A Parameterized Thermal Simulation Method Based on Physics-Informed Neural Networks for Fast Power Module Thermal Design

Yayong Yang, *Student Member, IEEE*, Zhiqiang Wang, *Senior Member, IEEE*, Yu Liao, *Student Member, IEEE*, Wubin Kong, *Senior Member, IEEE*, Xiaojie Shi, *Senior Member, IEEE*, Run Hu, *Member, IEEE*, and Yonggang Yao, *Member, IEEE*

***Abstract*—This article proposes a parameterized 3D thermal simulation methodology based on physics-informed neural networks (PINNs) to achieve rapid design space exploration for power module thermal design. Leveraging the capability of PINNs to quickly approximate the solutions to the parameterized partial differential equations describing the thermal behavior of power modules, a thermal field simulation framework for a SiC three-phase half-bridge power module is developed for parameterized simulations. After a single unsupervised training session, the PINNs-based model can quickly predict the thermal field distribution results of the power module for different combinations of input parameters. The comparison results show that the PINNs predict results are approximately consistent with both COMSOL numerical simulations and experimental measurements in different combination cases. Moreover, in the task of large design space exploration for parameter optimization, the simulation process can be hundreds of times faster than traditional numerical simulation methods, significantly reducing the time cost required for thermal simulations.**

***Index Terms*—Thermal simulation, physics-informed neural networks (PINNs), power module, parameter optimization.**

I. Introduction

THERMAL simulation is crucial for power module thermal design [1]. Traditional numerical simulation software, such as COMSOL Multiphysics and ANSYS Icepack, typically perform thermal simulations under fixed parameter configurations. Any changes in equation parameters, boundary conditions, or the shape of the solution domain require adjusting simulation settings and recalculating to obtain new results. In sensitivity analysis or parameter optimization tasks, multiple thermal simulations are required for different parameter combinations. When the exploration space is large, hundreds or even thousands of thermal simulations may be needed [2]. In such industrial application scenarios, traditional numerical simulation methods are undoubtedly cumbersome and time-consuming [3]. To address this issue and accelerate thermal design and optimization, there is an urgent need for more efficient simulation approaches that can rapidly adapt to varying parameters during large design space exploration.

Fast thermal simulation methods, such as 3D thermal network methods, analytical methods, and data-driven approaches, have been extensively developed to address the slow computation speeds of traditional finite-element method (FEM) and finite volume method. When the thermal network model remains largely unchanged while the power loss model varies, 3D thermal network approaches can significantly reduce computation time compared to numerical simulations [4]. However, in scenarios where the thermal model varies across cases, such as during the thermal optimization of heat sinks, multiple FEM simulations are required to extract the thermal resistance and capacitance parameters for the 3D thermal network model, necessitating a substantial amount of numerical simulations or experimental support [5].

Analytical methods are based on the fundamental physical laws of heat conduction and employ mathematical tools to solve PDEs. Fourier transforms and Green function methods are two common analytical approaches in the thermal simulation of power modules [6], [7]. Fourier transforms are well-suited for thermal transfer problems with periodic boundary conditions, while Green functions efficiently address local heat source temperature responses. These methods achieve extremely high computational efficiency in scenarios with regular geometries and well-defined boundary conditions. However, their applicability is limited by geometric complexity and nonlinear conditions, rendering them unsuitable for directly handling complex structures and dynamic boundary problems.

Received 18 October 2024; revised 13 January 2025; accepted 24 February 2025. Date of publication 5 March 2025; date of current version 14 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 52277179 and in part by the Science and Technology Project of China Southern Power Grid Corporation under Grant GDKJXM20222074. Recommended for publication by Associate Editor Y. Zhang. *(Corresponding author: Zhiqiang Wang.)*

Yayong Yang is with the Institute of Artificial Intelligence, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yangyayong@hust.edu.cn).

Zhiqiang Wang, Wubin Kong, Xiaojie Shi, and Run Hu are with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zhiqiangwang@hust.edu.cn; wbkong@hust.edu.cn; xiaojie_shi@hust.edu.cn; hurun@hust.edu.cn).

Yu Liao is with the Electric Power Research Institute of Guangdong Power Grid Company Ltd., Guangzhou 510030, China (e-mail: liaoyu@gddky.csg.cn).

Yonggang Yao is with the School of Materials Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yaoyg@hust.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2025.3547390.

Digital Object Identifier 10.1109/TPEL.2025.3547390

Data-driven methods rely on experimental or simulation data to establish efficient surrogate models through machine learning or other statistical techniques, enabling rapid prediction of nonlinear thermal behaviors in complex scenarios. These approaches significantly reduce computational costs while maintaining high accuracy. Data-driven approaches excel in real-time applications and nonlinear problem-solving but depend heavily on high-quality data. Compared to traditional numerical methods, they exhibit limitations in generalization capabilities and interpretability. A hybrid data-driven and mechanistic modeling approach for power module rapid thermal analysis is proposed in [8]. This method improves design efficiency by accelerating thermal field calculations by 300 times during the design stage. Its accuracy is validated against FEM and experimental results, offering a reliable tool for power module design. This method is developed specifically for thermal modeling in layout optimization. Nonetheless, its adaptability to other optimization tasks, such as thermal optimization, may be constrained.

The limitations of the aforementioned rapid thermal simulation method highlight the necessity of developing a more general and efficient simulation framework capable of adapting to complex geometries and various changing parameter conditions. As an emerging deep learning-based method that incorporates physical constraints into machine learning for approximating solutions to PDEs, PINNs have become a promising solution to tackle these challenges [9], [10]. Similar to traditional numerical simulations, PINNs can generate complete field distribution results across various physical fields, rather than being restricted to computations at only a few discrete points [11]. By taking continuous spatial coordinates as the input for the neural networks, PINNs can generate continuous solutions throughout the entire domain [12]. However, unlike traditional numerical solvers, PINNs leverage neural networks to approximate the solution to PDEs while incorporating the underlying physical laws directly into the loss function. This allows PINNs to bypass the need for predefining grid-based discretization [13]. Moreover, PINNs are highly adaptable, capable of handling both non-parameterized and parameterized thermal simulations. In nonparameterized cases, PINNs can directly simulate the behavior of thermal fields without requiring specific input parameters. For parameterized thermal simulations, PINNs offer a significant advantage by efficiently handling variations in input parameters and providing solutions that reflect these changes [14].

PINNs have been extensively applied to approximating solutions to nonparameterized PDEs in fields like fluid dynamics, heat transfer, solid mechanics, etc. [15]. In fluid dynamics, the literature [16] introduces a PINNs-based approach for approximating the solution to the Navier–Stokes (N-S) equations, enabling the simulation of complex fluid behavior. Building on this, researchers have developed Navier–Stokes Flow nets, which utilize PINNs for approximating the incompressible N-S equations, showing potential in turbulence simulation [17]. In the field of heat transfer, Cai et al. [11] uses PINNs to address steady-state heat conduction problems, demonstrating their effectiveness in complex geometries. Another study employs PINNs to simulate heat conduction at the microscale, providing valuable insights for thermal design in nanotechnology [18]. In solid mechanics, a PINNs-based framework for solid mechanics systems is proposed in [19], successfully predicting structural responses under applied loads. Additionally, PINNs have been used to model the nonlinear behavior of elastomers, offering an innovative approach for designing flexible electronics [20].

Although the use of PINNs for approximating non-parameterized PDEs has become widespread, there is considerable debate regarding whether PINNs offer faster solutions compared to traditional methods. This is due to the varying complexity of the PDEs being solved across different studies, as well as differences in the computational resources employed. Consequently, an increasing number of studies are now focusing on leveraging PINNs capability for parametric simulations in high-dimensional parameter spaces to enhance simulation efficiency. In [21], PINNs are employed to approximate the parameterized equations in different fields. By incorporating parameterized PINNs, the researchers demonstrate that the proposed approach surpasses baseline methods in terms of both accuracy and parameter efficiency when applied to benchmark 1D and 2D parameterized PDEs. In [22], an efficient surrogate modeling framework based on PINNs is developed for 2D parameterized premixed combustion systems. The PINNs surrogate model, with five input parameters, accurately predicts combustion fields, closely matching direct numerical simulations. Simulating 161051 parameter combinations with numerical methods takes 3000 times longer than with the PINNs model. Similar 2D parameterized simulation methods based on PINNs have been explored in [23] and [24].

