Parallel Physics-Informed Neural Networks with Bidirectional Balance

Yuhao Huang*

*Beijing Jiaotong University, Beijing, China
21121552@bjtu.edu.cn

Abstract—As an emerging technology in deep learning, physics-informed neural networks (PINNs) have been widely used to solve various partial differential equations (PDEs) in engineering. However, PDEs based on practical considerations contain multiple physical quantities and complex initial boundary conditions, thus PINNs often returns incorrect results. Here we take heat transfer problem in multilayer fabrics as a typical example. It is coupled by multiple temperature fields with strong correlation, and the values of variables are extremely unbalanced among different dimensions. We clarify the potential difficulties of solving such problems by classic PINNs, and propose a parallel physics-informed neural networks with bidirectional balance. In detail, our parallel solving framework synchronously fits coupled equations through several multilayer perceptions. Moreover, we design two modules to balance forward process of data and back-propagation process of loss gradient. This bidirectional balance not only enables the whole network to converge stably, but also helps to fully learn various physical conditions in PDEs. We provide a series of ablation experiments to verify the effectiveness of the proposed methods. The results show that our approach makes the PINNs unsolvable problem solvable, and achieves excellent solving accuracy.

Index Terms—Machine learning, Physics-informed neural networks, Heat transfer, Coupled differential equations

I. INTRODUCTION

With the rapid development of universal data and computing resources, deep learning technology has produced abundant achievements in different subjects [1,3]. Due to the universal approximation capability of neural networks, related technologies have also been affecting the fields of computing science and engineering. As early as the 1990s, artificial neural network learned convective heat transfer coefficient from data [4]. Further, Owhadi [5] exploited prior knowledge to make an attempt in numerical homogenization problem. Convolutional neural network was also used for explore heat transport properties of turbulent Rayleigh-Benard convection [6]. Obviously, these supervised learning tasks cannot be used for regular numerical solving, because the ground truth are often unknown. In subsequent studies, Raisi et al. [7] proposed physics-informed neural networks (PINNs), in which the losses are derived from physical conditions. This pioneering work has achieved plenty of remarkable results, including hydrodynamics [8,10] bioengineering [11,12], high-dimensional PDEs [13,14] and etc.

However, PINNs usually cannot be correctly solved in practical engineering problems, especially when the data distribution is highly uneven [9,15]. Wang et al. [16] believed that gradient pathology will occur when data showed high-frequency. They proposed a self-adaptive method which adjust weights of different terms in composite loss function. Although this method improves accuracy significantly, it cannot generalize to other possible data situation. Just like the common heat transfer of clothing in daily life [17], it is accompanied by data imbalance between multiple dimensions. PINNs will have abnormal training feedback with extremely high losses and unrealistic results in this situation, and there is no relevant research. Hence, we take this problem in a more extreme environment as a prime example to tap more potential of PINNs.

Compared with the development of PINNs, the study of numerical heat transfer is obviously much more mature and complete. At first, Torvi and Dale [18] clearly summarized heat transfer process in thin fabric under high heat flux. Chitrphiromsri and Kuznetsov [19] analyzed simultaneous heat and moisture transfer through fabric. Further, Zhu and Zhang [20] added pyrolysis of fabrics and related shrinkage on the basis of Torvi’s work. More comprehensively, Ghazy and Bergstrom [21] fully considered the heat transfer in air gap. As we can see, when the related mathematical models are getting closer and closer to reality, traditional numerical methods are also getting bloated. On the contrary, PINNs is usually not affected by the complexity of equations due to its special mechanism, that it transforms the iterative solving into the parameters learning. It shows great potential in numerical heat transfer. In this way, it is very valuable to solve the training pathology in PINNs.

Based on the above discussion, we conduct research on the application of PINNs to thermal protective clothing. Our specific contributions can be summarized in the following points:

- Our analysis points out that the failure of PINNs is related to the numerical imbalance in data forward and gradient back-propagation processes.
- We solve the coupled PDEs system in parallel by combining multiple neural sub-networks, which has stronger fitting ability.
- We propose the Forward Balance Module which maintain the forward value of the overall network units within a reasonable range. Correspondingly, the Backward Balance Module is proposed to pay balanced attention to
various physical constraints by scaling losses.
- Our approach can solve the thermal protective clothing problem with high performance, making the PINNs unsolvable problem solvable.

Taken together, our development provides a new perspective for the training of constrained neural networks, which can help us endow deep learning tools with prior knowledge and reduce barriers in extending to other scientific fields.

