-based method to dynamically adjust the weight coefficients for the loss functions of different tasks. The model in this study is constructed based on task-dependent and homoscedastic uncertainty in aleatoric uncertainty [41].

For a certain task  $k$ , the model predicts both a prediction  $y_k$  and the homoscedastic uncertainty  $\sigma_k$  of the model. Thus, the loss function for MTL is defined as:

$$\mathcal{L}(W, \sigma_1, \sigma_2, \dots, \sigma_k) = \sum_k \frac{1}{2\sigma_k^2} \mathcal{L}_k(W) + \log \sigma_k^2 \quad (15)$$

where  $W$  means the weight matrix,  $\mathcal{L}_k(W)$  is the loss function of task  $k$ .  $\sigma_k$  means noise of the model of task  $k$ ,  $\frac{1}{2\sigma_k^2}$  is the weight coefficient of the corresponding task, and  $\log \sigma_k^2$  is the

regularization item to prevent  $\sigma_k$  from learning too large. In this way, higher weights can be given to simple tasks, and the learning of other tasks is driven by them.

For the total of  $K$  regression tasks driven by physics, the overall loss function is defined as:

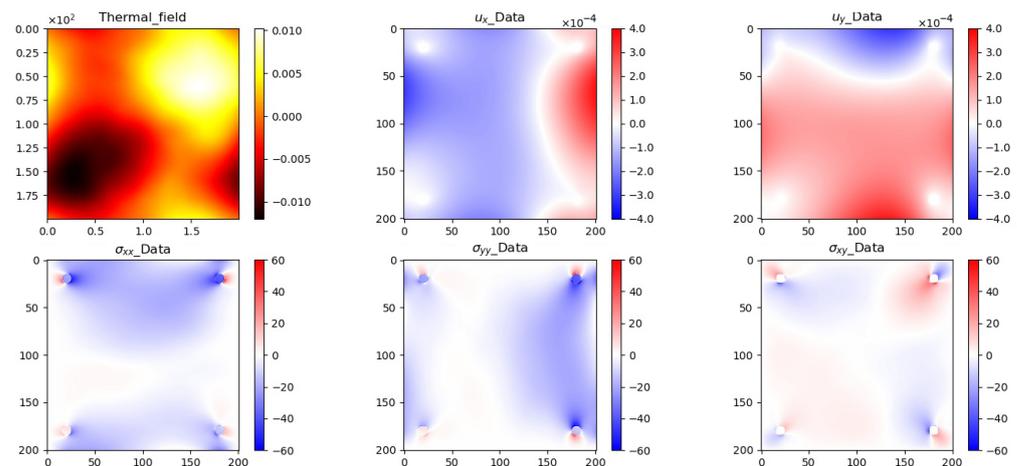
$$\mathcal{L} = \sum_{k=1}^K \left( \frac{1}{2\sigma_k^2} \mathcal{L}_{\text{data}_k}(W) + \log \sigma_k^2 \right) + \frac{1}{2\sigma_{pde}^2} \mathcal{L}_{pde}(W) + \log \sigma_{pde}^2 \quad (16)$$

#### 4. Experiment Results

In this section, the training details and the results of experiments are shown. Section 4.1 introduces the data set and training metrics. Section 4.2 discusses the effect of dynamic loss balancing strategy, MTA-UNet, and the physics-informed method in detail, respectively.

##### 4.1. Training Steps

**Data set:** In MTA-UNet, the input is a two-dimensional planar temperature field on a  $200 \times 200$  uniform grid and the output is five corresponding thermal stress matrices and thermal deformation matrices, including x-directional displacement  $u_x$ , y-directional displacement  $u_y$ , x-directional thermal stress  $\sigma_{xx}$ , y-directional thermal stress  $\sigma_{yy}$ , and tangential thermal stress  $\sigma_{xy}$ . The OpenFOAM software is applied to build the data sets, and the computational domain is a uniform grid. Figure 7 shows one of the data set samples.