Despite the pioneering research on PINNs in parameterized simulations, no studies have yet been found applying PINNs to parameterized 3D thermal simulation in the field of power electronics. Therefore, this article focuses on SiC power modules with complex geometrical structures, developing a PINNs-based parameterized simulation framework to evaluate the capability of PINNs in addressing steady-state thermal simulation problems of power electronics. Due to issues, such as poor convergence and excessive computational overhead, transient thermal simulation lies beyond the current scope of this article.

The rest of this article is structured as follows. Section II provides a detailed explanation of the proposed PINNs-based parameterized thermal simulation method employing a specific case study as an example. In Section III, the simulation speed, accuracy, and scalability of the proposed methodology are evaluated by comparing it with COMSOL thermal simulations and experimental results. Finally, Section IV concludes this article.

II. Proposed Simulation Methodology

A. Methodology Overview

The core objective of steady-state thermal simulation for SiC power modules involves solving steady-state PDEs, such as the heat conduction equation, the convection equation, and the N-S equations. A PINN approximator estimates the solution to a given PDE along with a set of boundary conditions through a neural network. The model is trained by defining a loss function that reflects how accurately the network adheres to the PDE and

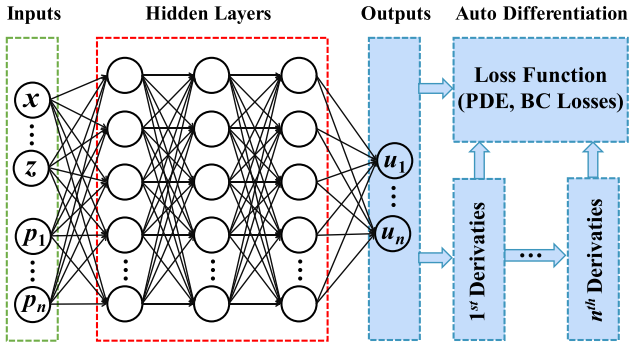

Fig. 1. Schematic of the structure of a PINN approximator.

constraints. By minimizing this loss function, the network effectively approximates the solution to the PDE. Fig. 1 illustrates the structure of a neural network approximator. The inputs of the network are the spatial coordinates of the point cloud and the values from the parameter space. These inputs are mapped to the quantities of interest through a neural network with nonlinear activation functions. To train the network, a loss function is employed, which includes the derivatives of the output with respect to the input (computed via automatic differentiation) as well as boundary condition information.

A steady-state PDE typically has the following general form.

$$\mathcal{F}\left[u\right]\left(\boldsymbol{x}\right)-f\left(\boldsymbol{x}\right)=0 \tag{1}$$

$$\mathcal{B}\left[u\right]\left(\boldsymbol{x}\right)-g\left(\boldsymbol{x}\right)=0 \tag{2}$$

where $\mathcal{F}$ represents general differential operators, and $\boldsymbol{x}$ refers to the set of independent variables, encompassing both spatial coordinates and the parameterized input flexible parameters. The function $u(\boldsymbol{x})$ serves as the solution to the PDE. The operator $\mathcal{B}$ is associated with boundary conditions, which may be Dirichlet, Neumann, or Robin types. $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are employed to characterize specific physical conditions within the problem domain and on the boundary, respectively.

A neural network $\hat{u}(\boldsymbol{x};\theta)$ is constructed to approximate the solution $u(\boldsymbol{x})$, where θ represents the trainable parameters of the network. To train this neural network, a loss function is constructed that penalizes the divergence of the approximate solution $\hat{u}(\boldsymbol{x};\theta)$ from the PDE. For this purpose, the following residuals are defined:

$$R_{\mathrm{PDE}}\left(\boldsymbol{x}\right)=\mathcal{F}\left[\hat{u}\right]\left(\boldsymbol{x}\right)-f\left(\boldsymbol{x}\right) \tag{3}$$

$$R_{\mathrm{BC}}\left(\boldsymbol{x}\right)=\mathcal{B}\left[\hat{u}\right]\left(\boldsymbol{x}\right)-g\left(\boldsymbol{x}\right) \tag{4}$$

where R_{PDE} and R_{BC} represent the residuals of the PDE and the boundary conditions, respectively. Consequently, the loss function for the neural network is expressed as follows:

$$\mathcal{L}_{\mathrm{res}}=\frac{\lambda_f}{N_f}\sum_{i=1}^{N_f}\left|R_{\mathrm{PDE}}\left(x_i\right)\right|^2+\frac{\lambda_b}{N_b}\sum_{i=1}^{N_b}\left|R_{\mathrm{BC}}\left(x_i\right)\right|^2 \tag{5}$$

where λ_f and λ_b are weight factors. N_f and N_b represent the number of sample points for interior and boundary sampling in the parameter space, respectively. $\mathcal{L}_{\mathrm{res}}$ measures how well the neural network satisfies the PDE and boundary constraints. The smaller $\mathcal{L}_{\mathrm{res}}$ is, the closer $\hat{u}(\boldsymbol{x};\theta)$ is to the true solution $u(\boldsymbol{x})$.

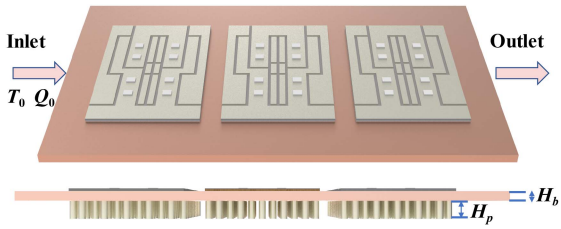

Fig. 2. SiC MOSFET power semiconductor module with Pin-Fin structure.

By utilizing PyTorch's automatic differentiation capabilities, the derivatives of different orders from the neural network output with respect to the input can be computed. These derivatives are then substituted into (3) and (4) to determine the residuals, which are subsequently used to compute $\mathcal{L}_{\mathrm{res}}$. Finally, the approximate solution $\hat{u}(\boldsymbol{x};\theta)$ is trained by iteratively optimizing the network parameters θ through the stochastic gradient descent methods like the Adam optimizer. Following the PINN solution method for a single PDE as described above, multiple neural networks can fit the all PDEs involved in the thermal field simulation.

PINNs incorporate physical laws as constraints, guiding the training of the neural network by leveraging the residuals of PDEs and boundary conditions. PINNs approximate PDEs via neural network optimization, bypassing mesh grids and excelling in high-dimensional problem. In fact, PINNs are a novel PDE approximator that integrates the powerful generalization capabilities of machine learning, which sets them apart from traditional methods. Unlike traditional machine learning methods, which require labeled data to compute loss values, PINNs do not rely on labeled datasets.

B. *Introduction of the Studied Case*

A direct water-cooled three-phase half-bridge SiC power module (see Fig. 2) serves as the studied case. Each switch position consists of four 1.2 kV SiC MOSFET chips (Rohm S4108), without any anti-parallel diode chips. The DBC AlN substrates are fixed onto a copper baseplate integrated with a Pin-Fin liquid cooling structure through soldering. The flow channel structure is a series flow channel, and the cooling medium is deionized water. More detailed information about the SiC power module is provided in Table I.