II. PRELIMINARIES

In this section, we will briefly review the classic PINNs and the mathematical model of thermal protective clothing. These preliminaries will help the subsequent theoretical analysis and method proposed.

A. Physics-informed Neural Networks

PINNs is designed to solve forward and inverse problems of PDEs. Generally speaking, the mixed initial-boundary value problem can be summarized as:

\[
\begin{align*}
    u_t + \mathcal{N}_x[u] &= 0, & x \in \Omega, & t \in [0, T] \\
    u(x, 0) &= h(x), & x \in \Omega \\
    u(x, t) &= g(x, t), & t \in [0, T], & x \in \partial \Omega
\end{align*}
\]  

where, \((x, t)\) is the coordinates in the finite computational domain; \(u\) is the solution of this PDE; \(\mathcal{N}_x\) is a general linear or nonlinear differential operator in control equation; The initial and boundary condition can be expressed as in (1b)-(1c). This typical PDEs covers a range of problems in mathematical physics, including conservation laws, diffusion processes and advection-diffusion-reaction systems.

According to the original work of PINNs [17], we can fit analytical solution \(u(x, t)\) by neural network with parameters \(\theta\) (weight or bias). The network takes discrete points \((x, t)\) as inputs and corresponding value \(f_\theta(x, t)\) as outputs. The learning of parameters is constrained by the physical properties in PDEs (control equations, initial conditions, boundary conditions), which can be defined as the following residual.

\[
\begin{align*}
    u_t(x, t) &= \frac{\partial}{\partial t}f_\theta(x, t) + \mathcal{N}_x[f_\theta(x, t)] \\
    u_o(x, 0) &= f_\theta(x, 0) - h(x) \\
    u_b(x, t) &= f_\theta(x, t) - g(x, t)
\end{align*}
\]  

The partial derivative term can be quickly calculated by automatic differential technique [22]. Thus, \(\theta\) can be trained by minimizing a composite loss function as follows.

\[
\begin{align*}
    \mathcal{L}(\theta) := \sum_i \mathcal{L}_i(\theta) & \quad \text{(3a)} \\
    \mathcal{L}_r &= \frac{1}{N_r} \sum_{j=1}^{N_r} \left[ u_r(x^r_j, t^r_j) \right]^2 & \quad \text{(3b)} \\
    \mathcal{L}_0 &= \frac{1}{N_0} \sum_{j=1}^{N_0} \left[ u_0(x^0_j, 0) \right]^2 & \quad \text{(3c)} \\
    \mathcal{L}_b &= \frac{1}{N_b} \sum_{j=1}^{N_b} \left[ u_b(x^b_j, t^b_j) \right]^2 & \quad \text{(3d)}
\end{align*}
\]

Here, \((x^r_j, 0)\) denotes the data under the initial conditions, \((x^b_j, t^b_j)\) denotes the data under the boundary conditions, and \((x^r_j, t^r_j)\) denotes the remaining data in the entire computational domain. All loss terms used mean square error loss as in (3b)-(3d).

Based on these, PINNs uses soft constraints to make predictions that satisfy any conditions derived from physical law such as symmetry, invariance and conservation.

B. Heat Transfer Model of Multi-layer Fabric

A typical thermal protective clothing [17] is a multilayer arrangement (usually 3 layers: outer shell, moisture barrier and thermal liner) as shown in Fig. 2. The outer surface of the shell is exposed to flash fire, and energy is transmitted to the shell by thermal convection. Further, heat passes through other fabric layers by conduction. In order to clearly introduce the focus of this manuscript, we simplify the heat transfer problem to one-dimensional and assume that the materials of each layer are isotropic. According to Fourier’s Law, we can build control equations for thermal diffusion reaction as follow [18].

\[
\begin{align*}
    C^{A \text{shl}} \frac{\partial T_{\text{shl}}}{\partial t} &= k_{\text{shl}} \frac{\partial^2 T_{\text{shl}}}{\partial x^2} & \quad \text{(4)} \\
    C^{A \text{msr}} \frac{\partial T_{\text{msr}}}{\partial t} &= k_{\text{msr}} \frac{\partial^2 T_{\text{msr}}}{\partial x^2} & \quad \text{(5)} \\
    C^{A \text{lin}} \frac{\partial T_{\text{lin}}}{\partial t} &= k_{\text{lin}} \frac{\partial^2 T_{\text{lin}}}{\partial x^2} & \quad \text{(6)}
\end{align*}
\]

where, the subscripts shl, msr and lin indicate the outer shell, the moisture barrier and the thermal liner, respectively; \(T\) is the
temperature value (K); \(x\) and \(t\) are spatial (m) and temporal (s) coordinates, respectively; \(C^A\) is the apparent heat capacity \((J/(m^3 \cdot K))\); \(k\) is the thermal conductivity \((W/(m \cdot K))\).