**Figure 7.** A sample of data set.

The input temperature field is a Gaussian Random Field (GRF) [43], and the roughness of the temperature field matrix changes when the mean and covariance of GRF are adjusted. This paper generates training sets with sample sizes of 100, 200, 500, 1000, 2000, 5000, and 10,000. Firstly, the prediction of the STL model U-Net and the MTL model MTA-UNet are compared in the training set with a sufficient number of 10,000 samples to confirm the superiority of our model. Subsequently, the MTA-UNet model is trained in training sets with sample sizes of 100, 200, 500, 1000, 2000, and 5000, respectively, to verify the effectiveness of the physics-informed approach. The trained models are tested in a general test set with a sample size of 500.

**Metrics:** The prediction performance of different models for task  $k$  is measured by metrics  $MAE$  and  $MRE$ , defined as:

$$MAE_k = \frac{1}{|\mathcal{P}_f|} \sum_{(x_k, y_k) \in \mathcal{P}_f} |Y_k - \hat{Y}_k| \quad (17)$$

$$MRE_k = \frac{1}{|\mathcal{P}_f|} \sum_{(x_k, y_k) \in \mathcal{P}_f} \frac{|Y_k - \hat{Y}_k|}{\hat{Y}_k} \quad (18)$$

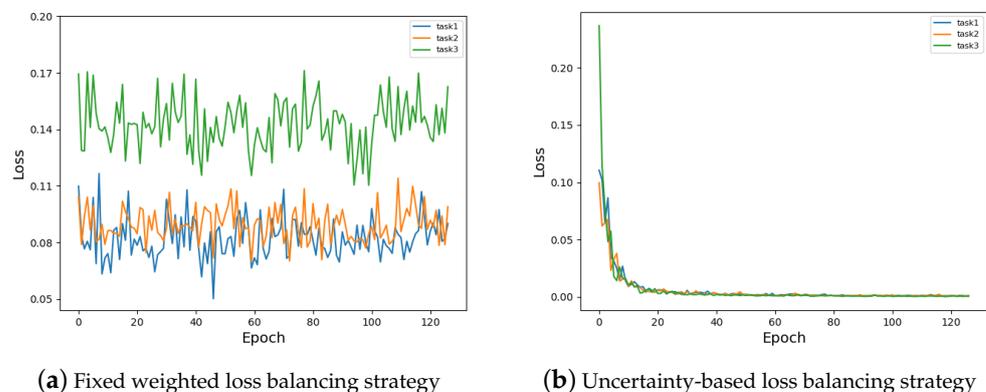
#### 4.2. Performance of Multiple Strategies

As mentioned in Section 3, the loss balancing technique, the MTA-UNet network and the physics-informed training strategy are applied in the construction of the deep learning surrogate model. In this section, we verify the effectiveness of these three techniques separately.

##### 4.2.1. Effectiveness of Dynamic Loss Balancing Strategies

To verify the necessity and effectiveness of the dynamic loss balancing strategy in our MTL regression model, training is carried out with the fixed-weighted loss balancing strategy and uncertainty-based dynamic loss balancing strategy, respectively. Figure 8 shows the training curves. The training curves of the fixed weighted strategy shock in a certain range, making it difficult for training among multiple tasks. While the training curves of dynamic uncertainty-based loss balancing strategy decrease at a stable speed, enabling good prediction for multiple tasks simultaneously. The results indicate that if the loss function of MTL is balanced by fixed weight, multiple tasks will deviate in different directions due to different task objectives.

In this way, the better learning of one task decreases the accuracy of another. Unlike fixed weight, the loss functions of multiple tasks simultaneously decrease at a stable speed with the uncertainty-based dynamic balancing strategy. The dynamic strategy avoids the phenomenon of competition and enables multiple tasks to achieve good performance simultaneously.



**Figure 8.** Training curves of different loss balancing strategies.

##### 4.2.2. Performances of Models

We train the MTA-UNet model and the U-Net model in the training set of 10,000 samples and test them in the general test set, respectively. The MAE and MRE errors obtained from the training are shown in Tables 1 and 2. The  $\sigma_{MAE}$  and  $\sigma_{MRE}$  mean the standard deviations of MAE and MRE, respectively, and the data in bold represent the result of MTA-UNet.