For incompressible deionized water, the conservation equations governing the transfer of mass, momentum, and energy are as follows:

$$\begin{cases}\nabla\cdot\boldsymbol{v}=0\\ \rho\left(\boldsymbol{v}\cdot\nabla\right)\boldsymbol{v}=-\nabla p+\mu\nabla^2\boldsymbol{v}\\ \rho C_p\left(\boldsymbol{v}\cdot\nabla T_f\right)=k_f\,\nabla^2 T_f\end{cases} \tag{6}$$

ρ, μ, k_f, and C_p are the density, dynamic viscosity, thermal conductivity and specific heat capacity of the fluid medium, respectively; $\boldsymbol{v}$ is the velocity field vector which can be represented by its components $(u,\ v, w)$, corresponding to the

TABLE I
PACKAGING MATERIALS AND DIMENSIONS

Component	Material	Size (mm)
Chip	SiC	3.06 × 2.45 × 0.38
chip solder	SAC305	3.06 × 2.45 × 0.15
DBC copper	Cu	60 × 40 × 0.3
DBC ceramic	AlN	60 × 40 × 0.63
Baseplate solder	SAC305	60 × 40 × 0.25
Baseplate	Cu	——
Cold plate	Al 6061	152 × 90 × 18
Cooling medium	Water	——

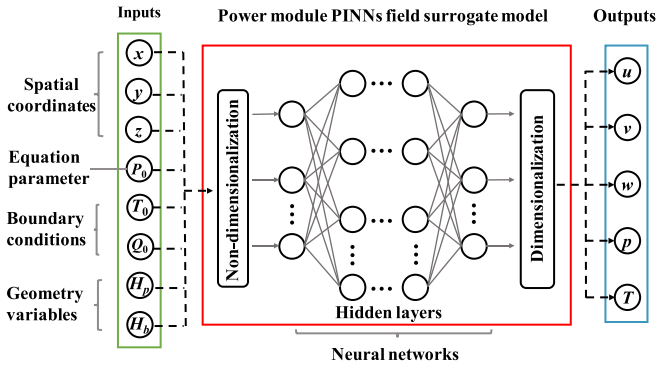

Fig. 3. Schematic of the power module PINNs field surrogate model.

velocity components of the fluid in the x, y, and z directions, respectively. p is the pressure. T_f is the water temperature.

For the solid materials within the power module, the heat transfer equation is expressed as follows:

$$k_s \nabla^2 T_s + Q_s = 0. \tag{7}$$

The thermal conductivity of the solid material is represented by k_s, the temperature of the solid is indicated by T_s, and the heat source loss is denoted as Q_s.

At the interfaces between different solid materials, as well as at the interfaces between fluid (water) and solid (baseplate), the same boundary conditions of temperature and heat flux continuity need to be satisfied. Taking the solid-fluid interface as an example, the following equations need to be satisfied:

$$\begin{cases} T_s = T_f \\ k_s \frac{\partial T_s}{\partial \boldsymbol{n}} = k_f \frac{\partial T_f}{\partial \boldsymbol{n}} \end{cases}. \tag{8}$$

The vector $\boldsymbol{n}$ represents the unit normal directed outward at the solid-liquid interface.

In addition to the continuity boundary condition, the fluid velocity at the inlet is specified as Q_0, indicating a given flow rate. The inlet temperature is prescribed as T_0. At the outlet, the pressure is set to $p = 0$, and no-slip boundary conditions are applied to the channel walls.

C. Parameterized Simulation Procedure

As shown in Fig. 3, in the 3D parameterized multiphysics coupling study case, the PINNs model introduces two boundary condition parameters ($T_0 \in [25,\ 35]^{\circ}\mathrm{C}$, $Q_0 \in [8,\ 12]\ \mathrm{L/min}$), one equation parameter (chip total power loss $P_0 \in [1200,\ 1800]\ \mathrm{W}$), and two geometric parameters (baseplate thickness $H_b \in [2,\ 4]\ \mathrm{mm}$ and Pin-Fin height $H_p \in [4,\ 8]\ \mathrm{mm}$) as input variables, to comprehensively demonstrate the ability of PINNs to handle various types of variable input parameters. The model inputs, denoted as $\boldsymbol{x}$, encompass the spatial coordinates along with five adjustable parameters. The model output is the set of all physical quantities to be solved in the PDEs involved in the thermal system of the SiC power module. Within the model, a nondimensionalization preprocessing step is initially applied to normalize all thermal system parameters, ensuring that the velocity, temperature, and pressure scales are approximately within the $[0,\ 1]$ ranges. This step is essential for achieving rapid convergence and maintaining balanced loss during the training of the neural network model. The process of nondimensionalization begins with selecting appropriate representative scales. For thermal simulations of power modules, it is crucial to establish scales for the four primary physical quantities: length L, mass M, time t, and temperature T. The scales for other physical quantities are subsequently determined based on their relationships with these fundamental quantities. The scales for the four basic quantities are defined as follows:

$$L_{\text{scale}} = 0.02\ \mathrm{m}, \qquad M_{\text{scale}} = 1 \times 10^{-5}\ \mathrm{kg} \tag{9}$$

$$t_{\text{scale}} = 0.01\ \mathrm{s}, \qquad T_{\text{scale}} = 500\ \mathrm{K}. \tag{10}$$

The second step involves nondimensionalizing the original variables by dividing each by its respective scale. In the third step, these nondimensionalized variables are substituted into the original physical equations to obtain their nondimensional forms. Finally, the derived nondimensional variables and parameters are utilized in the training of neural networks.

Dimensionalization is the process of converting nondimensional results back to their original physical units. This typically happens after model training and prediction to map the nondimensional outputs back to the actual physical quantities. To recover the original variables from non-dimensional ones, multiply by the corresponding scales.

Subsequently, neural networks, recognized as universal approximators, are employed to construct the high-dimensional non-linear mapping. The inputs $\boldsymbol{x}$ and outputs $\boldsymbol{Y}$ are linked through the forward propagation across multiple hidden layers $\boldsymbol{H}_i$ expressed as

$$\emptyset_E(\boldsymbol{x}) = [\sin(2\pi \boldsymbol{f} \times \boldsymbol{x});\ \cos(2\pi \boldsymbol{f} \times \boldsymbol{x})]^T \tag{11}$$

$$\boldsymbol{H}_1 = \sigma(\boldsymbol{W}_1 \emptyset_E(\boldsymbol{x}) + \boldsymbol{b}_1) \tag{12}$$

$$\boldsymbol{H}_i = \sigma(\boldsymbol{W}_i \boldsymbol{H}_{i-1} + \boldsymbol{b}_i),\ \text{for } i = 2, 3, \ldots,\ n_i \tag{13}$$

$$\boldsymbol{Y}(\boldsymbol{x}) = \boldsymbol{W}_{n_i+1} \boldsymbol{H}_{n_i} + \boldsymbol{b}_{n_i+1} \tag{14}$$

where $\boldsymbol{f}$ represents the trainable frequency matrix. $\boldsymbol{H}_i$ is the hidden layer of the ith layer, with weights and biases denoted as $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$. The activation function $\sigma(\cdot)$, specifically the Swish