Assuming that \(T_0\) is the normal temperature, the coupled system meets the same initial condition as follow:

\[
T_{\text{shl}}|_{t=0} = T_{\text{msr}}|_{t=0} = T_{\text{lin}}|_{t=0} = T_0
\]

For the left and right boundary of the whole system, i.e. the outer surface of the outer shell and the inner surface of the thermal liner, there is convective heat transfer from air. The difference is that the outer shell is in contact with the hot gases, while the thermal liner is in contact with the ambient environment. According to Newton’s law of cooling, the Neumann boundary conditions are given as follows:

\[
-k_{\text{shl}} \frac{\partial T_{\text{shl}}}{\partial x} \bigg|_{x=0} = h_g (T_g - T_{\text{shl}}|_{x=0})
\]

\[
-k_{\text{lin}} \frac{\partial T_{\text{lin}}}{\partial x} \bigg|_{x=L_{\text{shl}}} = h_{\text{air}} (T_{\text{lin}}|_{x=L_{\text{shl}}} - T_0)
\]

where, \(L_{\text{shl}}\) is the total thickness of all fabric layers; \(h_g\) and \(h_{\text{air}}\) are the convective heat transfer coefficients from flame to the outer shell and from ambient to the thermal liner, respectively.

Since the boundary between adjacent layers satisfies the same temperature and heat flux, we can obtain the following Dirichlet and Neumann boundary conditions for adjacent layers [21].

\[
T_{\text{shl}}|_{x=L_{\text{shl}}} = T_{\text{msr}}|_{x=L_{\text{shl}}}
\]

\[
-k_{\text{shl}} \frac{\partial T_{\text{shl}}}{\partial x} \bigg|_{x=L_{\text{shl}}} = -k_{\text{msr}} \frac{\partial T_{\text{msr}}}{\partial x} \bigg|_{x=L_{\text{shl}}}
\]

\[
T_{\text{msr}}|_{x=L_{\text{msr}}} = T_{\text{lin}}|_{x=L_{\text{msr}}}
\]

\[
-k_{\text{msr}} \frac{\partial T_{\text{msr}}}{\partial x} \bigg|_{x=L_{\text{msr}}} = -k_{\text{lin}} \frac{\partial T_{\text{lin}}}{\partial x} \bigg|_{x=L_{\text{msr}}}
\]

In summary, the heat transfer model in multilayer fabric is established by such fixed solution problem as in [4]-[13].

### III. Methodology

The failure of PINNs is often accompanied by abnormal loss values and unrealistic prediction results. In this section, we deeply analyze the causes of this failure and why our methods can work successfully.

#### A. Parallel Solving Framework

As mentioned earlier, heat transfer model in multilayer fabric is actually the coupled PDEs problem. Each fabric layer cannot be viewed as an independent heat transfer system due to the conditions that the temperature and heat flux of adjacent boundary are consistent. On the other hand, we can’t construct a well-posed problem without these boundary conditions [23].

In order to solve this kind of coupled PDEs problem, we propose the Parallel Solving Framework (PSF) of PINNs as shown in Fig. 3 which fits the temperature function of each layer in parallel to make the interdependent boundary conditions update dynamically in the training process. For three fabrics, we give three MLP to fit the temperature field respectively. These sub-networks use the same structure, but their parameters are not shared. Similar to (3a), the residual terms of all equations in the coupled PDEs are involved in the composite loss function as follow, where \(\mathcal{L}(\cdot)\) is mean square error loss function. You can find the details for this part in Appendix A.

\[
\text{LOSS} = \sum \mathcal{L}(u^l_{\text{layer}})
\]

\[
= \mathcal{L}(u^l_1) + \mathcal{L}(u^l_2) + \mathcal{L}(u^l_3) + \mathcal{L}(u^l_{\text{shl&mrs}}) + \mathcal{L}(u^l_{\text{shl&mrs}}) + \mathcal{L}(u^l_{\text{shl&mrs}}) + \mathcal{L}(u^l_{\text{shl&mrs}})
\]

According to different loss terms, the calculating domain is divided into multiple sub-domains. Then, we use the gradient descent algorithm to optimize the parameters of all sub-networks simultaneously. It is worth noting that this PINNs framework is a global unsupervised training, which only relies on existing constraints derived from physical laws to learn.

