Thermoelectric Cooler Modeling and Optimization via Surrogate Modeling Using Implicit Physics-Constrained Neural Networks

Liang Chen, *Member, IEEE*, Wentian Jin, *Student Member, IEEE*, Jinwei Zhang, *Student Member, IEEE*, and Sheldon X.-D. Tan, *Senior Member, IEEE*

***Abstract*—Thermoelectric cooler (TEC) is a promising active cooling device to remove the localized hot spots precisely in VLSI chips. In this article, we use a novel implicit physics-constrained neural networks (called IPCNNs) to build a surrogate model for the single TEC device with the reduction from 3-D to 1-D. First, the surrogate model represented by the deep neural networks (DNNs) allows parameterization of key design and running parameters, such as current density, length, and thermal boundary conditions of the TEC. Second, the proposed method tries to partition the physics laws into two different groups, which then are enforced by supervised learning and physics-informed neural networks (PINNs) framework sequentially. Such implicit PCNN scheme can lead to much faster training speed and better convergent accuracy for the unsupervised training. The existing plain PINN enforces all the physics laws via the loss functions and the network tends to have very slow training speed and a large convergent error for large problems. An extreme learning machine (ELM) is used for the networks in the first stage. Compared with fully connected network (FCN) trained by the traditional back-propagation algorithm, ELM can be easily trained and converges much faster. Furthermore, by leveraging the differential nature of the DNN model, we can directly estimate the derivative of the cooling heat flux with respect to current density instead of using a finite difference approximation. The calculated derivatives are used to find the optimal current density to achieve maximum cooling heat flux via Newton's method. Last but not least, we propose a novel hybrid finite element neural network (FENN) method to perform thermal analysis of the VLSI chip system with the TEC device. The DNN model is embedded into COMSOL through the heat flux boundary conditions. Experimental results show that the machine learning-based method can achieve about 8.5× speedup with good accuracy than the COMSOL-based finite element method. Furthermore, the proposed IPCNN is more stable and accurate than the existing PINN. The proposed FENN can have a 5.1× speedup and 5.4× memory reduction over the traditional numerical method.**

***Index Terms*—Heat cooling flux, implicit physics-constrained neural networks (IPCNNs), temperature-dependent materials, thermoelectric cooler (TEC).**

Manuscript received 24 August 2022; revised 16 December 2022 and 3 April 2023; accepted 17 April 2023. Date of publication 21 April 2023; date of current version 20 October 2023. This work was supported in part by NSF under Grant CCF-2007135 and Grant CCF-2113928. This article was recommended by Associate Editor C.-K. Cheng. *(Corresponding author: Sheldon X.-D. Tan.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: stan@ece.ucr.edu).

Digital Object Identifier 10.1109/TCAD.2023.3269385

I. Introduction

THERMOELECTRIC cooler (TEC) is a promising heat removal solution for VLSI chips due to its precise control, compact size, and noiseless operation [1], [2], [3]. With the Peltier effect, TEC can actively provide efficient and localized cooling for hot spots in the chip. Localized heat fluxes of the microprocessors can reach >300 W/cm^2 [1]. The power densities generated by the chip will continue to increase as the technology node advanced into the next generation. To meet the increasing cooling demand, several efforts have been made to improve the maximum cooling heat flux of TEC devices. New thermoelectric materials with a high figure of merit *ZT* were developed and the thickness of thermoelectric materials was reduced [1], [2]. For instance, the TEC with Sb_2Te_3/Bi_2Te_3 superlattice has a cooling heat flux of 258 W/cm^2 [2]. Maximum cooling heat flux is a key performance metric of the single TEC device at a temperature difference across the module of zero [2], [4]. In general, the current density of the single TEC has an optimal value where the cooling heat flux is maximum. Accurate TEC modeling and current density optimization are important to study the performance of the single TEC device.

Based on the energy conversation, simplified energy equilibrium model [1], [2], [3], [4] was first proposed to describe the TEC device by using the approximated expressions. This model can not consider a large thermal gradient and temperature-dependent coefficients [3] because it suffers from low accuracy. In contrast to the simplified energy equilibrium model, partial differential equations (PDEs) [3], [5], [6], [7], [8] describing the TEC effect can predict more accurate results even though the thermal gradient is large and the coefficients are nonlinear.

There are several numerical and analytical methods employed to solve the PDEs for modeling a single TEC device. Numerical methods can perform accurate simulations for large and complex structures with temperature-dependent materials. Antonova and Looman first used the commercial code in ANSYS to perform finite element (FE) analysis for 3-D TEC geometry [5]. Chen et al. [6] implemented 3-D thermoelectric generator (TEG) model in a finite volume method CFD package, which is FLUENT software. Fateh et al. [7] utilized a finite difference method to explore the design and optimization of TEG devices. However, numerical methods require huge

1937-4151 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

computation costs and are very time consuming due to the discretization of both spatial and temporal space. For each new parameter, numerical methods have to assemble a new linear system to recalculate the final results. Such methods are impractical for real-time applications and multiple predictions required by optimization problem. To improve the efficiency, an analytical approach [8], [9] is also used to understand the thermal characteristics of the TEC. Sheikhnejad et al. [8] first reduced 3-D to 1-D with a realistic approximation and then used the separation of variables method to obtain the exact solution of the PDEs. Chen et al. [9] proposed a new boundary condition which considers the TEC effect and derived an analytical solution for optimal current density to achieve the maximum cooling heat flux. However, these analytical works assumed that thermal conductivity, Seebeck coefficient and electrical conductivity are constant. Experimental measurement shows these coefficients are temperature dependent [10]. The analytical methods failed to solve the nonlinear 1-D TEC equation with temperature-dependent coefficients. Therefore, new efficient and accurate methods are highly desired for solving the nonlinear 1-D TEC equations.

Recently, machine learning (ML) methods are increasingly being used to develop a surrogate model for the PDEs in various applications based on the universal approximation and excellent expressivity of neural networks [11], [12], [13], [14], [15]. Therefore, the ML-based surrogate model is an alternative way to describe the nonlinear 1-D TEC equation due to its capability of representing strong nonlinearity and high dimensionality. An ML-based surrogate model approximates a relationship between inputs and outputs of a system. The inputs can be any parameters, such as geometry information, boundary conditions, operation conditions, coefficients, etc. Such ML-based surrogate model can be evaluated several times and efficiently for real-time applications and optimization problem. One of the ML-based approaches to build the surrogate model is physics-informed neural networks (PINNs) [15] which is a supervised learning method. PINN is proposed to integrate the mathematical formula such as PDEs into neural networks. Compared with data-driven methods [12], [13], [14], the class of PINN methods does not need the labeled dataset. Basically, the PINN framework can be viewed as a meshfree method, which employs the meshless points, such as the problem data, the boundary and initial conditions, to produce the final results. The main advantage of the PINN over traditional numerical methods is that first we can add design parameters into the networks so that the trained network can work for all the parameter space in the solutions (no retraining is required), which was demonstrated in works [16], [17], [18]. Second, the PINN can obtain the sensitivity information trivially by simply using back propagation, which is extremely useful for sensitivity-based optimization. For the traditional numerical method, the numerical differential is needed, which requires solving for more solution points and can be expensive. The PINNs method has shown very promising performance in forward and optimization problems [11], [15], [16], [17], [18].

In this work, we propose a novel implicit physics-constrained neural networks (IPCNNs) to build a versatile surrogate model for the nonlinear 1-D TEC PDEs first, which consists of a prebuilt physics-enforced model for coefficients as shown in Fig. 1(b). Our key contributions are summarized as follows.