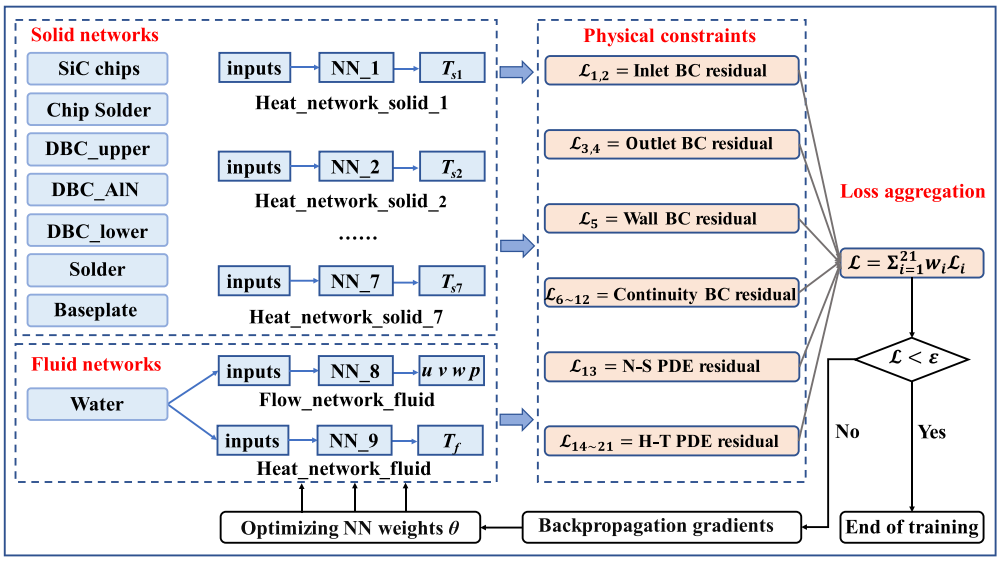

Fig. 4. Detailed implementation flowchart of the PINNs model (NN represents neural network, BC refers to boundary condition, and H-T stands for heat transfer.

function, is employed to construct a continuously differentiable neural representation. The total number of hidden layers is denoted as n_i. $\emptyset_E$ represents the input encoding layer of the Fourier network, which maps inputs to a higher dimensional feature space using high-frequency functions. This Fourier network architecture enhances model accuracy and mitigates the spectral bias typically present in standard fully connected neural networks [25].

As illustrated in Fig. 4, the power module PINNs model comprises multiple Fourier networks. For the solid part, since each layer of the power module has different material properties, the PDE corresponding to that domain varies. Each layer requires a Fourier network to represent and fit the heat transfer equation corresponding to that physical domain, necessitating seven networks. For the fluid part, two networks are required to fit the fluid flow PDE and the convective heat transfer PDE, respectively. In this studied case, a total of nine Fourier neural networks are employed, with each network responsible for satisfying the physical constraints corresponding to different regions of the SiC power module. Due to the use of a soft coupling constraint method, the boundary condition terms in the loss functions of each network do not directly include the outputs of other networks. By introducing continuity boundary condition residual penalty terms into the global loss, this method indirectly enhances the consistency between networks without requiring strict satisfaction of interface continuity boundary conditions at every iteration. The inputs for each network are the same, and the output is the physical quantity to be solved for the corresponding PDE. Furthermore, the entire Fourier network approximators are structured with 6 hidden layers, each containing 512 neurons, which typically produces optimal and stable outcomes.

Next, it is necessary to minimize the loss functions based on the neural network outputs and their derivatives. The loss function consists of two parts: the residual of the partial differential equation and the mean squared error of the boundary conditions. These losses are calculated based on user-defined

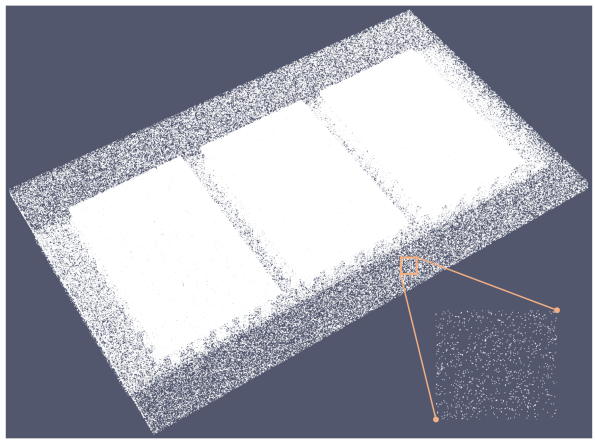

Fig. 5. Point cloud graph about the randomized sampling points.

sampling points randomly generated throughout the solution domain and respective boundaries. The sampling method adopted is uniform random sampling, with each material layer and interface sampled separately. This approach is motivated by several key considerations, including avoiding overfitting, ensuring compatibility with stochastic gradient descent, ease of implementation, enhancing generalization, and adapting effectively to high-dimensional problems. The point cloud graph about the sampling points is shown in Fig. 5.

All the losses are aggregated with appropriate weights. In this weighting scheme, a higher weight is assigned to the PDE residual to ensure accurate fitting of complex equations. At the same time, boundary conditions are given a lower weight because they are typically easier to satisfy. The interface continuity weight is initially set relatively low to avoid overshadowing other terms. As the model learns the solutions in each region, this weight is gradually increased to refine interface continuity and boost overall performance. Specifically, the weights for $\mathcal{L}_{1\sim5}$ are set to

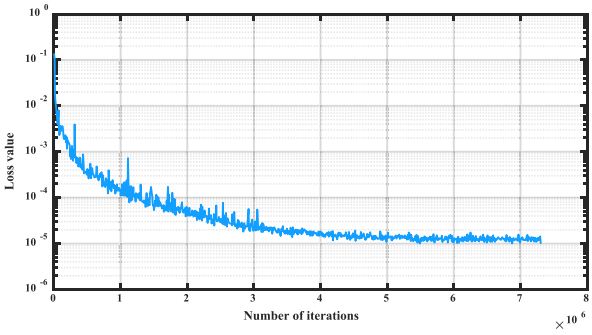

Fig. 6. Aggregated loss variation curve for the studied case.

$1, \mathcal{L}_{13}$ is set to 20, $\mathcal{L}_{14\sim21}$ are set to 10, and $\mathcal{L}_{6\sim12}$ are dynamically adjusted according to the training iteration number n

$$w(n) = (w_{\max} - w_{\min}) \times \frac{1}{1 + e^{-k(n-n_0)}} + w_{\min} \tag{15}$$

where $w_{\max} = 10, w_{\min} = 1, k = 4 \times 10^{(-6)}, n_0 = 4 \times 10^6$.

If the aggregated loss $\mathcal{L}$ exceeds the defined threshold, backpropagation is performed to optimize the trainable weights θ, iterating until $\mathcal{L}$ decreases to the predefined error threshold (set to 10^{-5} in this article). In this studied case, the aggregated loss variation during the training process is presented in Fig. 6. After 7 305 000 iterations, the loss value dropped below 10^{-5}, marking the end of the PINNs model training. It is worth noting that, to enable strongly coupled networks to converge quickly and synchronously, the unified aggregated loss function, the adaptive optimization algorithm Adam, and the exponential decay learning rate mechanism are employed. These networks are jointly trained to ensure synchronized parameter updates.

Once the training is complete, it can run inference for any combination of the five input parameters, yielding corresponding thermal simulation results without retraining the neural network from scratch.

III. Experimental Verification

A. Speed Verification

To evaluate the proposed PINNs-based thermal simulation method against traditional numerical methods, COMSOL is used as a benchmark. Both methods utilize identical PDEs and boundary conditions for a fair comparison. While the inference time of a trained PINNs model is minimal (about 1∼2 s), the training cost must also be considered. The PINNs model is trained utilizing 2 NVIDIA A800 GPUs, whereas each corresponding COMSOL thermal model simulation employs 2 Intel Xeon Platinum 8380 CPUs, each featuring 40 cores, and is equipped with 1TB of RAM. To ensure the fairness of the comparison as much as possible, COMSOL computations are equipped with top-tier computational resources. Under the currently available resources, its simulation efficiency is difficult to improve further. Table II gives PINNs training and COMSOL computation time costs for different simulation scenarios, along with the maximum junction temperature error between the two.

TABLE II
PINNs Training and COMSOL Computation Details

Case description	Time costs (hours)	Errors
COMSOL single run	1.7	1.2%
Non-parameterized PINNs single run	25.6	
COMSOL 100 cases run	170	2.6%
Parameterized PINNs 100 cases run	153.8	
COMSOL 10 000 cases run	17 000	3.0%
Parameterized PINNs 10 000 cases run	159.3	

For the nonparameterized case that can only simulate a specific situation with inputs limited to spatial coordinates, the time cost of training a PINNs model (25.6 h) is approximately 15 times that of executing a single COMSOL thermal simulation with refined mesh division (1.7 h). For nonparameterized simulations, the simulation efficiency of the PINNs model can only potentially surpass traditional numerical methods when a larger number of GPUs are utilized for PINNs model training. In scenarios where GPU resources are limited, traditional numerical methods are more efficient for non-parameterized simulation than PINNs for the power module 3D thermal simulation.