The loss terms of adjacent layer, such as \(\mathcal{L}(u^l_{\text{shl&mrs}})\), are slightly different from other terms, because their gradients back-propagation will inevitably affect two adjacent sub-networks, which allows communication between sub-networks. Such a mechanism provides a fault-tolerant space for the initial stage of unsupervised training, and further enables all sub-networks to adjust and learn together.

---

**Fig. 3. Overview of parallel solution framework (PSF).** In the data preparation stage, the input grid will be divided into several sub-domains. These datasets will be input into the parallel network to generate the corresponding solutions, and the PDEs constraints will be obtained by the automatic differentiation technique (auto-diff), and then the total loss function will be calculated. Through iterative training, a global prediction solution with the expected accuracy is obtained.
**B. Forward Balance Module**

As the input of PINNs, the fabric thickness is usually a few tenths of a millimeter, and the flash time can reach tens of seconds. But as the output, the temperature can reach several hundred Kelvin theoretically. Such numerical imbalance brings higher requirements for neural networks in large-scale mapping and nonlinear fitting.

Despite the existence of the universal approximation theorem [24], not any network structure can meet our needs. The activation function profoundly affects the performance of the shallow network [25]. Among them, Tanh-like functions restrict the output to a small range, while ReLU-like functions restrict the gradient through linear units to alleviate gradient disappearance. Obviously, the former is not conducive to large-scale mapping, and the latter has poor nonlinear fitting ability in shallow networks.

Based on this, the Forward Balance Module (FBM) scales the data pairs losslessly to a similar range, so that the shallow network with Tanh that has sufficient nonlinear fitting capabilities does not need to focus on large-scale mapping. Specifically, the FBM first transforms the physical base-units of input in PDEs. The unit of length $x$ is converted from meter (m) to millimeter (mm), and the unit of time $t$ remains unchanged. Then, we use equation (soft constraint) transformation to indirectly turn the unit of the output temperature value $T$ from Kelvin (K) to kilokelvin (kK). Taking the outer shell as an example, the change of the residual of the control equation is given as follow.

$$
u_{r}^{\text{shl}} = C_{\text{shl}}^A \frac{\partial T_{\text{shl}}}{\partial t} - k_{\text{shl}} \frac{\partial^2 T_{\text{shl}}}{\partial x^2}$$

$$= C_{\text{shl}}^A \frac{\partial}{\partial t} (T_{\text{shl}}^r \times 10^3) - k_{\text{shl}} \frac{\partial^2}{\partial (x')^2} (T_{\text{shl}}^r \times 10^{-3})^2$$

$$= C_{\text{shl}}^A \frac{\partial T_{\text{shl}}^r}{\partial t} \times 10^3 - k_{\text{shl}} \frac{\partial^2 T_{\text{shl}}^r}{\partial x'^2} \times 10^9$$  \hspace{1cm} (15)$$

FBM is placed in the data preparation stage of the overall solving framework (see Fig. 4). It balances the values in the subsequent neural network just like normalization or nondimensionalization. The difference is that FBM maintains equivalence compared to normalization, and can still retain physical meaning compared to nondimensionalization.

**C. Backward Balance Module**

The PINNs-based solving approach relies on soft constraints derived from physical laws. Such constraints directly affect the gradient flow, making the network parameters update in the direction of gradient descent. This process can be described as the following forward Euler discretization.

$$\theta_{n+1} = \theta_n - \eta \nabla \theta_n \text{LOSS} = \theta_n - \eta \sum \nabla \theta_n \mathcal{L} (u^\text{layer}_i)$$ \hspace{1cm} (16)$$

where, $\eta$ is the learning rate, and the gradient $\nabla \theta_n \mathcal{L} (u^\text{layer}_i)$ usually increases as the loss term $\mathcal{L} (u^\text{layer}_i)$ increases. When the residual terms $u^\text{layer}_i$ is unbalanced during training, the gradient flow will be affected by this, causing the neural network to pay more attention to larger terms [16]. It is well known that PDEs may have infinitely many solutions without proper initial-boundary conditions [26], so PINNs with learning imbalance often returns incorrect predictions.