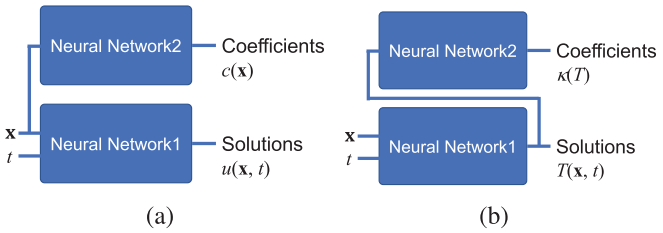


Fig. 1. Two neural networks connected (a) in parallel and (b) in cascade for solving nonlinear partial equations with nonlinear coefficients.

1) We propose to apply the PINN concept to build a parameterized surrogate model for a single TEC device and optimize its cooling heat flux. Based on heat conduction equation, the 1-D TEC equation integrates Peltier effect to provide active cooling heat flux. The resulting PDE surrogate models represented by the deep neural networks (DNNs) allows parameterization of design parameters (such as current density and length) and running parameters (such as thermal boundary conditions) of the TEC. Among them, the current density is the key design parameter in our TEC device optimization.
2) To mitigate the slow convergence of loss function for existing plain PINNs, we propose a novel IPCNN framework to build a parameterized surrogate model for nonlinear 1-D TEC equations. First, we divide the physics requirements into two different groups. In the first stage, an extreme learning machine (ELM) algorithm is employed to approximate temperature-dependent parameters [10] via a supervised learning process with the published data. ELM is easily trained and converges much faster than a fully connected network (FCN) trained by a back-propagation method. Then for the second stage, the physics-constrained loss is applied with the trained ELM from the first stage. In this way, some physics laws are implicitly enforced in the first stage and the number of variables and constraints are significantly reduced for the second stage for solving the overall PDE problem via PINN. Therefore, the two-stage training method leads to much faster training and convergence speed for the unsupervised training.
3) By capitalizing the differential nature of the ML-based surrogate model and parameterization of current density, we can directly estimate the derivative of the cooling heat flux with respect to current density instead of finite difference approximation. Then, we use derivative-based Newton's method to find optimal current density to achieve maximum cooling heat flux with the derivatives. This fast optimization algorithm benefits from the differentiability of the ML-based model.
4) Based on the ML-based surrogate model and parameterization of boundary conditions, we propose a novel hybrid FE neural network (FENN) method to perform thermal analysis of VLSI chip systems with the TEC

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

device. This method adds the ML-based surrogate model into FE-based COMSOL via heat flux boundary conditions. In heat flux boundaries, temperature and heat flux should be continuous. This FENN method can significantly reduce the number of degrees of freedom while maintaining high accuracy.

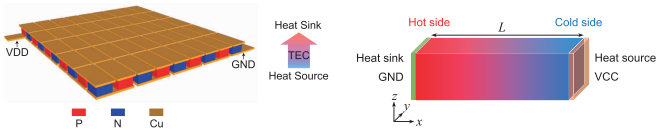
Experimental results show that the ML model has very high accuracy for temperature, its derivatives and cooling heat flux q_c compared with FE-based COMSOL. The ML model can achieve 8.5× speedup over the COMSOL. Our proposed IPCNN is more robust and stable than existing PINN. The proposed FENN can have a 5.1× speedup and 5.4× memory reduction over the traditional numerical method. In addition, simulation considering temperature-dependent parameters can predict the rapid degradation of TEC cooling performance at high temperature, which is more realistic than the results with constant coefficients.

This article is organized as follows: Section II reviews some related prior works. Section III reviews the TEC models. Section IV presents the new IPCNN framework to build a parameterized surrogate model for the nonlinear 1-D TEC equation. Based on the ML model, we calculate the cooling heat flux q_c and further optimize q_c. In Section V, a novel FENN method is proposed to perform thermal analysis of VLSI chip systems with the TEC devices. Experimental results are presented in Section VI. Finally, Section VII concludes this article.

II. Related Prior Works of Machine Learning Approaches

Recently, ML methods have shown breakthroughs in computer vision and natural language processing, which inspires new solutions to solve PDEs for many applications in scientific communities [19]. Furthermore, similar to the analytical methods, the trained ML models are differentiable so that they can calculate the derivatives directly without discretization errors [20], [21]. Several ML methods have been applied to solve heat conduction equation for thermal analysis without the TEC effect, which are divided into two strategies. One approach is the data-driven method which is supervised learning. Zhang et al. [22] employed the FCN to predict the thermal response of many temperature sensors on the processors. Sadiqbatcha et al. [23] applied long short-term memory (LSTM) network to capture dynamic temperature profiles measured by infrared thermal imaging setup. Jin et al. [24] took the performance metrics as input to generate full-chip thermal maps by using generative adversarial networks. Chhabria et al. [25] performed thermal analysis by using a convolutional neural network (CNN) based on encoder–decoder architecture. Those data-driven methods require a database with ground truth to train the model, which restricts their applications in real problems as data generation is a big issue.

On the other hand, an unsupervised learning method, called PINN [15], was proposed recently to tackle the problem of data generation. Based on automatic differentiation, PINN estimates the differential operator and adds the information of physics law, such as governing equations, boundary conditions, and initial conditions, into the loss functions, which is used to train the neural network by back-propagation without a dataset. Cai et al. [26] have applied PINN to various prototype heat transfer problems, including forced and mixed convection, and two-phase Stefan problem. Central ML Team at ANSYS investigates heat transfer in electronic chips using PINN [27]. NVIDIA presents a PINN-based PDE solver, known as Modulus, and uses the toolkit to simulate and optimize heat sink design with the parameterized technique [17]. Those methods tried to seek optimal design simply by looping through all parameter combinations, which is a brute-force and inefficient method. The aforementioned works only consider the constant thermal conductivity in heat transfer problems without the TEC effect. However, temperature-dependent parameters can lead to nonlinearity in the heat conduction equation, which is difficult to be solved. Shukla et al. [28] used two neural networks to represent the solutions of nonlinear elastic wave equations and position-dependent parameters, respectively. The inputs of two neural networks contain the same variables which is the position $\mathbf{x}$. Therefore, the two neural networks are connected in parallel, as shown in Fig. 1(a). Their works [28] and NVIDIA Modulus [17] train two neural networks simultaneously since the two neural networks are connected parallelly. However, for the nonlinear thermal problem, we can not use the two parallel neural networks to represent temperature and coefficients, respectively, because outputs of the first neural network are inputs of the second neural network. Therefore, PINN for the nonlinear thermal analysis of the TEC still needs to be explored.



(a) (b)

Fig. 2. (a) 3-D view of thin-film TEC device. (b) 1-D reduction from 3-D TEC device. Heat source and VCC are placed on the cold side. Heat sink and GND are located on the hot side.

III. Single TEC Device and Its Modeling in Nutshell

Fig. 2(a) shows a detailed 3-D view of the thin-film TEC device, which consists of N-type and P-type materials. These N–P pairs of legs are connected in series. Heat is absorbed from heat sources located at the cold (bottom) side and generated on the hot (top) side with the heat sink based on the thermoelectric effect which is an energy conversion between electric and thermal fields. The complex thermoelectric effect consists of Seebeck, Peltier, Thomson, Joule heating, and Fourier transfer effects.

A Simplified 1-D energy equilibrium model [1], [2], [3], [4] was first proposed and frequently used to predict the performance of such TEC device, as shown in Fig. 2(a). The cooling heat flux at the cold side is represented by

$$q_c = ST_cJ - \frac{1}{2}\frac{J^2L}{\sigma} - \frac{\kappa}{L}(T_h - T_c) \tag{1}$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

where κ is the thermal conductivity, S is the Seebeck coefficient, σ is the electrical conductivity, T_c and T_h are the temperatures on the cold and hot sides, respectively, J is the current density, and L is the thickness of the TEC. However, this model is overly simplified and gives an approximated expression. The results provided by the model are inaccurate when the thermal gradient is large and coefficients are temperature dependent.