**Table 1.** The MAE of different networks.

Task	MAE		$\sigma_{MAE}$	
	U-Net	MTA-UNet	U-Net	MTA-UNet
$u_x$	$1.9754 \times 10^{-6}$	<b><math>1.1556 \times 10^{-6}</math></b>	$\pm 8.2727 \times 10^{-7}$	<b><math>\pm 1.9903 \times 10^{-7}</math></b>
$u_y$	$2.0947 \times 10^{-6}$	<b><math>1.0721 \times 10^{-6}</math></b>	$\pm 8.3263 \times 10^{-7}$	<b><math>\pm 2.2980 \times 10^{-7}</math></b>
$\sigma_{xx}$	0.0945	<b>0.0713</b>	$\pm 0.01$	<b><math>\pm 0.0097</math></b>
$\sigma_{yy}$	0.0924	<b>0.0728</b>	$\pm 0.01$	<b><math>\pm 0.0098</math></b>
$\sigma_{xy}$	0.0474	<b>0.0402</b>	$\pm 0.01$	<b><math>\pm 0.0087</math></b>

**Table 2.** The MRE of different networks.

Task	MRE/%		$\sigma_{MRE}$	
	U-Net	MTA-UNet	U-Net	MTA-UNet
$u_x$	2.23	<b>1.30</b>	$\pm 0.0056$	<b><math>\pm 0.0044</math></b>
$u_y$	2.39	<b>1.21</b>	$\pm 0.0058$	<b><math>\pm 0.0039</math></b>
$\sigma_{xx}$	1.27	<b>0.96</b>	$\pm 0.0036$	<b><math>\pm 0.0027</math></b>
$\sigma_{yy}$	1.29	<b>0.98</b>	$\pm 0.0039$	<b><math>\pm 0.0027</math></b>
$\sigma_{xy}$	1.36	<b>1.03</b>	$\pm 0.0038$	<b><math>\pm 0.0030</math></b>

As for the comparison between the STL U-Net and the MTL MTA-UNet model proposed, it can be seen in Table 1 that the MTA-UNet exhibits better prediction performance on each physical prediction task by fully sharing feature information between different tasks. Especially in  $u_x$ , the proposed MTL MTA-UNet model reduces the MAE by at least 41% lower than the STL U-Net model in our task. Since different tasks have different noise, the sharing of feature parameters between tasks by MTA-UNet cancels part of the noise in different directions to a certain extent, improving the prediction accuracy. According to variances of MAE and MRE in Table 2 for different models, the MTA-UNet model is more robust than U-Net. The sharing mechanism in MTA-UNet plays the role of data enhancement. Moreover, some parameters required in a single task are better trained by other tasks, and the sharing of features between different tasks make sense in this respect. The effectiveness of the MTA-UNet can be attributed to three reasons. Firstly, implicit data augmentation. The simultaneous training of multiple tasks effectively increases the number of training instances. As different tasks have different noise patterns, a model that learns several tasks simultaneously is able to learn more general representations. Secondly, the eavesdropping mechanism. With the sharing in MTA-UNet, we can allow the model to eavesdrop features that are hard to learn from other tasks. Thirdly, the regularization mechanism. In MTA-UNet, Multi-Task Learning plays the role of regularization by introducing an inductive bias, which reduces the risk of overfitting.

In addition, we investigated the prediction performance. According to Table 2, the minimum per-pixel MRE of the proposed MTA-UNet model is only 0.96%, and the model shows good performance in predicting thermal stress and thermal deformation of satellite motherboards. Figure 9 shows the prediction result of MTA-UNet. The mean per-pixel error is calculated with high precision according to the ground truth, demonstrating the feasibility of using the MTA-UNet model for regression between ultra-high dimensional variables.

Then, the training cost was discussed. This deep learning surrogate model can reduce the prediction time of thermal stress and thermal deformation from 2 min to 0.23 s, effectively saving the time cost, which also verifies the effectiveness of the deep learning-based thermal stress and thermal deformation surrogate model.