Back to the heat transfer problem to explore the imbalance in the backward process of the network. By comparing expressions in Appendix A, we can find that so many residual terms can also be summarized as following partition. where, the elements of $C_1$ all contain $T$ term, $C_2$ all contain $\frac{\partial T}{\partial t}$ term, and $C_3$ all contain $\frac{\partial^2 T}{\partial x^2}$ term.

$$C_1 = \{ u_{b_0}^{\text{shl}} , u_{b_1}^{\text{shl}} , u_0^{\text{msr}} , u_0^{\text{lin}} , u_1^{\text{msr}} , u_1^{\text{lin}} \} \hspace{1cm} (17a)$$

$$C_2 = \{ u_{b_2}^{\text{shl}} , u_{b_2}^{\text{msr}} , u_2^{\text{msr}} , u_2^{\text{lin}} , u_b \} \hspace{1cm} (17b)$$

$$C_3 = \{ u_{b_3}^{\text{shl}} , u_{b_3}^{\text{msr}} , u_3^{\text{lin}} \} \hspace{1cm} (17c)$$

Such a rule indicates the existence of learning imbalance. To quantify this imbalance, We define the range of $C_j$ as range ($C_j$) = $\left[ \min \{ \mathcal{L} (C_j) \} , \max \{ \mathcal{L} (C_j) \} \right]$, where $\mathcal{L} (C) = \{ \sum_{k=1}^{n_{\text{exp}}} \mathcal{L} (u_j)_{k}/n_{\text{exp}} \mid u_j \in C \}$ (i.e., a set composed of the
mean square error of each element in $C$ under $n_{\text{exp}}$ repeated experiments. If above three classes are balanced, their their range ($C_i$) should also highly overlap. We define this degree of overlap with the help of Intersection of Union (IOU) as in (18). When the IOU is closer to 1, the degree of overlap between sets is higher, and vice versa.

$$\text{IOU}_{ij}(k_1, k_2) = \frac{\text{range}(k_1 \cdot C_i) \cap \text{range}(k_2 \cdot C_j)}{\text{range}(k_1 \cdot C_i) \cup \text{range}(k_2 \cdot C_j)}$$

Based on these definitions, we propose the Backward Balance Module (BBM), which multiplies the three types of sets with scaling coefficients $\alpha, \beta, \gamma$ to make $\text{IOU}_{12}(\alpha, \beta) \rightarrow 1$ and $\text{IOU}_{13}(\alpha, \gamma) \rightarrow 1$. As shown in Fig. 4, the BBM is usually placed between the residual and the loss calculation. It balances the gradient flow in the back-propagation through the scaling operation, so that the network can fully learn the different physical laws in PDEs.

IV. NUMERICAL RESULTS

In this section, we provide a series of experiments aimed at evaluating the solving performance in different PINNs structures. We develop a benchmark in all situations:

First, we assume thermophysical properties and initial conditions of each fabric layer are shown in Tables I and II. Secondly, we divide the three fabric layers into 50, 70, and 200 segments respectively through a uniform grid, and divide whole time domain into 300 segments. This setting can generate a training dataset (network input coordinates) with size of (300×320). Then, the backbone is an MLP with 4 hidden layers with 10 channels. Lastly, this benchmark uses the default Adam optimizer [27] (learning rate is 0.001) and the Kaiming random initialization method [28].

### TABLE I

| Material      | Density ($\text{kg/m}^3$) | Specific heat ($\text{J/(kg \cdot K)}$) | Heat conductivity ($\text{W/(m \cdot K)}$) | Thickness (mm) |
|---------------|---------------------------|---------------------------------------|------------------------------------------|----------------|
| Outer shell   | 300                       | 1377                                  | 0.082                                    | 0.6            |
| Moisture barrier | 862                      | 2100                                  | 0.37                                     | 0.85           |
| Thermal liner | 74.2                      | 1726                                  | 0.045                                    | 3.6            |

### TABLE II

| Property                              | Value          |
|---------------------------------------|----------------|
| Normal temperature                    | 310.15 K       |
| Hot gases temperature                 | 2000 K         |
| Flame convective heat transfer coefficient | 40 W/(m$^2$K) |
| Air convective heat transfer coefficient | 9.496 W/(m$^2$K) |

If the scaling coefficient is too small, the loss will be scaled to near 0, which can also make IOU approach 1. At this time, various losses have reached some kind of balance, but too small loss value may cause vanishing gradient. This is obviously not what we want. To this end, we need to limit the scaling coefficients. Based on Table III, we set $\alpha = 1 \times 10^{-2}$. The optimal value of other coefficients can be obtained by linear search (see Fig. 5). When $\beta \approx 1.27 \times 10^{-4}$ and $\gamma \approx 4.72 \times 10^{-8}$, the value of IOU reaches maximum, and the three classes of loss almost achieve maximum balance.