Therefore, a PDEs-based model [3], [5], [6], [7], [8], which describes the TEC effect more accurately, was developed to understand the thermal characteristics of the thermoelectric applications. The 3-D TEC array device in Fig. 2(a) only has vertical heat flux. Heat does not transfer along horizontal directions. All thermal legs have the same current density as they are connected in series. For single TEC device simulation and optimization, the boundary conditions are uniform. Therefore, to simulate single 3-D TEC device, we can reduce the 3-D TEC device into 1-D problem, as shown in Fig. 2(b). The reduction from 3-D to 1-D is the same as the previous approach [8]. The 1-D PDEs to describe 3-D TEC can be expressed as follows:

$$\begin{aligned}
&\frac{\partial}{\partial x}\left[-\kappa(T)\frac{\partial T}{\partial x}+S(T)TJ\right]=\frac{J^2}{\sigma(T)}\\
&\text{BC}: T(x=0)=T_0 \text{ K}\\
&\text{BC}: -n\cdot\left(-\kappa(T)\frac{\partial T}{\partial x}+S(T)TJ\right)\Big|_{x=L}=q_c \text{ W/m}^2 \quad (2)
\end{aligned}$$

where $T(x)$ is the temperature, J is the current density along x-direction, $\kappa(T)$ is the temperature-dependent thermal conductivity, $S(T)$ is the temperature-dependent Seebeck coefficient, and $\sigma(T)$ is the temperature-dependent electrical conductivity, n is the unit outward vector of the boundary surface, T_0 is the ambient temperature, and q_c is the heat flux at the boundary. The governing equation considers the Peltier effect, which is different from the traditional heat conduction. We can parameterize the ambient temperature T_0 and heat flux q_c boundary conditions to consider different boundary conditions.

Cooling heat flux represents the performance of the TEC. The maximum cooling heat flux $q_{\max}$ is defined as the maximum cooling heat flux that the thermoelectric module is capable of providing at a temperature difference across the module of zero [2], [4]. To estimate $q_{\max}$, the temperature at two sides should be the same, $T(0)=T(L)=T_0$ K, which are different from the boundary conditions of (2). $T(0)$ represents boundary conditions at the heat sink. $T(L)$ represents boundary conditions at the heat source. Therefore, one straightforward method uses an optimization algorithm to find q_c to meet the constraint $T(0)=T(L)=T_0$ K. Another method is that we directly set the boundary conditions as $T(0)=T(L)=T_0$ K and cooling heat flux is estimated by

$$q_c=-n\cdot\left(-\kappa(T)\frac{\partial T}{\partial x}+S(T)TJ\right). \quad (3)$$

Based on cooling heat flux q_c with constraint $T(0)=T(L)$, we optimize the current density to achieve maximum cooling heat flux $q_{\max}$. Note that the smaller length L and higher ambient temperature T_0 of TEC lead to the higher maximum cooling heat flux. Therefore, we do not need to optimize the length and ambient temperature.

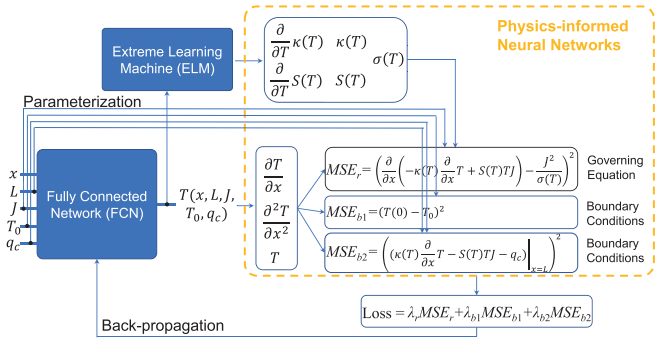


Fig. 3. Overall framework of IPCNNs for temperature-dependent TEC. In the first stage, based on experimental data of materials, an ELM is trained first to approximate temperature-dependent parameters (such as thermal conductivity, Seebeck coefficient, and electrical conductivity) and their derivatives. In the second stage, an FCN is employed to model temperature T with respect to position x, length L, current density J, temperature boundary condition T_0, and heat flux boundary condition q_c. The two networks are then applied to construct the 1-D governing equation loss MSE_r, boundary conditions loss MSE_{b1} and MSE_{b2} with their derivatives obtained by automatic differentiation. The only FCN is trained by Adam optimization algorithm based on back-propagation of loss function.

IV. Implicit Physics-Constrained Neural Networks for Temperature-Dependent TEC

A schematic diagram of IPCNN to build a surrogate model for temperature-dependent TEC is shown in Fig. 3. In the proposed IPCNN, we propose to use two cascade neural networks to describe the nonlinear TEC problem and parameterize the current density, length, and boundary conditions of the TEC. Among them, current density is the key design parameter in our TEC device optimization. Then, corresponding 1-D TEC equations should be normalized as we normalize the inputs and outputs of neural networks. After that, we propose to apply the two-step training method to train the two cascade neural networks. The details are illustrated below.

A. Parameterization of Length, Current Density, and Boundary Conditions

Universal approximation theorem [29] shows that neural networks can be employed to approximate any complex and nonlinear functions, such as the solution $T(x, L, J, T_0, q_c)$ of PDEs (2). Any parameter impacting on the output can be taken as the input of neural networks. In this article, we take length, current density, and boundary conditions as the inputs. Therefore, we have

$$T=\mathcal{N}(x, L, J, T_0, q_c) \quad (4)$$

where the notation $\mathcal{N}$ represents the FCN to approximate temperature T with respect to position x, length L, current density J, temperature boundary condition T_0, and heat flux boundary condition q_c.

With this parameterized technique, the IPCNN method has several advantages over traditional numerical methods: first, we can add design parameters into the networks so that the

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

trained network can work for all the parameter space in the solutions (no retraining is required), which was demonstrated in works [16], [17], [18]. For this problem, we parameterize the current density as it is a key design parameter for our optimization problem. Second, the IPCNN with parameterized technique can obtain the sensitivity information trivially by simply using back propagation, which is extremely useful for sensitivity-based optimization as demonstrated in our work. For a traditional numerical method, the numerical differential is needed, which requires solving for more solution points and can be expensive. Section IV-E describes the optimization of current density for TEC based on the neural network model.

B. Normalization of Inputs and Outputs

In this article, the length of TEC is in the range of 0.8×10^{-3}–1×10^{-3} m and position x is in the range of 0–1×10^{-3}. Then we determine the range of current density, temperature, and heat flux that we are interested in. The current density J ranges between -1×10^7 and 0 A/m^2. The temperature T is the range of 300–600 K. The ambient temperature T_0 is the range of 300–400 K. The heat flux q_c ranges between 0.8×10^5–1.2×10^5 W/m^2. Therefore, we normalize the inputs and outputs data to the interval $[-2, 2]$, which is easy to be trained for neural networks. The scaling transformations of x, J, T, and q_c are formulated as follows:

$$\hat{x} = \alpha x, \ \hat{J} = \beta J, \ \hat{T} = \gamma T, \ \hat{q}_c = \theta q_c \tag{5}$$

where $\alpha = 10^3$, $\beta = 10^{-7}$, $\gamma = 1/300$, and $\theta = 10^{-5}$ are scaling factors. Once we normalized the inputs and outputs, the 1-D TEC PDEs will also be changed by adding scaling factors. Based on (5), the 1-D TEC PDEs (2) can be rewritten as follows:

$$\begin{aligned}
&-\frac{\alpha^2}{\gamma^2}\frac{\partial \kappa(T)}{\partial T}\left(\frac{\partial \hat{T}}{\partial \hat{x}}\right)^2 - \frac{\alpha^2}{\gamma}\kappa(T)\frac{\partial^2 \hat{T}}{\partial \hat{x}^2} \\
&\quad + \frac{\alpha}{\gamma^2\beta}\hat{T}\hat{J}\frac{\partial S(T)}{\partial T}\frac{\partial \hat{T}}{\partial \hat{x}} + \frac{\alpha}{\gamma\beta}S(T)\hat{J}\frac{\partial \hat{T}}{\partial \hat{x}} = \frac{1}{\beta^2}\frac{\hat{J}^2}{\sigma(T)} \\
&\text{BC}: \hat{T}(0) = \gamma T_0 \\
&\text{BC}: \ -n \cdot \left(-\frac{\alpha}{\gamma}\kappa(T)\frac{\partial \hat{T}}{\partial \hat{x}} + \frac{1}{\gamma\beta}S(T)\hat{T}\hat{J}\right)\Big|_{\hat{x}=\alpha L} = \frac{\hat{q}_c}{\theta}.
\end{aligned} \tag{6}$$

Then, the FCN can be formulated by

$$\hat{T} = \mathcal{N}\left(\hat{x}, \hat{L}, \hat{J}, \hat{T}_0, \hat{q_c}\right). \tag{7}$$

C. Two-Step Training Method

After the preprocessing, we can apply the novel IPCNN to build a parameterized surrogate model for the transformed 1-D TEC PDEs (6) using the two-step training method.