In the studied case, when employing the proposed PINNs parameterized method for 100 simulations with different input parameter combinations (P_0, T_0, Q_0, H_p, H_b), the training and inference time (153.8 h) of the parameterized PINNs model is comparable to the time required for COMSOL to perform 100 simulations (170 h). However, when exploring a larger parameter space, such as conducting 10 000 simulations with different combinations of the five input parameters, the computational cost for COMSOL (17 000 h) extends to several months, which is 106 times that of the PINNs-based parameterized simulation method (159.3 h). The parameterized PINNs approach has a significant advantage in simulation efficiency for the large design space exploration. The larger the design space required for parameter optimization, the more pronounced the advantage in simulation efficiency.

B. Accuracy Verification

To verify the accuracy of thermal simulations, a unified liquid cooling performance testing platform for SiC power modules is established, as illustrated in Fig. 7. The pumping power of the entire liquid cooling system is primarily provided by an external water chiller, which ensures that the temperature of the cooling fluid at the inlet of the SiC power module cold plate is maintained at the set value. The cooling fluid flow rate is precisely regulated by controlling the flow valve. A precision pressure sensor (range: 0–600 kPa and accuracy: 0.2%) is placed at both the inlet and outlet sides near the SiC power module cold plate to measure the actual pressure at these points. In addition, to ensure accurate measurement of the SiC power module chip temperature, the upper surface of its internal DBC substrate is evenly coated with black insulating paint, and an infrared thermal imaging camera (Guide PS610) is used for temperature measurement. The experiment utilizes a constant

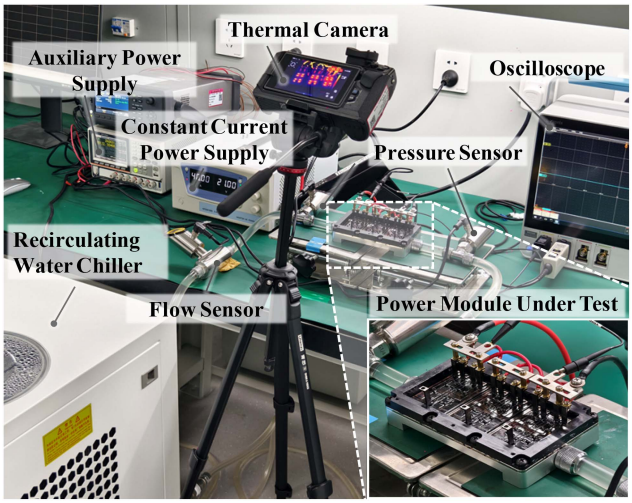

Fig. 7. Power module thermal performance testing platform.

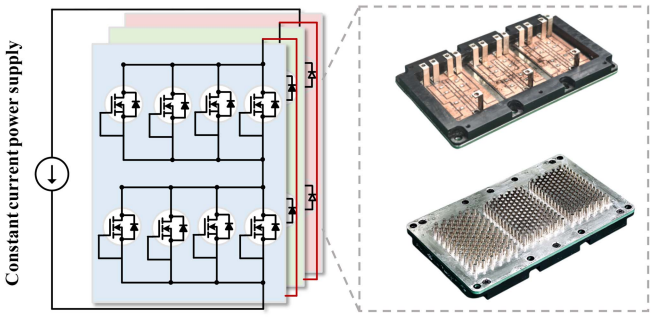

Fig. 8. Power module testing platform circuit schematic and the fabricated power module. (a) Circuit schematic. (b) Fabricated SiC power module.

current and constant voltage power supply (0–30 V, 0–100 A) to power the three-phase SiC power module. The schematic of the experimental circuit is depicted in Fig. 8. Three half-bridge DBC substrates are connected in series to ensure current consistency. The gate and source of the SiC MOSFET chips are short-circuited, allowing reverse current flow through the body diodes of the chips. In the experiment, the power losses of the SiC chips are controlled by adjusting the output of the constant current power supply.

Based on this experimental platform, experimental and simulation results corresponding to the lower, middle, and upper limits of the five input parameters are selected for comparison. The thermal field results of the three cases are shown in Fig. 9. The pressure field distribution results for the middle case are illustrated in Fig. 10. The thermal field and pressure field simulation results of parameterized PINNs are composed of point clouds, with visualization achieved through ParaView software. It is worth noting that PINNs have been provided with an adequate number of sampling points to ensure the accuracy. The comparison of errors in peak temperature and inlet/outlet pressure drop between the two simulation methods and experiments is given in Table III. The pressure drop in the experiments is measured using a high-precision pressure

TABLE III
COMPARISON OF PEAK TEMPERATURE AND PRESSURE DROP ERRORS BETWEEN TWO SIMULATION METHODS AND EXPERIMENTS

Case description	Method	Peak temperature errors	Pressure drop errors
Case 1	COMSOL	1.0%	3.9%
	PINNs	0.6%	8.0%
Case 2	COMSOL	0.8%	4.4%
	PINNs	2.3%	9.1%
Case 3	COMSOL	1.1%	5.5%
	PINNs	2.9%	7.9%

sensor. The measured pressure drops for the three cases are 8.15, 9.98, and 12.27 kPa. It can be observed that the accuracy of the parameterized PINNs method is slightly lower than that of COMSOL, both in terms of temperature results and pressure drop results. The maximum temperature error between the two simulation methods is under 3%, while the maximum pressure drop discrepancy is less than 5%. The maximum temperature error between PINNs and the experiment is 2.9%, while the maximum pressure drop error is 9.1%. The relatively large error in pressure drop calculation is attributed to the high complexity involved in solving the N-S equations. The comparison results demonstrate the high temperature simulation accuracy of the proposed parameterized simulation method.

C. Scalability Verification

For nonparameterized PINNs simulation models, similar to traditional numerical simulation methods, training can only simulate a specific set of parameters at a time. Any change in parameters necessitates retraining to improve simulation accuracy. In contrast, parameterized PINNs simulation models can efficiently simulate under multiple parameter combinations by introducing variable parameters into the neural network inputs. However, parameterized PINNs are only effective when the introduced adjustable parameters change. If other parameters vary, retraining is necessary to account for these changes. This is an unavoidable issue for all parameterized simulation methods. The more input parameters in the parameterized PINNs model, the stronger its generalization ability. Due to the numerous variable parameters in actual power module systems, it is impossible to construct a universal parameterized model that can respond quickly and accurately to all parameter variations. At this point, the scalability of parameterized simulation models becomes crucial.