#### 4.2.3. Effects of the Physics-Informed Strategy

To verify the contributions of the physics-informed strategy to the accuracy of the model and sample size, the MTA-UNet surrogate model is trained in data sets with different

sample sizes and tested in a general test set, and the comparison of  $MRE$  between data-driven only and the physics-informed is shown in Table 3. The  $\sigma_{MRE}$  means the standard deviation of  $MRE$ , and the data in bold represent the result of the physics-informed strategy.

According to the experimental results, the physics-informed surrogate model exhibits better performance than the data-driven only in data sets with different sample sizes. Relatively speaking, the physics-informed strategy improves more significantly in the case of small samples, and the gap between the data-driven strategy and the physics-informed strategy gradually narrows as the number of samples increases, which demonstrates the potential of the physics-informed strategy in cases with small sample size.

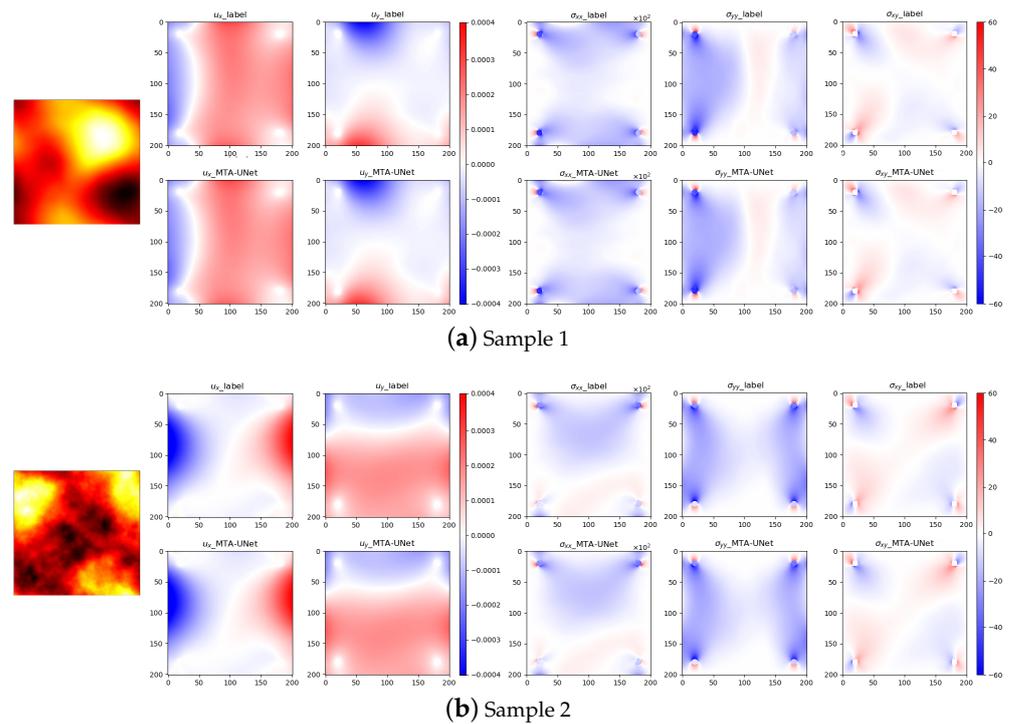


Figure 9. Prediction of the MTA-UNet model.

Table 3.  $MRE$  of training strategies on dataset with different scales.