From (6), we can observe that the derivatives $\partial\kappa(T)/\partial T$ and $\partial S(T)/\partial T$ are required to be evaluated. However, the FCN $\mathcal{N}(\hat{x}, t, \hat{J})$ can only provide the derivatives $\partial\hat{T}/\partial\hat{x}$, $\partial^2\hat{T}/\partial\hat{x}^2$, and $\partial\hat{T}/\partial\hat{t}$. Therefore, in the first stage, based on supervised learning method, a simple and differentiable approximator, which can provide the corresponding derivatives, should be used to fit the temperature-dependent thermal conductivity $\kappa(T)$, Seebeck coefficient $S(T)$ and electrical conductivity $\sigma(T)$, which are obtained from experimental data [10]. The ELM algorithm [30] is used since it is a convenient function approximator using a feed-forward neural network with one hidden layer. It is an optimization-free neural network, which means its weights and biases can be directly determined by solving linear equations. The ELM can be expressed as follows:

$$f(x) = \sum_{i=1}^{n} v_i \phi(w_i x + b_i) + b \tag{8}$$

where v_i and w_i are weights, b_i and b are biases, $\phi(\cdot)$ is activation function, and n is the number of neurons in the hidden layer. w_i and b_i are randomly generated. In this article, $\tanh(\cdot)$ is chosen as the activation function. The data pairs with m sampling points are $(x_j, f(x_j))$ and $j = 1, 2, \ldots, m$. Then, (8) can be written in matrix form

$$\mathbf{H}\mathbf{v} = \boldsymbol{\beta} \tag{9}$$

where

$$\mathbf{H} = \begin{bmatrix}
\phi(w_1x_1 + b_1) & \phi(w_2x_1 + b_2) & \cdots & \phi(w_nx_1 + b_n) & 1 \\
\phi(w_1x_2 + b_1) & \phi(w_2x_2 + b_2) & \cdots & \phi(w_nx_2 + b_n) & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\phi(w_1x_m + b_1) & \phi(w_2x_m + b_2) & \cdots & \phi(w_nx_m + b_n) & 1
\end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n & b \end{bmatrix}^T$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix}^T.$$

Then, the unknown parameters v_i and b can be estimated by

$$\mathbf{v} = \text{pinv}(\mathbf{H})\boldsymbol{\beta} \tag{10}$$

where pinv represents the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ since the matrix $\mathbf{H}$ is not to be square in most of the cases. Finally, we can calculate the derivative of function $f(x)$, which is given by

$$\frac{\partial f(x)}{\partial x} = \sum_{i=1}^{n} v_i w_i \phi'(w_i x + b_i). \tag{11}$$

FCN can be also used to approximate the temperature-dependent parameters. However, ELM can be trained faster than FCN by using the optimization-free algorithm, called Moore–Penrose generalized inverse pinv. Therefore, we employ ELM to fit experimental data [10]. Once ELM is trained, the weights v_i and bias b are fixed. In the next step, we only train the FCN using a physics-constrained loss function and do not change the weights v_i and bias b of ELM.

A physics-constrained loss function is an unsupervised learning strategy, which does not require labeled training data because they add physics information of governing equation, boundary conditions and initial conditions into the loss function of FCN [15]. In the second stage, the parameters of FCN can only be trained by minimizing the physics-constrained loss function using the unsupervised learning method. For this TEC problem, the physics-constrained loss function is represented by

$$\mathcal{L} = \lambda_r MSE_r + \lambda_{b1} MSE_{b1} + \lambda_{b2} MSE_{b2} \tag{12}$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

where

$$MSE_r = \frac{1}{N_r}\sum_{i=1}^{N_r}\left|\left(-\frac{\alpha^2}{\gamma^2}\frac{\partial\kappa(T)}{\partial T}\left(\frac{\partial\hat{T}}{\partial\hat{x}}\right)^2 - \frac{\alpha^2}{\gamma}\kappa(T)\frac{\partial^2\hat{T}}{\partial\hat{x}^2} + \frac{\alpha}{\gamma^2\beta}\hat{T}\hat{J}\frac{\partial S(T)}{\partial T}\frac{\partial\hat{T}}{\partial\hat{x}} + \frac{\alpha}{\gamma\beta}S(T)\hat{J}\frac{\partial\hat{T}}{\partial\hat{x}} - \frac{1}{\beta^2}\frac{\hat{J}^2}{\sigma(T)}\right)\Big|_{\left(\hat{x}_i,\hat{L}_i,\hat{J}_i,\hat{T}_{0,i},\hat{q}_{c,i}\right)}\right|^2$$

$$MSE_{b1} = \frac{1}{N_{b1}}\sum_{i=1}^{N_{b1}}\left|\left(\hat{T}-\gamma T_0\right)|_{(\hat{x}=0)}\right|^2$$

$$MSE_{b2} = \frac{1}{N_{b2}}\sum_{i=1}^{N_{b2}}\left|\left(\frac{\alpha}{\gamma}\kappa(T)\frac{\partial\hat{T}}{\partial\hat{x}} - \frac{1}{\gamma\beta}S(T)\hat{T}\hat{J} - \frac{\hat{q}_c}{\theta}\right)|_{(\hat{x}=\alpha L)}\right|^2$$

where MSE_r, MSE_{b1}, and MSE_{b2} are the mean-square errors of governing equation, left boundary condition, and right boundary condition, respectively, λ_r, λ_{b1}, and λ_{b2} are the penalty coefficients, N_r is the number of sampling points, which are randomly selected in the combinational domains of space, length, current densities, and boundary conditions, N_{b1} and N_{b2} are the numbers of sampling points, which are selected randomly in the combinational domains of length, current densities, temperature and heat flux boundary conditions at the boundaries $x = 0$ and at the boundaries $x = L$, respectively.

D. Comparison of Proposed Method and Existing PINN

Existing PINN adds all physics laws into loss function which is represented by

$$\mathcal{L} = \lambda_r MSE_r + \lambda_{b1}MSE_{b1} + \lambda_{b2}MSE_{b2} + \lambda_{\text{data}}MSE_{\text{data}} \quad (13)$$

where

$$MSE_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}}\left|\mathcal{N}_{\text{data}}(T_i) - [\kappa(T_i), S(T_i), \sigma(T_i)]\right|^2$$

where $\mathcal{N}_{\text{data}}$ represents the additional FCN, MSE_{data} is the mean-square error of $\mathcal{N}_{\text{data}}$ and experimental data, λ_{data} is the penalty coefficient, N_{data} represents the number of experimental data. Therefore, existing PINN trains the two FCNs simultaneously by using the loss function (13), which is called the one-step training method. In our IPCNN framework, we train ELM and FCN separately, which is called two-step training method. In the first stage, we use the Moore–Penrose generalized inverse algorithm to train ELM using experimental data [10] and determine the weights and bias of ELM so that it cannot be changed later. In the second stage, we only train the FCN $\mathcal{N}$ using a physics-constrained loss function (12).