Fig. 5. Schematic of linear search results. The red dot indicates the optimal value.

Fig. 6. Box plot of initial loss distribution after using BBM. The ordinate is the loss value under repeated experiments, and the abscissa is the loss item in different classes.
To further visualize the impact of BBM, we track the gradient distribution of each loss during training. In addition, we also track the gradient distribution of each loss during training. Fig. 6 and 7 show that such a set of scaling coefficients effectively plays a balancing role, and enables the gradient flow of the training process to keep a good learning state.

According to the poor performance of the M2-M4 models, we can know that the bidirectional balance is critical to successfully solving heat transfer problem of multilayer fabric. Using either FBM or BBM alone can slightly improve accuracy, but only when this two coexist can it produce a reliable and strong balance constraint in PINNs.

Achieving M1 performance requires not only a suitable balance strategy, but also the high fitting capabilities of the parallel solving framework. Table IV also shows the accuracy gap between the M5 and the M1 (use PSF or not). Fig. 5d more intuitively reflects the superiority of using PSF in the coupled problem. As analyzed in Section III-B, the temperature gradient between adjacent layers is not continuous and not constant. Therefore, the multiple networks and parallelism of PSF play a key role in fitting such sudden change. In conclusion, these experiments strongly proved the feasibility of our methods and emphasized that the maximum effect can only be achieved when these methods coexist.

### V. Summary and Outlook

In this paper, we describe the problems of classical PINNs in a more general PDEs, and propose Parallel Physics-Informed Neural Networks with Bidirectional Balance to solve it. Specifically, we focus on heat transfer problem in multilayer fabrics. On the one hand, this problem is composed of the coupled PDEs corresponding to multilayer fabrics, which brings a challenge to network fitting. On the other hand, the input and output physical quantities of the problem show a huge difference in magnitude, and the many conditions in the coupling system exacerbate this imbalance. Whether in the forward or backward process of the network, this imbalance will profoundly hinder training. For this reason, we propose a parallel solution framework to solve the difficulty of fitting coupled equations. Furthermore, two balancing strategies are proposed to be applied to the forward and backward processes respectively, so that the network can be trained correctly. The bidirectional balance method is proposed systematically for the first time. From the perspective of final performance, our approach is very successful in the heat transfer problem under different parameter settings.

Despite there are enough theories to support our approach to other complex problems, we have not yet conducted empirical investigations on this. In addition, we use fully connected neural networks for all sub-networks, but we can also use other...
advanced network structures, such as convolutional neural networks or attention mechanism. These modifications can further improve the accuracy of the solution.

**APPENDIX A**

**EXPRESSION OF THE RESIDUAL TERMS**

According to the mathematical model of heat transfer of multilayer fabric, here we will give the expression of each residual term in the network, and thus derive the complex compound loss function in Section II-A of the paper. In addition, since the residual terms are affected by the forward balance module (FBM) mentioned in Section III-B, we will also give their transformed expression. First, we regard the physical base-units transformation as follows:

\[
T = T' \times 10^3 \\
x = x' \times 10^{-3}
\]

Section II-B of the paper analyzed the heat conduction and heat convection processes in the model. According to the existing physical equations, we can derive the residual terms and their transformation as follows:

1) Initial conditions:

\[
\begin{align*}
\bar{u}_0^{\text{sh}} &= T_{\text{sh}}|_{t=0} - T_0 = (T'_0|_{t=0} - T_0) \times 10^3 \\
\bar{u}_0^{\text{msr}} &= T_{\text{msr}}|_{t=0} - T_0 = (T'_0|_{t=0} - T_0) \times 10^3 \\
\bar{u}_0^{\text{lin}} &= T_{\text{lin}}|_{t=0} - T_0 = (T'_0|_{t=0} - T_0) \times 10^3
\end{align*}
\]