To verify the performance of the proposed parameterized simulation model in terms of parametric scalability, nine geometric parameters, as shown in Fig. 11, are added as new input parameters based on the five input parameters from the original study case. As a result, the number of input parameters for the parameterized PINNs model increased to 14. The radius R of the cylindrical fins within each region is defined within the range of [0.5, 1.5] mm, the horizontal spacing D_h ranges from [3, 6] mm, and the vertical spacing D_v is specified within [3, 5] mm. Except for the differences in input parameters, the other

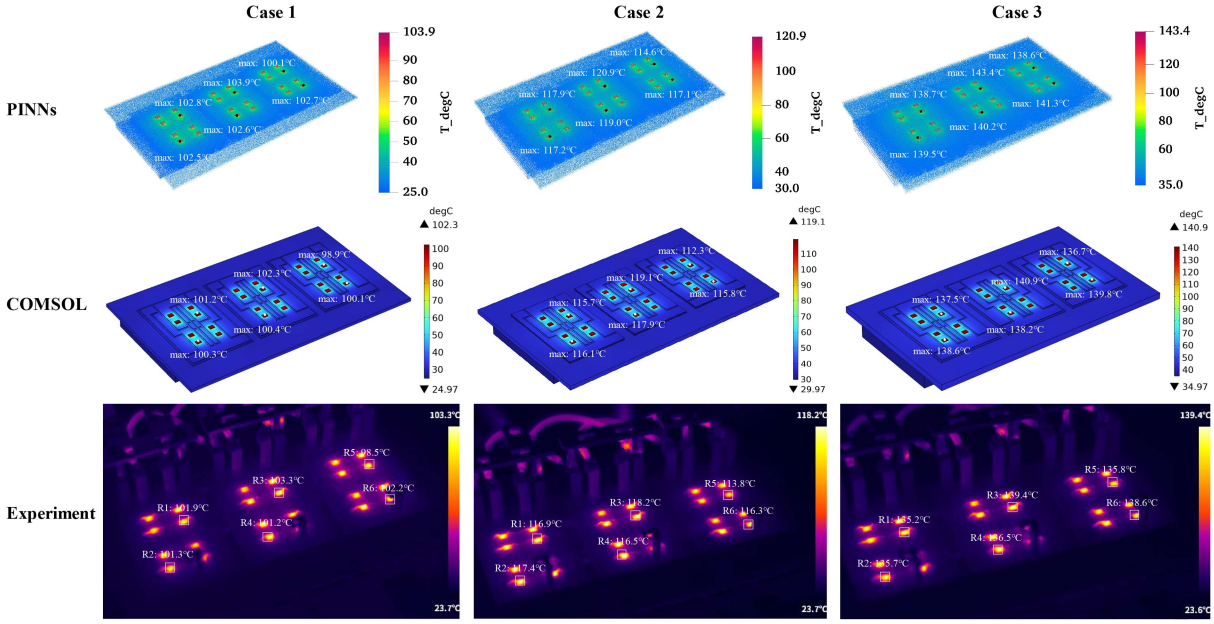

Fig. 9. Comparison of the temperature distribution results from two simulation methods and experimental results for three different cases (case 1: $T_0 = 25\,^{\circ}\mathrm{C}$, $Q_0 = 8$ L/ min, $P_0 = 1200$ W, $H_b = 2$ mm, $H_p = 4$ mm; case 2 : $T_0 = 30\,^{\circ}\mathrm{C}$, $Q_0 = 10$ L/ min, $P_0 = 1500$ W, $H_b = 3$ mm, $H_p = 6$ mm; and case 3 : $T_0 = 35\,^{\circ}\mathrm{C}$, $Q_0 = 12$ L/ min, $P_0 = 1800$ W, $H_b = 4$ mm, $H_p = 8$ mm.).

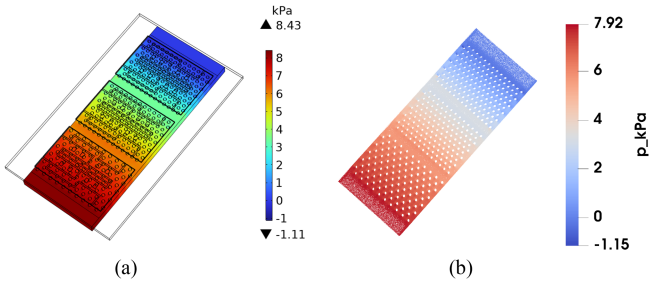

Fig. 10. Comparison of the fluid pressure distribution results from two simulation methods for case 2. (a) COMSOL simulation. (b) PINNs simulation.

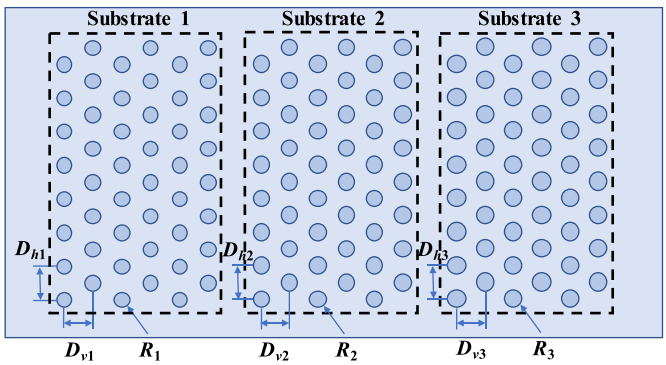

Fig. 11. Schematic of the parameters to be parameterized for the Pin-Fin liquid cooling structure.

settings remain consistent with the previous study case. The variation curve of the aggregated loss values during the training process of the new parameterized PINNs model is depicted in Fig. 12. The small loss function value indicates that the neural networks have successfully satisfied the physical constraints,

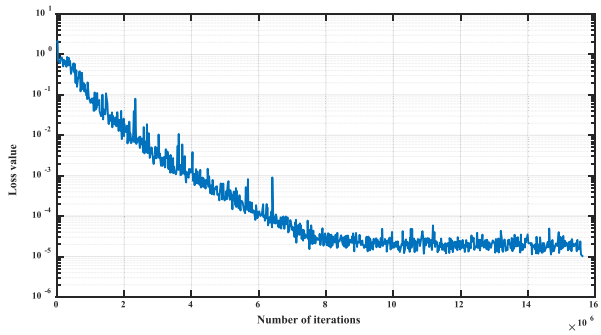

Fig. 12. Aggregated loss variation curve for the new study case.

suggesting that the networks have approximated the solution to the parameterized PDEs effectively. The thermal field simulation result for one of the many parameter combinations inferred using the newly trained PINNs model is compared with experimental results, as shown in Fig. 13. The corresponding Pin-Fin structure and the pressure field distribution results obtained from the PINNs simulation are illustrated in Fig. 14. The maximum error in junction temperature between the PINNs simulation and the experiment is 2.3 °C. The experimentally measured pressure drop between the inlet and outlet is 10.66 kPa, and the error between the PINNs simulation and the experiment is 0.59 kPa. This further demonstrates the high accuracy of the new parameterized PINNs simulation model. The high accuracy of the newly trained parameterized model demonstrates the strong parameter scalability of the PINNs model when extending parameters. The training time for the new model is 356.8 h. It can be observed that the more parameters used for parameterization, the longer the training time required under the same GPU resources. In practical applications, the pre-selection of input parameters

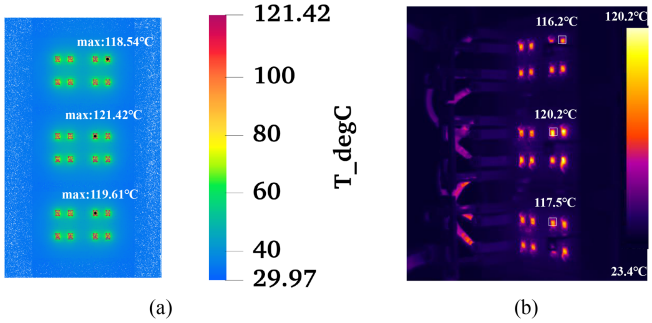

Fig. 13. Steady-state temperature distribution of the power module. (a) PINNs simulation result. (b) Experiment result (the highest temperature of each phase is marked).

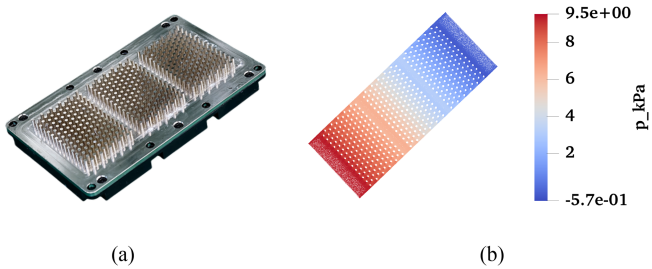

Fig. 14. The Pin-Fin structure and pressure field distribution. (a) Pin-Fin structure. (b) Pressure field distribution ($T_0 = 30°C$, $Q_0 = 10\,\mathrm{L/min}$, $P_0 = 1500$ W, $H_b = 3$ mm, $H_p = 6$ mm, $R_1 = R_2 = R_3 = 1.1$ mm, $D_{h1} = D_{h2} = D_{h3} = 4.2$ mm, $D_{v1} = D_{v2} = D_{v3} = 3.6$ mm.).