Scale	Method	$MRE/\%$					$\sigma_{MRE}$				
		$u_x$	$u_y$	$\sigma_{xx}$	$\sigma_{yy}$	$\sigma_{xy}$	$u_x$	$u_y$	$\sigma_{xx}$	$\sigma_{yy}$	$\sigma_{xy}$
200	Data	6.46	6.42	4.43	4.55	5.52	$\pm 0.0421$	$\pm 0.0396$	$\pm 0.0145$	$\pm 0.0143$	$\pm 0.0200$
	PDE	<b>6.07</b>	<b>6.02</b>	<b>4.02</b>	<b>4.17</b>	<b>5.11</b>	<b><math>\pm 0.0305</math></b>	<b><math>\pm 0.0297</math></b>	<b><math>\pm 0.0103</math></b>	<b><math>\pm 0.0102</math></b>	<b><math>\pm 0.0142</math></b>
500	Data	3.72	3.76	2.77	2.78	2.93	$\pm 0.0185$	$\pm 0.0193$	$\pm 0.0083$	$\pm 0.0088$	$\pm 0.0104$
	PDE	<b>3.48</b>	<b>3.50</b>	<b>2.37</b>	<b>2.45</b>	<b>2.66</b>	<b><math>\pm 0.0095</math></b>	<b><math>\pm 0.0096</math></b>	<b><math>\pm 0.0065</math></b>	<b><math>\pm 0.0056</math></b>	<b><math>\pm 0.0093</math></b>
1000	Data	3.11	3.19	1.99	2.01	2.29	$\pm 0.0114$	$\pm 0.0115$	$\pm 0.0056$	$\pm 0.0057$	$\pm 0.0086$
	PDE	<b>2.96</b>	<b>2.98</b>	<b>1.82</b>	<b>1.78</b>	<b>2.09</b>	<b><math>\pm 0.0073</math></b>	<b><math>\pm 0.0073</math></b>	<b><math>\pm 0.0046</math></b>	<b><math>\pm 0.0044</math></b>	<b><math>\pm 0.0071</math></b>
2000	Data	2.62	2.65	1.55	1.61	1.72	$\pm 0.0089$	$\pm 0.0092$	$\pm 0.0049$	$\pm 0.0052$	$\pm 0.0065$
	PDE	<b>2.49</b>	<b>2.51</b>	<b>1.41</b>	<b>1.40</b>	<b>1.53</b>	<b><math>\pm 0.0063</math></b>	<b><math>\pm 0.0062</math></b>	<b><math>\pm 0.0044</math></b>	<b><math>\pm 0.0042</math></b>	<b><math>\pm 0.0044</math></b>
5000	Data	2.28	2.41	1.19	1.25	1.41	$\pm 0.0057$	$\pm 0.0069$	$\pm 0.0039$	$\pm 0.0040$	$\pm 0.0046$
	PDE	<b>2.17</b>	<b>2.30</b>	<b>1.11</b>	<b>1.12</b>	<b>1.22</b>	<b><math>\pm 0.0048</math></b>	<b><math>\pm 0.0052</math></b>	<b><math>\pm 0.0029</math></b>	<b><math>\pm 0.0030</math></b>	<b><math>\pm 0.0031</math></b>

In addition, we conducted training on a smaller size of samples. Figure 10 shows the prediction obtained by the data-driven strategy and the physics-informed strategy, respectively, for 100 samples. For the data-driven results in Figure 10, we can find a clear difference compared to the label, but the physics-informed results are closer, especially in the near-boundary domain. The comparison shows that physics-informed strategy

obtains predictions that are more in line with the practical meaning of physics and realistic working conditions.

Through the physics-informed loss function, the prediction results are guided in the direction that conforms to the physical law, which provides the possibility of being consistent with physical laws, reducing the search space of deep learning surrogate models. What's more, compared with the data-driven model, the deep learning model that follows the physical characteristics is more likely to be generalized to the case outside the sample distribution.