2) Thermal diffusion control equations:

\[
\begin{align*}
\bar{u}_r^{\text{sh}} &= C_{\text{sh}} A \frac{dT_{\text{sh}}}{dt} - k_{\text{sh}} \frac{\partial^2 T_{\text{sh}}}{\partial x^2} \\
&= C_{\text{sh}} A \frac{dT_r'}{dt} \times 10^3 - k_{\text{sh}} \frac{\partial^2 T_r'}{\partial x'^2} \times 10^9 \\
\bar{u}_r^{\text{msr}} &= C_{\text{msr}} A \frac{dT_{\text{msr}}}{dt} - k_{\text{msr}} \frac{\partial^2 T_{\text{msr}}}{\partial x^2} \\
&= C_{\text{msr}} A \frac{dT_r'}{dt} \times 10^3 - k_{\text{msr}} \frac{\partial^2 T_r'}{\partial x'^2} \times 10^9 \\
\bar{u}_r^{\text{lin}} &= C_{\text{lin}} A \frac{dT_{\text{lin}}}{dt} - k_{\text{lin}} \frac{\partial^2 T_{\text{lin}}}{\partial x^2} \\
&= C_{\text{lin}} A \frac{dT_r'}{dt} \times 10^3 - k_{\text{lin}} \frac{\partial^2 T_r'}{\partial x'^2} \times 10^9
\end{align*}
\]

3) Boundary conditions of outer surfaces:

\[
\begin{align*}
\bar{u}_b^{\text{sh}} &= -k_{\text{sh}} \frac{\partial T_{\text{sh}}}{\partial x} \bigg|_{x=0} = -h_g (T_g - T_{\text{sh}}|_{x=0}) \\
&= -k_{\text{sh}} \frac{\partial T_r'}{\partial x'} \bigg|_{x'=0} \times 10^6 \\
&= -h_g (T_g' - T_{\text{sh}}|_{x'=0}) \times 10^3 \\
\bar{u}_b^{\text{msr}} &= -k_{\text{msr}} \frac{\partial T_{\text{msr}}}{\partial x} \bigg|_{x=L_{\text{sh}}} = -h_{\text{air}} (T_{\text{lin}}|_{x=L_{\text{sh}}} - T_0) \\
&= -k_{\text{msr}} \frac{\partial T_r'}{\partial x'} \bigg|_{x'=L_{\text{sh}}} \times 10^6 \\
&= -h_{\text{air}} (T_g' - T_{\text{msr}}|_{x'=L_{\text{sh}}}) \times 10^3 \\
\bar{u}_b^{\text{lin}} &= -k_{\text{lin}} \frac{\partial T_{\text{lin}}}{\partial x} \bigg|_{x=L_{\text{sh}}} = -h_{\text{air}} (T_{\text{lin}}|_{x=L_{\text{sh}}} - T_0) \\
&= -k_{\text{lin}} \frac{\partial T_r'}{\partial x'} \bigg|_{x'=L_{\text{sh}}} \times 10^6 \\
&= -h_{\text{air}} (T_g' - T_{\text{lin}}|_{x'=L_{\text{sh}}}) \times 10^3
\end{align*}
\]

4) Boundary conditions of adjacent surfaces:

\[
\begin{align*}
\bar{u}_{b1}^{\text{sh} \& \text{msr}} &= T_{\text{sh}}|_{x=L_{\text{sh}}} - T_{\text{lin}}|_{x=L_{\text{sh}}} = -h_{\text{air}} (T_{\text{lin}}|_{x=L_{\text{sh}}} - T_0)
\end{align*}
\]
During this time, the temperature changes drastically, which is a challenge to our proposed numerical solving method. To verify this performance, we add experiments with various time settings. Other settings are the same as M1 in the original text. The solving accuracy and 2D visualization results are shown below.

### Appendix B

**Additional Experiments**

In extreme flash fire environments, even thermal protective clothing can only be effective for the first tens of seconds. During this time, the temperature changes drastically, which is a challenge to our proposed numerical solving method. To verify this performance, we add experiments with various time settings. Other settings are the same as M1 in the original text. The solving accuracy and 2D visualization results are shown below.

#### Table V

| Time | Mean Square Error between Ground Truth and Prediction |
|------|------------------------------------------------------|
|      | 10s | 30s | 60s | 120s |
| SHL  | 5.2382 × 10⁻⁶ | 1.5055 × 10⁻⁵ | 5.8359 × 10⁻⁵ | 3.1514 × 10⁻⁴ |
| MSR  | 2.3912 × 10⁻⁶ | 5.1216 × 10⁻⁶ | 8.2250 × 10⁻⁵ | 3.7955 × 10⁻⁴ |
| LIN  | 1.8996 × 10⁻⁵ | 1.2241 × 10⁻⁴ | 1.9701 × 10⁻⁴ | 1.7403 × 10⁻³ |
| Total| 1.3214 × 10⁻⁵ | 7.9884 × 10⁻⁵ | 1.5024 × 10⁻⁴ | 1.2199 × 10⁻³ |

### References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, no. 2, 2012.

[2] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *Computer Science*, 2014.