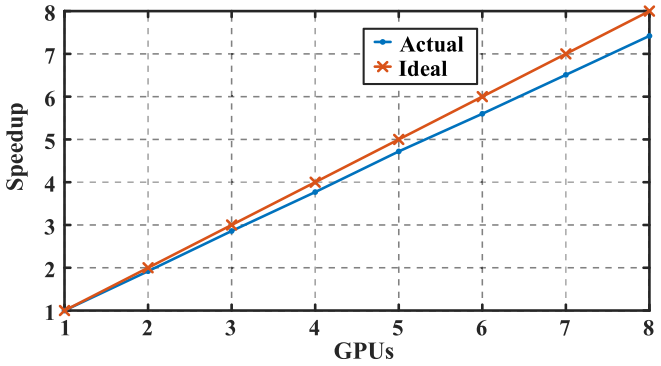

Fig. 15. Ideal and actual speedup ratios with different numbers of GPUs for the new study case.

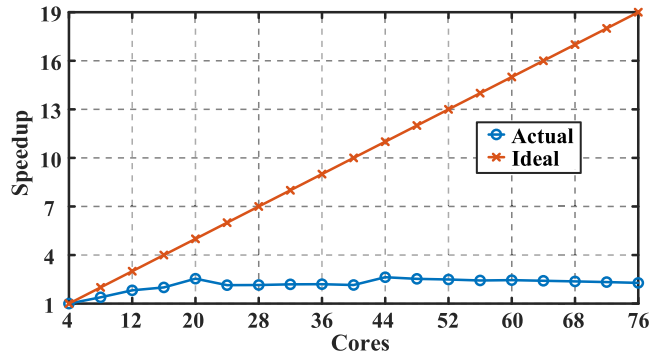

Fig. 16. Ideal and actual speedup ratios with different numbers of CPU cores for COMSOL thermal simulation.

based on requirements is crucial for maximizing efficiency under limited GPU resources.

When sufficient GPU resources are available, an effective way to shorten the convergence time of the PINNs model is to parallelize the training process across multiple GPUs. The most common multi-GPU parallelization strategy is data parallelism, where the global training batch is split into multiple sub-batches and assigned to each GPU. Each GPU processes forward and backward passes on its assigned sub-batch, with gradients being aggregated across all GPUs through the AllReduce algorithm. To evaluate the acceleration performance of the PINNs model in a multi-GPU environment, the training speedup ratio of the new parameterized PINNs model is tested under different numbers of GPUs using the aforementioned data parallel method. Due to the limited availability of GPU resources, this article tests with up to 8 GPUs, and the corresponding test results are shown in Fig. 15. The ideal speedup is linear (i.e., using N GPUs achieves N times speedup), but in practical applications, due to factors, such as communication overhead, hardware architecture, and bandwidth limitations, perfect linear speedup is unattainable. However, since the large computational cost of PINNs models is dominant, these factors have little impact on the overall speed. Furthermore, since nearly 100% of the PINNs model can be parallelized, multi-GPU parallel computing achieves excellent speedup. These two factors typically result in the speedup of PINNs model training with multiple GPUs approaching linear acceleration. Based on the trend of the test results, the training speed of the PINNs model can further improve as the number of GPUs increases. The test results indicate that the proposed parameterized thermal simulation method exhibits excellent scalability with respect to GPU computational resources.

Additionally, this article investigates the impact of different CPU core counts on the computation speed of the COMSOL thermal simulation model. The reason for choosing the number of cores as the research focus is that COMSOL implements parallel computing through multicore parallelization. Testing is conducted with four CPU cores as the baseline (single COMSOL thermal simulation takes 3.6 h), ensuring sufficient memory space. The corresponding results are shown in Fig. 16. The results indicate that increasing core count initially accelerates thermal simulations, but the speedup plateaus as more cores are added, with actual speedup falling well below the ideal. Similar tests on other power module thermal simulation models reveal a similar trend in speedup, which remains well below the ideal level. This is due to the limited proportion of the COMSOL model that can be parallelized, and as core count increases, the overhead of communication and synchronization grows, diminishing the speedup effect or even resulting in worse performance than with fewer cores. In fact, for each COMSOL thermal simulation model, there is an optimal core count, which depends on the CPU specification used and the size of the thermal model. More cores are not necessarily better, and fewer cores are not necessarily worse. The test results reveal the poor

scalability of COMSOL thermal models, meaning that increasing the number of CPU cores does not significantly improve parallel computational efficiency.

IV. Conclusion

This article proposes a parameterized 3D thermal simulation methodology based on PINNs to achieve rapid design space exploration. Compared to numerical methods, which require recalculation when parameters change, the trained machine learning model can predict complete physical field information for any input parameter combination within the design space instantly without needing retraining. The simulation process can be up to hundreds of times faster than traditional numerical methods. The larger the design parameter space, the more apparent the efficiency advantage becomes. Furthermore, the PINNs method exhibits strong scalability, and if needed, it can adapt to different scenarios by adding or adjusting input parameters and retraining. With sufficient GPU resources, PINNs can achieve large-scale expansion due to their powerful nonlinear fitting and parallel training capabilities. This method is not only applicable to power modules but is also suitable for the parameterized thermal simulation of other power electronic systems. The proposed PINNs-based method can serve as an efficient and robust virtual prototyping modeling method for the thermal design and optimization of power electronic systems.

Future work involves optimizing neural network architectures in PINNs, improving sampling strategies, utilizing transfer learning, and achieving transient thermal simulation to further enhance the performance and application scope of PINNs in thermal simulation for power electronics.

References

[1] C. Zhan et al., "Intelligent condition monitoring of multiple thermal degradation of IGBT modules based on case temperature matrix," *IEEE Trans. Power Electron.*, vol. 39, no. 10, pp. 12490–12501, Oct. 2024.

[2] Y. Yang, Z. Wang, Y. Ge, G. Xin, and X. Shi, "An automated field-circuit coupling simulation method based on PSpice-MATLAB-COMSOL for SiC power module design," *IEEE Trans. Power Electron.*, vol. 38, no. 10, pp. 12634–12647, Oct. 2023.

[3] T. Wu, Z. Wang, B. Ozpineci, M. Chinthavali, and S. Campbell, "Automated heatsink optimization for air-cooled power semiconductor modules," *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5027–5031, Jun. 2019.

[4] C. H. van der Broeck, L. A. Ruppert, A. Hinz, M. Conrad, and R. W. De Doncker, "Spatial electro-thermal modeling and simulation of power electronic modules," *IEEE Trans. Ind. Appl.*, vol. 54, no. 1, pp. 404–415, Jan./Feb. 2018.

[5] I. AlRazi, Q. Le, T. M. Evans, H. A. Mantooth, and Y. Peng, "PowerSynth 2: Physical design automation for high-density 3-D multichip power modules," *IEEE Trans. Power Electron.*, vol. 38, no. 4, pp. 4698–4713, Apr. 2023.

[6] L. Xie, X. Yuan, and W. Wang, "Thermal modeling of fan-cooled plate–fin heatsink considering air temperature rise for virtual prototyping of power electronics," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 11, pp. 1829–1839, Nov. 2020.

[7] K. R. Choudhury and D. J. Rogers, "Steady-state thermal modeling of a power module: An *N*-layer Fourier approach," *IEEE Trans. Power Electron.*, vol. 34, no. 2, pp. 1500–1508, Feb. 2019.