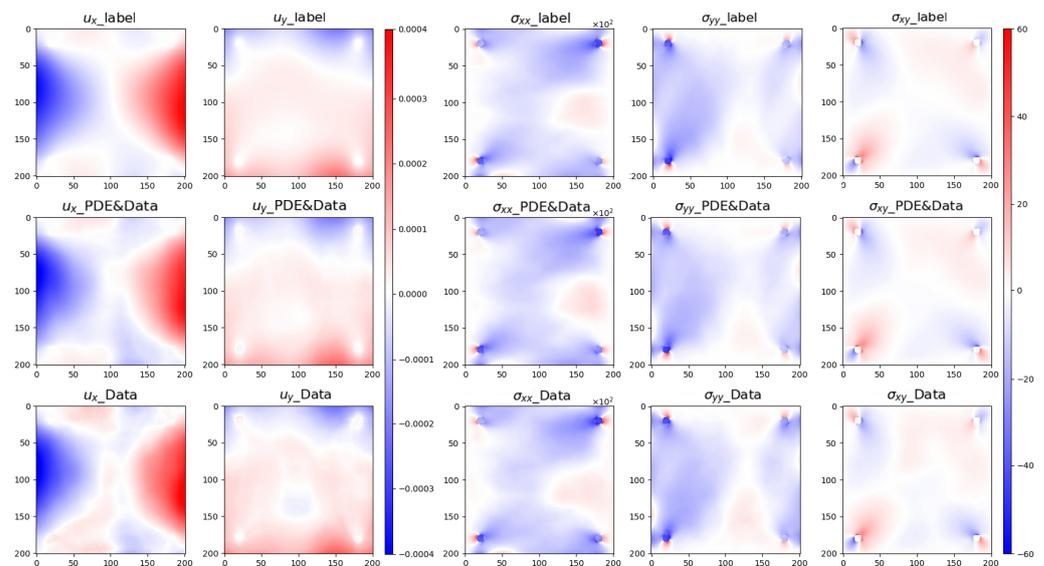


Figure 10. Predictions of the data-driven strategy and the physics-informed strategy.

## 5. Conclusions

To solve the regression problems of multiple physical quantities in thermal-mechanical coupled fields, this paper proposes a novel MTL network MTA-UNet. The MTA-UNet integrates the strengths of U-Net and MTL, having the advantage of selective sharing parameters of different tasks and layers. With shared encoding layers, task-specific decoding layers, and AG feature filtering operations of MTA-UNet, the correlation between different tasks is fully learned. The performance of the deep learning-based surrogate model is greatly enhanced, and the training time and system memory are also effectively decreased. In our problem, the proposed MTL MTA-UNet model reduces the *MAE* by at least 41% lower than the STL U-Net. Subsequently, an uncertainty-based dynamic loss-weighted strategy is used among loss functions of different tasks. With this strategy, the training speed and accuracy of each sub task are well balanced. In addition, a physics-informed method is applied during training to rapidly predict the thermal stress and thermal deformation of satellite motherboards. In the process, a set of PDEs is encoded into a loss function, making full use of the prior physics knowledge. Compared with the data-driven only strategy, the physics-informed method exhibits better performance in training sets with different sample sizes and obtains solutions more consistent with the laws of physics. Experiments demonstrate the framework we used reduces the *MRE* by at least 11% in our problem.

Despite the MTA-UNet neural network we constructed for the regression problems of multiple physical quantities, further research is still required. As our model structure is strongly general, it is worth exploring strategies to enhance the balance between multiple tasks further. Extension to more complex boundaries and more complicated nonlinear problems is also valuable. Furthermore, embedding physics knowledge into the network structure remains a challenging work.

**Author Contributions:** Conceptualization, Z.C., W.Y. and W.P.; methodology, Z.C. and W.P.; software, Z.C. and W.P.; validation, Z.C., W.Y. and X.Z.; formal analysis, W.Y. and X.Z.; investigation, Z.C. and X.Z.; resources, Z.C., W.Y. and W.P.; data curation, Z.C. and W.P.; writing—original draft preparation, Z.C.; writing—review and editing, Z.C., K.B., W.P. and X.Z.; visualization, Z.C. and K.B.; supervision, X.Z. and K.B.; project administration, W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (Nos. 11725211 and 52005505).