In the IPCNN, ELM has a single hidden layer with four neurons to approximate the temperature-dependent parameters. The FCN has six layers with 512 neurons. For the existing PINN method, the two FCNs have six layers with 512 neurons. This is because the default FCN is set as six layers with 512 neurons in the NVIDIA modulus [17]. The starting learning rate is 10^{-3}. Both IPCNN and the existing PINN methods use the cascaded neural network to solve this problem because the parallel neural networks can not solve the nonlinear equation with temperature-dependent parameters.

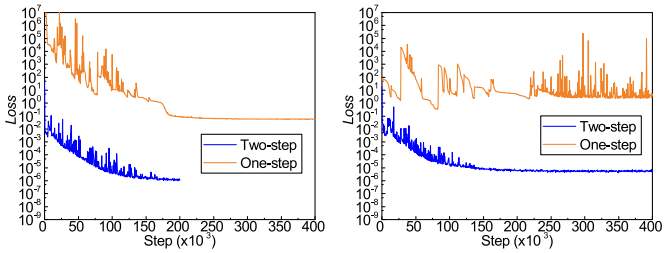


Fig. 4. Loss history of one-step and two-step training methods for the nonlinear TEC problem with the length range of (a) 0.8–1 mm and (b) 0.05–1.2 mm.

Fig. 4(a) shows loss history of one-step and two-step training methods applied for the nonlinear TEC problem with the length range of 0.8–1 mm. The current density J, ambient temperature T_0, and heat flux q_c are also parameterized. As we can see, the loss function (13) of one-step training method converges to 10^{-1}, which is a very large error. The loss function (12) of two-step training method converges successfully from 10 to 10^{-6} with much better accuracy.

To further demonstrate the stability and robustness of IPCNN and the existing PINN, we increase the length range to 0.05–1.2 mm. The other settings remain the same. The loss history is shown in Fig. 4(b). The loss function (12) of IPCNN converges successfully from 54 to 4×10^{-6}. However, the loss function (13) of existing PINN is oscillating and fails to converge. This is because the two FCNs are trained simultaneously, which increases the searching space of the optimization algorithm. Due to the two FCNs connected in cascade, one FCN can impact another FCN a lot and it is difficult for the optimizer to find the final solution. Therefore, our proposed IPCNN method is more stable and accurate than the existing PINN because we ensure that the FCN is not affected by the ELM.

As the range of length increases, the accuracy of IPCNN is reduced because the loss function increases from 10^{-6} to 4×10^{-6}. To improve the accuracy of neural networks for a wide range of lengths, we can use several neural networks to represent different subregions. For example, we can divide the range [0.05, 1.2] into six subregions ([0.05, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0], and [1.0, 1.2]) and use six individual neural networks for each subregion.

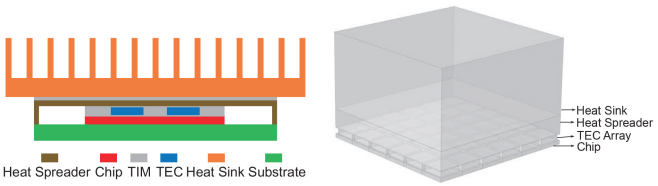
We highlight the features of proposed IPCNN: parameterization of several important parameters, normalized TEC equations, two cascaded neural networks, two-step training method, and an ELM algorithm which is used to replace FCN as ELM has a faster training speed over FCN.

E. Current Density Optimization

Once we obtain the ML-based parameterized surrogate model $\hat{T} = \mathcal{N}(\hat{x}, \hat{J})$ using IPCNN, we can calculate the cooling heat flux q_c based on the differentiability of FCN

$$q_c = \frac{\alpha}{\gamma}\kappa\left(\hat{T}\right)\mathcal{N}_{\hat{x}}\left(\hat{x}=\alpha L, \hat{J}\right) - \frac{1}{\gamma\beta}S(T)\hat{T}\hat{J} \quad (14)$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 5. (a) Side view of the VLSI chip system with the TEC device. (b) 3-D FE-based model in COMSOL for VLSI chip system with the TEC device.

where $\mathcal{N}_{\hat{x}}$ denotes the derivative of $\mathcal{N}$ with respect to $\hat{x}$, which can be easily obtained by automatic differentiation.

When $\partial q_c/\partial J = 0$, the cooling heat flux q_c reaches the maximum values. Newton's method is a quick iterative algorithm to find the solution of $\partial q_c/\partial J = 0$, which is formulated as follows:

$$J_{n+1} = J_n - \frac{\frac{\partial q_c}{\partial J}|_{J=J_n}}{\frac{\partial^2 q_c}{\partial J^2}|_{J=J_n}} \tag{15}$$

where

$$\frac{\partial q_c}{\partial J} = \kappa(T)\frac{\partial^2 T}{\partial x \partial J} - S(T)T, \frac{\partial^2 q_c}{\partial J^2} = \kappa(T)\frac{\partial^3 T}{\partial x \partial J^2}.$$

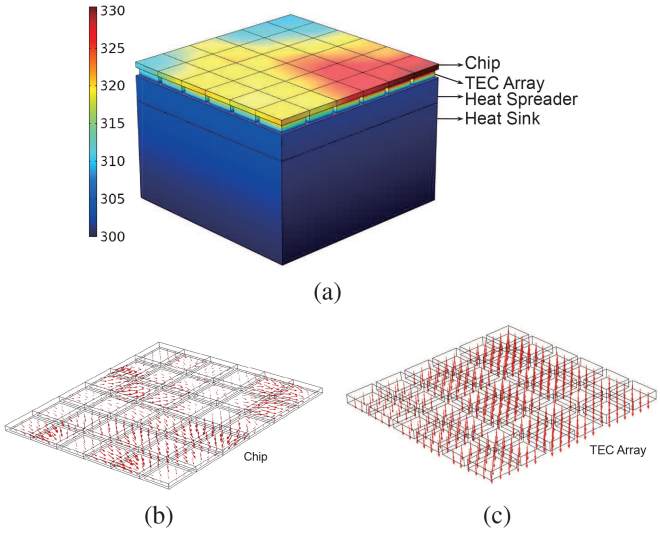
Based on the FCN model, (15) is rewritten as follows:

$$J_{n+1} = J_n - \frac{\alpha\beta\kappa(T)\mathcal{N}_{\hat{x}\hat{J}}\left(\hat{x}=\alpha L, \hat{J}_n\right) - S(T)\hat{T}}{\alpha\beta^2\kappa(T)\mathcal{N}_{\hat{x}\hat{J}^2}\left(\hat{x}=\alpha L, \hat{J}_n\right)} \tag{16}$$

where $\mathcal{N}_{\hat{x}\hat{J}}$ is the derivative of $\mathcal{N}_{\hat{x}}$ with respect to $\hat{J}$, and $\mathcal{N}_{\hat{x}\hat{J}^2}$ is the derivative of $\mathcal{N}_{\hat{x}\hat{J}}$ with respect to $\hat{J}$. The FCN model uses automatic differentiation to estimate $\mathcal{N}_{\hat{x}\hat{J}}$ and $\mathcal{N}_{\hat{x}\hat{J}^2}$ without discretization error. Newton's method requires fewer evaluations of the FCN model compared to the brute-force method for this optimization. What is more, benefiting from the differential nature of the FCN, it can directly provide the derivatives instead of the finite difference approximation which needs more evaluations. Therefore, the ML-based surrogate model shows promises for the optimization problem.