[8] J. Zhang et al., "Hybrid data-driven and mechanistic modeling approach for power module rapid thermal analysis," *IEEE Trans. Power Electron.*, vol. 39, no. 11, pp. 14617–14629, Nov. 2024.

[9] Z. Zhang, A. Mehrabi, W. D. Van Driel, and R. H. Poelma, "The potential of machine learning for thermal modelling of SiC power modules - A review," in *Proc. IEEE 10th Electron. Syst.-Integr. Technol. Conf.*, 2024, pp. 1–8.

[10] O. Hennigh, S. Narasimhan, M. A. Nabian, K. Tangsali, Z. Fang, and S. Choudhry, "NVIDIA SimNet: An AI-accelerated multi-physics simulation framework," in *Proc. Int. Conf. Comp. Sci.*, 2021, pp. 447–461.

[11] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks for heat transfer problems," *J. Heat Transfer*, vol. 26, no. 7, pp. 1135–1150, Jun. 2021.

[12] E. Kharazmi, Z. Zhang, and G. Karniadakis, "Variational physics-informed neural networks for solving partial differential equations," *J. Comput. Phys.*, vol. 31, no. 4, pp. 1528–1552, Nov. 2019.

[13] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, no. 5, pp. 686–707, Jan. 2019.

[14] L. Sun, H. Gao, S. Pan, and J. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Comput. Methods Appl. Mech. Eng.*, vol. 361, no. 3, pp. 2524–2549, Apr. 2020.

[15] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An expert's guide to training physics-informed neural networks," *Mach. Learn. Sci. Technol.*, vol. 52, no. 10, pp. 725–761, Aug. 2023.

[16] Z. Mao, A. Jagtap, and G. Karniadakis, "Physics-informed neural networks for high-speed flows," *Comput. Methods Appl. Mech. Eng.*, vol. 360, no. 2, pp. 456–482, Mar. 2020.

[17] X. Jin, S. Cai, H. Li, and G. Karniadakis, "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," *J. Comput. Phys.*, vol. 426, no. 6, pp. 231–257, Feb. 2021.

[18] R. Li, J. Wang, E. Lee, and T. Luo, "Physics-informed deep learning for solving phonon Boltzmann transport equation with large temperature nonequilibrium," *NPJ Comput. Mater.*, vol. 352, no. 8, pp. 487–497, Feb. 2022.

[19] R. Li, J. Wang, E. Lee, and T. Luo, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 379, no. 5, pp. 487–497, Jun. 2021.

[20] L. Lu, X. Meng, Z. Mao, and G. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *Comput. Methods Appl. Mech. Eng.*, vol. 63, no. 7, pp. 1478–1499, Nov. 2021.

[21] W. Cho, M. Jo, H. Lim, D. Lee, S. Hong, and N. Park, "Parameterized physics-informed neural networks for parameterized PDEs," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 645–669.

[22] K. Liu et al., "Surrogate modeling of parameterized multi-dimensional premixed combustion with physics-informed neural networks for rapid exploration of design space," *Combustion Flame*, vol. 258, no. 2, pp. 248–267, Nov. 2023.

[23] R. Laubscher, "Simulation of multi-species flow and heat transfer using physics-informed neural networks," *Phys. Fluids*, vol. 34, no. 3, pp. 548–573, Aug. 2021.

[24] Y. Sun, U. Sengupta, and M. Juniper, "Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry," *Comput. Methods Appl. Mech. Eng.*, vol. 411, no. 2, pp. 227–250, Apr. 2023.

[25] N. Rahaman, A. Baratin, M. Lin, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 312–322.

Yayong Yang (Student Member, IEEE) was born in Henan, China. He received the B.S. degree in electrical engineering from Hunan University, Changsha, China, in 2020. He is currently working toward the Ph.D. degree in artificial intelligence with the Institute of Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China.

His current research interests include packaging and integration of SiC power modules, and application of artificial intelligence in power electronics.

Zhiqiang (Jack) Wang (Senior Member, IEEE) received the B.S. degree from Hunan University, Changsha, China, in 2007, and the M.S. degree from Zhejiang University, Hangzhou, China, in 2010, and the Ph.D. degree from the University of Tennessee, Knoxville, TN, USA, in 2015, all in electrical engineering.

He is currently a Full Professor with the Huazhong University of Science and Technology (HUST), Wuhan, China. Prior to joining HUST, he was with the Power Electronics and Electric Machinery Research Center, Oak Ridge National Laboratory, Oak Ridge, TN, USA, as a Postmaster Research Associate, from 2014 to 2015, a full-time R&D Associate Staff Member, from 2015 to 2018, and an R&D Staff Member in 2019. He has also been an Adjunct Professor with the University of Tennessee, Knoxville, TN, USA, since 2018. He has authored and coauthored more than 80 publications in international conferences and journals. His research interests include packaging and integration of wide bandgap power semiconductor devices, and its applications to high temperature, high frequency, and high-density power electronics systems.

Dr. Wang was the recipient of more than 10 awards from ORNL and IEEE. He was the Technical Program Chair for WiPDA-Asia 2021 conference and is currently the Transaction Paper Review Chair for the IEEE IAS Power Electronics Devices and Components Committee.

Yu Liao (Student Member, IEEE) was born in Guangdong, China. He received the B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2021 and 2024, respectively.

His research interests include packaging and integration of SiC power semiconductor modules, and advanced thermal management for power devices.

Wubin Kong (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2009 and 2014, respectively.

Since 2022, he has been a Professor with the Huazhong University of Science and Technology, Wuhan, China. His research interests include high power motor drives and intelligent control motor drive applied in electric vehicles and aircraft electrification.

Xiaojie Shi (Senior Member, IEEE) received the M.S. degree from Zhejiang University, Hangzhou, China, in 2011, and the Ph.D. degree from University of Tennessee, Knoxville, TN, USA, in 2015, both in electrical engineering.

She is currently a Full Professor with the Huazhong University of Science and Technology, Wuhan, China. Prior to joining HUST, she was with the Center for Ultra-Wide-Area Resilient Electric Energy Transmission Networks, University of Tennessee, Knoxville, TN, USA, as a Research Assistant Professor in 2016, and with the Electric Power Research Institute, Knoxville, TN, USA, as an Engineer/Scientist II, from 2017 to 2019, and Engineer/Scientist III from 2020 to 2021. Since 2019, she has also been an Adjunct Professor with the University of Tennessee, Knoxville, TN, USA. She has authored/coauthored more than 60 publications in international conferences and journals. Her research interests include driving and protection of power semiconductor devices, modeling and control of grid-connected power converters, microgrid design, and operation.

Run Hu (Member, IEEE) received the B.S. degree in thermal energy and power engineering and Ph.D. degree in engineering thermophysics from Huazhong University of Science and Technology (HUST) in 2010 and 2015, respectively.

He was a Visiting Scholar with Purdue University in and a JSPS Postdoctoral Fellow with the University of Tokyo, Tokyo, Japan. He is currently a Full Professor with the School of Energy and Power Engineering, HUST. He has authored more than 100 publications in journals. His main research interests include heat and mass transfer, thermal metamaterials and functional devices, and thermal management of optoelectronic devices.

Yonggang Yao (Member, IEEE) received the Ph.D. degree in materials science and engineering from the University of Maryland, College Park, MD, USA, in 2018.

He is currently a Professor with the School of Materials Science and Engineering, Huazhong University of Science and Technology. He has authored or coauthored more than 100 papers in high-profile journals like Science (Cover), Nature, Nature Nanotechnology, Nature Catalysis, and Science Advances, with a total citation of more than 10 000. His group mainly focuses on the transient high-temperature synthesis and data-driven material discovery.

Dr. Yao has also been ranked "Highly-Cited Researchers" by Clarivate. He was the recipient of the 2020 "R&D 100 award," the 2022 Metals Young Investigator Award, 2022 DAMO Young Academy Fellow, and MIT TR35 China.