**Data Availability Statement:** The code of neural network and data generator can be downloaded at: <https://github.com/KomorebiTso/MTA-UNet> (accessed on 3 September 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Montenbruck, O.; Gill, E.; Lutze, F. Satellite orbits: Models, methods, and applications. *Appl. Mech. Rev.* **2002**, *55*, B27–B28. [[CrossRef](#)]
2. Kodheli, O.; Lagunas, E.; Maturo, N.; Sharma, S.K.; Shankar, B.; Montoya, J.F.M.; Duncan, J.C.M.; Spano, D.; Chatzinotas, S.; Kisseleff, S.; et al. Satellite communications in the new space era: A survey and future challenges. *IEEE Commun. Surv. Tutor.* **2020**, *23*, 70–109. [[CrossRef](#)]
3. Chen, X.; Chen, X.; Zhou, W.; Zhang, J.; Yao, W. The heat source layout optimization using deep learning surrogate modeling. *Struct. Multidiscip. Optim.* **2020**, *62*, 3127–3148. [[CrossRef](#)]
4. Du, Z.; Zhu, M.; Wang, Z.; Yang, J. Design and application of composite platform with extreme low thermal deformation for satellite. *Compos. Struct.* **2016**, *152*, 693–703. [[CrossRef](#)]
5. Liu, X.; Cai, G. Thermal analysis and rigid-flexible coupling dynamics of a satellite with membrane antenna. *Int. J. Aerosp. Eng.* **2022**, *2022*, 3256825. [[CrossRef](#)]
6. Stohlman, O.R. Coupled radiative thermal and nonlinear stress analysis for thermal deformation in large space structures. In Proceedings of the 2018 AIAA Spacecraft Structures Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 0448.
7. Azadi, E.; Fazelzadeh, S.A.; Azadi, M. Thermally induced vibrations of smart solar panel in a low-orbit satellite. *Adv. Space Res.* **2017**, *59*, 1502–1513. [[CrossRef](#)]
8. Johnston, J.D.; Thornton, E.A. Thermally induced dynamics of satellite solar panels. *J. Spacecr. Rocket.* **2000**, *37*, 604–613. [[CrossRef](#)]
9. Shen, Z.; Li, H.; Liu, X.; Hu, G. Thermal-structural dynamic analysis of a satellite antenna with the cable-network and hoop-truss supports. *J. Therm. Stress.* **2019**, *42*, 1339–1356. [[CrossRef](#)]
10. Zhao, X.; Gong, Z.; Zhang, J.; Yao, W.; Chen, X. A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout. *Struct. Multidiscip. Optim.* **2021**, *64*, 2287–2306. [[CrossRef](#)]
11. Chen, X.; Yao, W.; Zhao, Y.; Chen, X.; Zheng, X. A practical satellite layout optimization design approach based on enhanced finite-circle method. *Struct. Multidiscip. Optim.* **2018**, *58*, 2635–2653. [[CrossRef](#)]
12. Cuco, A.P.C.; de Sousa, F.L.; Silva Neto, A.J. A multi-objective methodology for spacecraft equipment layouts. *Optim. Eng.* **2015**, *16*, 165–181. [[CrossRef](#)]
13. Chen, X.; Liu, S.; Sheng, T.; Zhao, Y.; Yao, W. The satellite layout optimization design approach for minimizing the residual magnetic flux density of micro-and nano-satellites. *Acta Astronaut.* **2019**, *163*, 299–306. [[CrossRef](#)]
14. Yao, W.; Chen, X.; Ouyang, Q.; Van Tooren, M. A surrogate based multistage-multilevel optimization procedure for multidisciplinary design optimization. *Struct. Multidiscip. Optim.* **2012**, *45*, 559–574. [[CrossRef](#)]
15. Goel, T.; Hafkta, R.T.; Shyy, W. Comparing error estimation measures for polynomial and kriging approximation of noise-free functions. *Struct. Multidiscip. Optim.* **2009**, *38*, 429–442. [[CrossRef](#)]
16. Clarke, S.M.; Griebisch, J.H.; Simpson, T.W. Analysis of support vector regression for approximation of complex engineering analyses. *J. Mech. Des.* **2005**, *127*, 1077–1087. [[CrossRef](#)]
17. Yao, W.; Chen, X.; Zhao, Y.; van Tooren, M. Concurrent subspace width optimization method for rbf neural network modeling. *IEEE Trans. Neural Netw. Learn. Syst.* **2011**, *23*, 247–259.
18. Zhang, Y.; Yao, W.; Ye, S.; Chen, X. A regularization method for constructing trend function in kriging model. *Struct. Multidiscip. Optim.* **2019**, *59*, 1221–1239. [[CrossRef](#)]
19. Zakeri, B.; Monsefi, A.K.; Darafarin, B. Deep learning prediction of heat propagation on 2-D domain via numerical solution. In Proceedings of the 7th International Conference on Contemporary Issues in Data Science, Zanjan, Iran, 6–8 March 2019; pp. 161–174.
20. Edalatifar, M.; Tavakoli, M.B.; Ghalambaz, M.; Setoudeh, F. Using deep learning to learn physics of conduction heat transfer. *J. Therm. Anal. Calorim.* **2021**, *146*, 1435–1452. [[CrossRef](#)]
21. Farimani, A.B.; Gomes, J.; Pande, V.S. Deep learning the physics of transport phenomena. *arXiv* **2017**, arXiv:170902432.
22. Sharma, R.; Farimani, A.B.; Gomes, J.; Eastman, P.; Pande, V. Weakly-supervised deep learning of heat transport via physics informed loss. *arXiv* **2018**, arXiv:180711374.