V. Application of TEC ML-Based Model in VLSI Chip

Fig. 5(a) shows that thin-film TEC devices are employed to remove the heat generated by VLSI chips. The thermal simulation for the VLSI system with the TEC device by using COMSOL software is complicated since it involves two kinds of equations: 1) heat conduction equation and 2) TEC equation. For convenience, we simplified the structure in Fig. 5(a) as the 3-D model in Fig. 5(b). We build this thermal model by using "General Form PDE" in the Mathematics module of COMSOL. In the TEC devices, the 3-D TEC equations are employed to consider TEC effects. In the other parts, 3-D heat conduction equations are used to describe the heat transfer in solid. Then, we configure the power map and boundary conditions and start the thermal simulation in COMSOL. The thermal results are shown in Fig. 6(a). Based on the temperature distribution, we observed that the temperature on the TEC legs is uniform in the y–z plane and increases proportionally



(a) (b) (c)

Fig. 6. (a) Temperature distribution of the VLSI chip system with the TEC device. (b) Heat flux distribution for the chip layer. (c) Heat flux distribution for the TEC array layer.

along the x-axis. We plot the heat flux for chip and TEC array in Fig. 6(b) and (c), respectively. As we can see, there are temperature gradients of x-, y-, and z- all directions in the chip layer. Therefore, 3-D thermal simulation should be considered for the chip layer. But in the TEC layer, heat mainly flows in the vertical direction due to Peltier cooling effect. As a result, it is reasonable to assume that the heat flux mainly exists in x-direction and heat does not transfer along y- and z- direction. Therefore, we can use 1-D TEC ML-based model to describe the single 3-D TEC device.

To apply the 1-D TEC ML-based model in the 3-D VLSI chip system, we set one side of the TEC leg as temperature boundary condition of 303.5 K since this side is connected with the heat spreader and heat sink (temperature is uniform). If we know the temperature T on another side of the TEC leg, the heat flux q can be calculated by the 1-D TEC ML-based model. By using the differentiability of FCN, we can obtain the heat flux

$$q = -\frac{\alpha}{\gamma}\kappa\left(\hat{T}\right)\mathcal{N}_{\hat{x}}\left(\hat{T}\right) + \frac{1}{\gamma\beta}S(T)\hat{T}\hat{J} \tag{17}$$

where $\hat{T}$ is the temperature of one side which is connected to the chip layer. We parameterize the temperature $\hat{T}$ to calculate the heat flux of the TEC leg based on different temperature values. Then, we proposed to combine the ML-based model and FE-based model to simulate the VLSI chip system with TEC devices based on interior boundary conditions (temperature and heat flux are continuous), as shown in Fig. 7. In the FE-based COMSOL, we set the heat flux boundary condition of the chip layer as the (17) obtained by the neural network. Then, we can significantly decrease the number of degrees of freedom by using this combination method (called FENN). We will show the accuracy and efficiency of the proposed FENN method in the following result Section VI.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

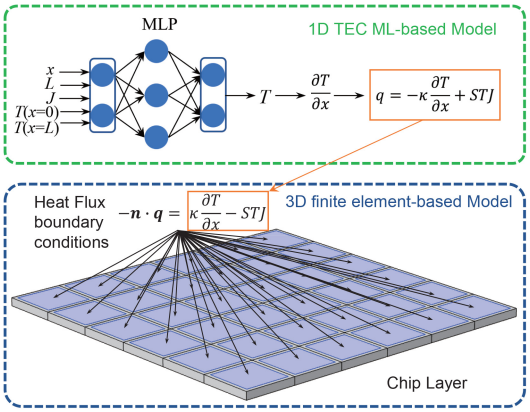


Fig. 7. Combination of 1-D TEC ML-based model and 3-D FE-based model for VLSI chip systems with the TEC device.

VI. Experimental Results and Discussions

In this section, we first use ELM to approximate the temperature-dependent parameters. Then, we demonstrate the accuracy and speed of the parameterized surrogate model FCN obtained by the IPCNN framework. With the parameterized FCN, we employ Newton's method to optimize current density to achieve the maximum cooling heat flux. Finally, we emphasize the importance of considering the temperature-dependent parameters compared with the results calculated by using constant parameters.

The parameterized IPCNN is implemented in the NVIDIA Modulus platform [17], which is developed for solving PDEs and built on TensorFlow. ELM is trained in MATLAB and then integrated into Modulus. The model is trained and tested on a Linux desktop with AMD Ryzen 5 1600 3.2-GHz processors and NVIDIA GTX 1060 GPU with 6G memory.

A. Temperature-Dependent Parameters Approximated by ELM

Fig. 8 shows the experimental data of one TEC material's parameters, such as thermal conductivity, Seebeck coefficient, and electrical conductivity, which are obtained from the research article [10]. A fast ELM algorithm is employed to fit the parameters and can output the derivatives of $\kappa(T)$, $S(T)$, and $\sigma(T)$ with respect to T. It can be seen from Fig. 8 that the fitting curves exactly fit the experimental data and are smooth.

B. Accuracy and Speed of the IPCNN Surrogate Model

To validate the accuracy of the new IPCNN framework, we first compare the temperature $T(x, J)$ obtained by FCN and COMSOL with the length $L = 1$ mm, ambient temperature $T_0 = 300$ K, and heat flux $q_c = 1 \times 10^5$ W/m^2, as shown in Fig. 9(a)–(c). The results from IPCNN agree well with the solutions from COMSOL. The maximum absolute error is 1.8 K, which is small compared with the maximum temperature 397 K. Based on the differentiability of FCN, we also validate the accuracy of the derivative of T with respect to x in Fig. 9(d)–(f) since it is used to compute cooling heat flux q_c. As we can see, the derivatives obtained by FCN and COMSOL are matched very well. Therefore, the derivatives calculated by FCN are also convincing and accurate.

TABLE I
Speed Comparison of FCN and COMSOL

	FCN	COMSOL	Speedup
Fig. 9(a) and Fig. 9(b) Estimation	0.94 s	8 s	8.5×

The FCN obtained by IPCNN in this work can parameterize current density, length, and thermal boundary conditions. But we want to stress that it can include many other parameters as needed. Due to these parameterizations, the trained FCN model can be viewed as a library and can be used for many different applications and optimization problems as long as all the relevant design and running parameters are included.

To further validate accuracy of the FCN, we sweep length L, ambient temperature T_0, and heat flux q_c. Fig. 10 shows that temperature on the line with different length L is estimated by IPCNN and COMSOL. They have a good agreement with each other for each length. We can also change boundary conditions, such as ambient temperature T_0 and heat flux q_c. We use FCN model to predict temperature distributions with different T_0 and q_c, as shown in Figs. 11 and 12. As we can see, the FCN model has a good accuracy by comparison to COMSOL. Therefore, the ML-based method shows the capability of solving high-dimensional PDEs with several parameters. Note that Figs. 9–12 are obtained by one FCN model which is only trained once.

To demonstrate the efficiency of FCN trained by the IPCNN framework, we take Fig. 9(a) and (b) as an example. We build 1-D TEC model (2) in COMSOL. The current density interval is $\Delta J = 1 \times 10^4$ A/m^2 and space interval is $\Delta x = 20$ μm. The mesh element size is set as 10 μm and the number of degrees of freedom is 201. To make their results consistent, the FCN predicts the temperature at 501 × 991 points. Table I shows the time to obtain Fig. 9(a) and (b) using FCN and COMSOL, respectively. The FCN can achieve 8.5× speedup over FE-based COMSOL.

C. Optimization of Cooling Heat Flux

To estimate the maximum cooling heat flux, we set the boundary conditions as $T(0) = T(L) = T_0$. Then, we use the IPCNN framework to obtain the FCN model. Fig. 13(a) shows that temperature on the line with different current densities is estimated by IPCNN and COMSOL. They have a good agreement with each other for each current density. As we can see, the IPCNN method can deal with different combinations of boundary conditions.