[3] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of dna- and rna-binding proteins by deep learning,” *Nature Biotechnology*, 2015.

[4] K. Jambunathan, S. L. Hartle, S. Ashforth-Frost, and V. N. Fontama, “Evaluating convective heat transfer coefficients using neural networks,” *International Journal of Heat Mass Transfer*, vol. 39, no. 11, pp. 2329–2332, 1996.

[5] H. Owhadi, “Bayesian numerical homogenization,” *Multiscale Modeling and Simulation*, vol. 13, no. 3, pp. 812–828, 2015.

[6] E. Fonda, A. Pandey, J. Schumacher, and K. R. Sreenivasan, “Deep learning in turbulent convection networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 18, pp. 8667–8672, 2019.

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

[8] M. Raissi, A. Yazdani, and G. E. Karniadakis, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations.” *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.

[9] L. Sun, H. Gao, S. Pan, and J.-X. Wang, “Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data,” *Computer Methods in Applied Mechanics and Engineering*, vol. 361, p. 112732, 2019.

[10] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, “Deep learning of vortex-induced vibrations,” *Journal of Fluid Mechanics*, vol. 861, pp. 119–137, 2019.

[11] F. S. Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, “Physics-informed neural networks for cardiac activation mapping,” *Frontiers of Physics in China*, vol. 8, p. 42, 2020.

[12] G. Kissas, Y. Yang, E. Hwang, W. R. Witschey, J. A. Detre, and P. Perdikaris, “Machine learning in cardiovascular flows modeling: Predicting arterial blood pres-
sure from non-invasive 4d flow mri data using physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering*, vol. 358, p. 112623, 2020.

[13] J. Sirignano and K. Spiliopoulos, “Dgm: A deep learning algorithm for solving partial differential equations,” *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018.

[14] J. Han, A. Jentzen, and W. E, “Solving high-dimensional partial differential equations using deep learning,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 34, pp. 8505–8510, 2018.

[15] O. Fuks and H. A. Tchelepi, “Limitations of physics informed machine learning for nonlinear two-phase transport in porous media,” *Journal of Machine Learning for Modeling and Computing*, vol. 1, no. 1, pp. 19–37, 2020.

[16] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient pathologies in physics-informed neural networks.” *arXiv preprint arXiv:2001.04536*, 2020.

[17] Udayraj, P. Talukdar, A. Das, and R. Alagirusamy, “Heat and mass transfer through thermal protective clothing – a review,” *International Journal of Thermal Sciences*, vol. 106, pp. 32–56, 2016.

[18] D. A. Torvi and J. D. Dale, “Heat transfer in thin fibrous materials under high heat flux,” *Fire Technology*, vol. 35, no. 3, pp. 210–231, 1999.

[19] P. Chitrphiromsri and A. V. Kuznetsov, “Modeling heat and moisture transport in firefighter protective clothing during flash fire exposure,” *Heat and Mass Transfer*, vol. 41, no. 3, pp. 206–215, 2003.

[20] F.-L. Zhu and W.-Y. Zhang, “Modeling heat transfer for heat-resistant fabrics considering pyrolysis effect under an external heat flux,” *Journal of Fire Sciences*, vol. 27, no. 1, pp. 81–96, 2009.

[21] A. Ghazy and D. J. Bergstrom, “Numerical simulation of heat transfer in firefighters’ protective clothing with multiple air gaps during flash fire exposure,” *Numerical Heat Transfer Part A-applications*, vol. 61, no. 8, pp. 569–593, 2012.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, and et al., “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8026–8037.

[23] Y. Han, “The existence and uniqueness of the global solutions for the mixed problem of some nonlinear heat equations,” *Journal of Sichuan Normal University (NATURAL SCIENCE)*, 1995.

[24] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[25] J. W. Siegel and J. Xu, “Approximation rates for neural networks with general activation functions,” *Neural Networks*, vol. 128, pp. 313–321, 2020.

[26] L. C. Evans, *Partial Differential Equations*. American Mathematical Society, 2010.

[27] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.