23. Chen, J.; Viquerat, J.; Hachem, E. U-net architectures for fast prediction of incompressible laminar flows. *arXiv* **2019**, arXiv:191013532.
24. Deshpande, S.; Lengiewicz, J.; Bordas, S. Fem-based real-time simulations of large deformations with probabilistic deep learning. *arXiv* **2021**, arXiv:211101867.
25. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
26. Liu, X.; Peng, W.; Gong, Z.; Zhou, W.; Yao, W. Temperature field inversion of heat-source systems via physics-informed neural networks. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104902. [[CrossRef](#)]
27. Bao, K.; Yao, W.; Zhang, X.; Peng, W.; Li, Y. A physics and data co-driven surrogate modeling approach for temperature field prediction on irregular geometric domain. *arXiv* **2022**, arXiv:220308150.
28. Zhao, X.; Gong, Z.; Zhang, Y.; Yao, W.; Chen, X. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. *arXiv* **2021**, arXiv:210912482.
29. Zhang, Y.; Yang, Q. An overview of multi-task learning. *Natl. Sci. Rev.* **2018**, *5*, 30–43. [[CrossRef](#)]
30. Zhang, Y.; Yang, Q. A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.* **2021**. [[CrossRef](#)]
31. Misra, I.; Shrivastava, A.; Gupta, A.; Hebert, M. Cross-stitch networks for multi-task learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3994–4003.
32. Tang, H.; Liu, J.; Zhao, M.; Gong, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, Virtual Event, 22–26 September 2020; pp. 269–278.
33. Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; Gai, K. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 1137–1140.
34. Ma, J.; Zhao, Z.; Chen, J.; Li, A.; Hong, L.; Chi, E.H. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton Hawaiian Village, Honolulu Hawaii, USA, 27 January–1 February 2019; Volume 33, pp. 216–223.
35. Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; Chi, E.H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1930–1939.
36. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
37. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:12125701.
38. Chen, Z.; Badrinarayanan, V.; Lee, C.Y.; Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 794–803.
39. Liu, S.; Johns, E.; Davison, A.J. End-to-end multi-task learning with attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1871–1880.
40. Guo, M.; Haque, A.; Huang, D.A.; Yeung, S.; Li, F.-F. Dynamic task prioritization for multitask learning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 270–287.
41. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
42. Xiang, Z.; Peng, W.; Zheng, X.; Zhao, X.; Yao, W. Self-adaptive loss balanced physics-informed neural networks for the incompressible navier-stokes equations. *arXiv* **2021**, arXiv:210406217.
43. Haran, M. Gaussian random field models for spatial data. In *Handbook of Markov Chain Monte Carlo*; Chapman and Hall/CRC: New York, USA, 2011; pp. 449–478.