Fig. 13(b) shows the first and second derivatives of q_c with respect to J, which are obtained by using the FCN model. We can use Newton's method to find optimal current density quickly with the derivatives. The iteration procedure using FCN is shown in Table II. N_{sim} represents the total number of FCN model evaluations. It takes four steps to find the optimal current density $J_{\text{opt}} = -8.0942 \times 10^6$ A/m^2. For numerical methods, like COMSOL, the derivatives cannot be calculated directly and are usually approximated by the finite difference method. $\partial q_c/\partial J$ and $\partial^2 q_c/\partial J^2$ can be approximated by

$$\frac{\partial q_c}{\partial J} \approx \frac{q_c(J+h) - q_c(J-h)}{2h}$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

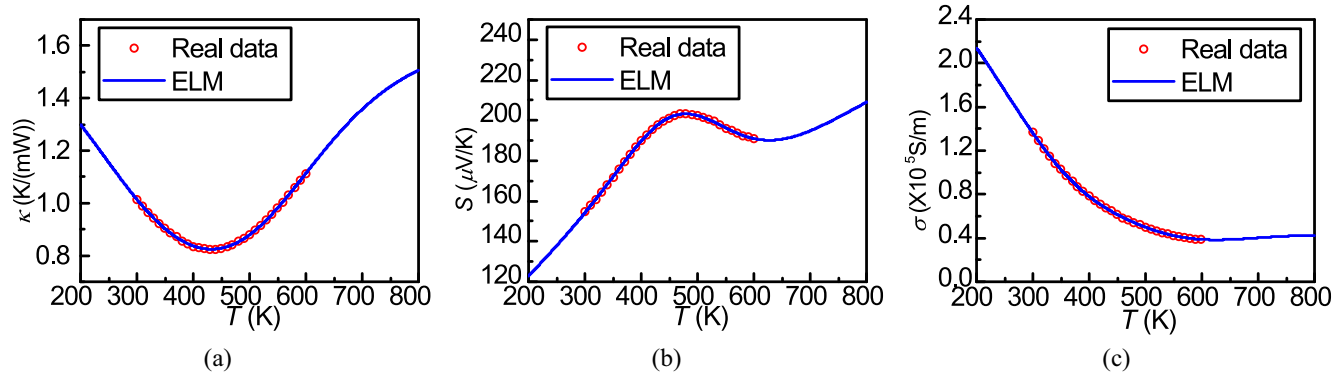


Fig. 8. Real data of (a) thermal conductivity, (b) Seebeck coefficient, and (c) electrical conductivity with different temperatures, which are approximated by ELM.

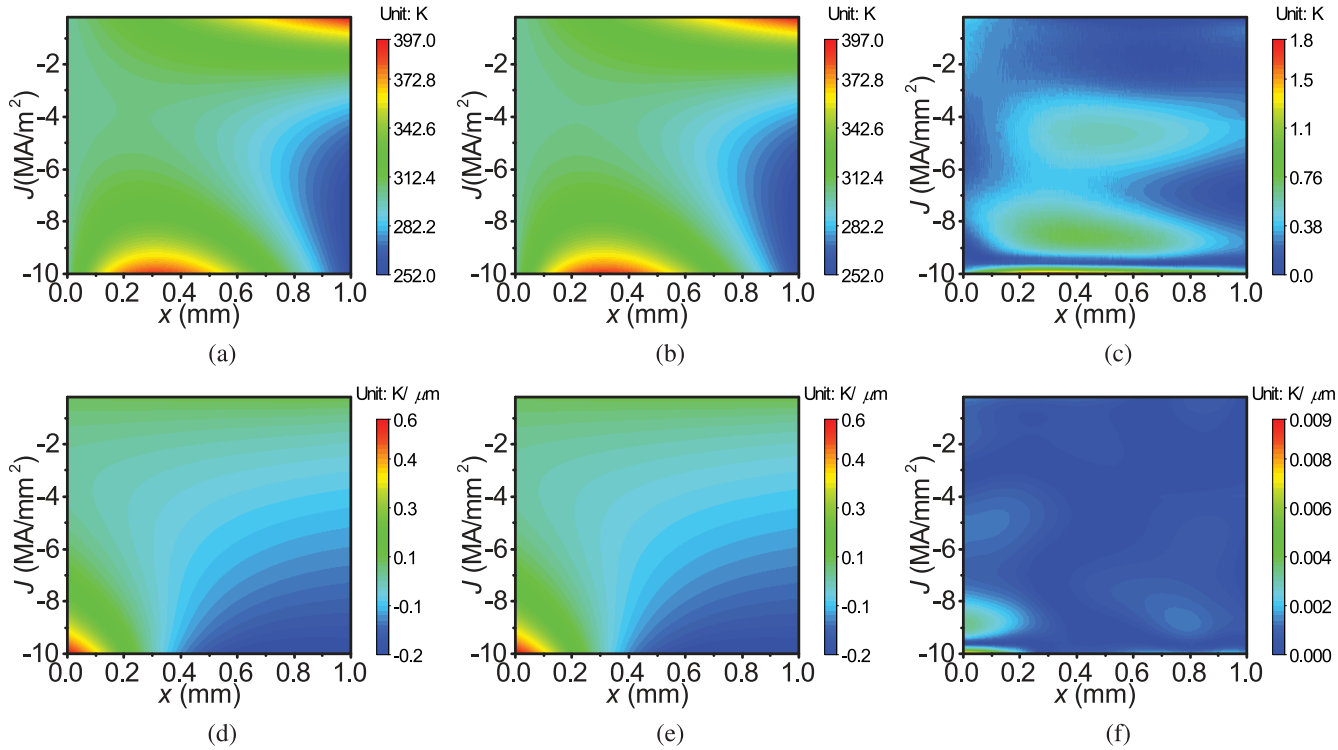


Fig. 9. Comparison of the IPCNN results with the solution obtained by using COMSOL. (a) IPCNN solution $T_{\text{FCN}}(x, J)$, (b) COMSOL solution $T_{\text{COMSOL}}(x, J)$, (c) absolute error $|T_{\text{FCN}} - T_{\text{COMSOL}}|$. (d) IPCNN solution $\partial T_{\text{FCN}}(x, J)/\partial x$, (e) COMSOL solution $\partial T_{\text{COMSOL}}(x, J)/\partial x$, and (f) absolute error $|\partial T_{\text{FCN}}/\partial x - \partial T_{\text{COMSOL}}/\partial x|$.

TABLE II
ITERATION PROCEDURE OF NEWTON METHOD USING FCN

Iteration	Current Density (A/m^2)	Cooling heat flux (W/m^2)	$\left\lvert\frac{q_c^{n+1}-q_c^n}{q_c^n}\right\rvert$	N_{sim}
1	-1.0000×10^5	4.8982×10^3	/	1
2	-7.1466×10^6	1.7745×10^5	3522.76%	2
3	-8.2167×10^6	1.8023×10^5	1.57%	3
4	-8.0942×10^6	1.8030×10^5	0.039%	4

TABLE III
ITERATION PROCEDURE OF NEWTON METHOD USING COMSOL

Iteration	Current Density (A/m^2)	Cooling heat flux (W/m^2)	$\left\lvert\frac{q_c^{n+1}-q_c^n}{q_c^n}\right\rvert$	N_{sim}
1	-1.0000×10^5	4.5894×10^3	/	3
2	-6.3802×10^6	1.7147×10^5	3636.22%	6
3	-8.0135×10^6	1.8021×10^5	5.1%	9
4	-8.0921×10^6	1.8023×10^5	0.011%	12

$$\frac{\partial^2 q_c}{\partial J^2} \approx \frac{q_c(J+h) - 2q_c(J) + q_c(J-h)}{h^2} \tag{18}$$

where h is a small value. In this article, h is set to 1×10^5. The number of COMSOL evaluations is 3 for each iteration of Newton's method because the derivatives require the values of $q_c(J+h)$, $q_c(J)$, and $q_c(J-h)$. Table III shows the iteration procedure using COMSOL. It also takes four steps to find the optimal current density $J_{\text{opt}} = -8.0921 \times 10^6$ A/m^2. But N_{sim} for COMSOL is 12, which is three times larger than N_{sim} of FCN. Therefore, for this optimization problem, the ML method can achieve $3 \times 8.5 = 25.5\times$ speedup over the conventional numerical method.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

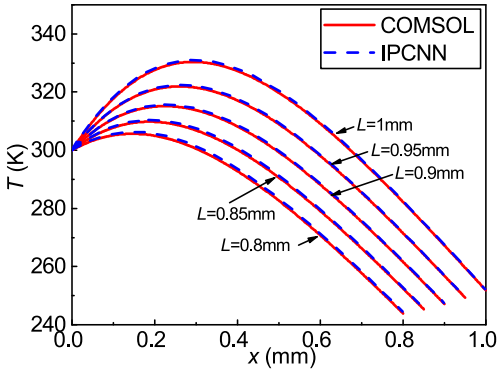


Fig. 10. Temperature comparison of IPCNN and COMSOL one the line with different length L. The current density $J = -8 \times 10^6$ A/m^2, the ambient temperature $T_0 = 300$ K, and heat flux $q_c = 1 \times 10^5$ W/m^2.

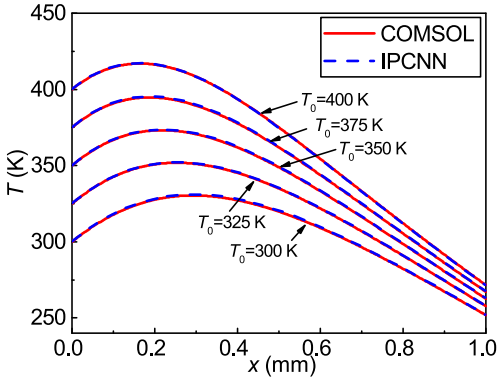


Fig. 11. Temperature comparison of IPCNN and COMSOL on the line with different temperature boundary conditions T_0. The current density $J = -8 \times 10^6$ A/m^2, the length $L = 1$ mm, and heat flux $q_c = 1 \times 10^5$ W/m^2.

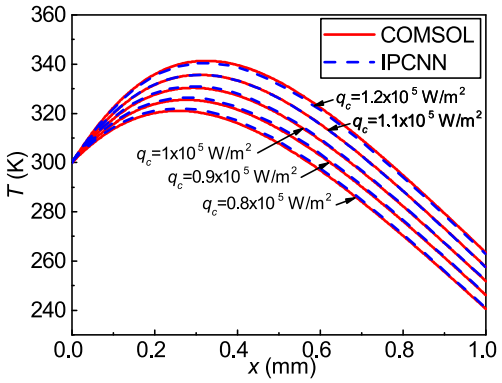


Fig. 12. Temperature comparison of IPCNN and COMSOL on the line with different heat flux boundary conditions q_c. The current density $J = -8 \times 10^6$ A/m^2, the length $L = 1$ mm, and the ambient temperature $T_0 = 300$ K.

D. Impact of Temperature-Dependent Parameters on Cooling Heat Flux

Fig. 14 shows the cooling heat flux estimated with temperature-dependent and constant temperature. As we can see, the cooling heat flux obtained by FCN agrees well with the results from COMSOL. For constant temperature, we set the temperature as 375 K. The notation "TD" represents temperature-dependent temperature and "CT" denotes constant temperature. As we can see, when the current density is not too large ($J < 1 \times 10^7$ A/m^2), their cooling heat fluxes are similar. However, when the current density is large and Joule heating effect leads to a large temperature rise, cooling heat flux for the temperature-dependent case drops quickly, which is quite different from the constant case. This is because high temperature can degrade the performance of TEC dramatically. That is the reason why we do not observe that the cooling heat flux increases infinitely as described in [9] even though the dimensionless figure of merit $ZT_0 > 1$. Therefore, considering temperature-dependent parameters are important for accurate modeling and simulation of TEC.

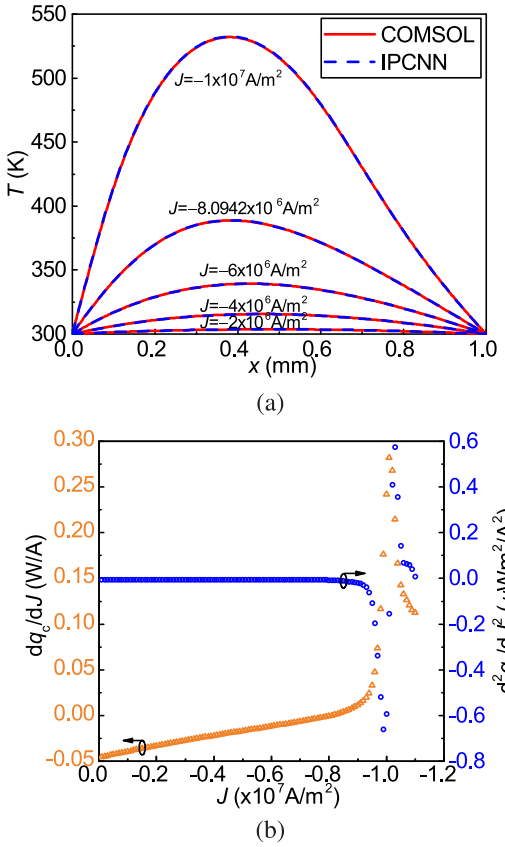


Fig. 13. (a) Temperature comparison of IPCNN and COMSOL on the line with different current densities J. (b) $\partial q_c/\partial J$ and $\partial^2 q_c/\partial J^2$ with different current densities.

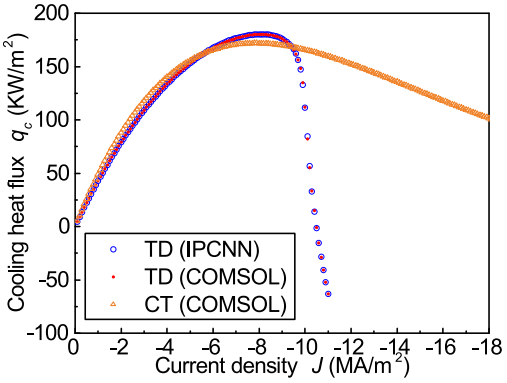


Fig. 14. Cooling heat flux q_c with temperature-dependent and constant temperature.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

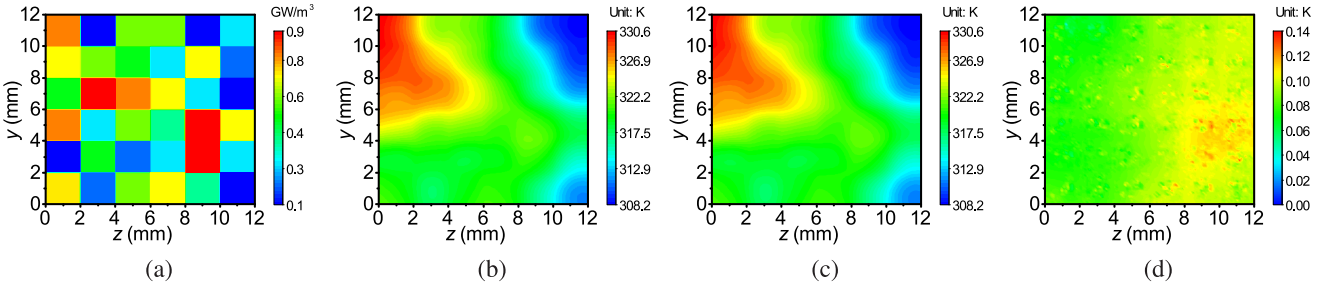


Fig. 15. Thermal map comparison between the proposed FENN and COMSOL. (a) Power density in chip layer, (b) FENN solution $T_{\text{FENN}}(y, z)$, (c) COMSOL solution $T_{\text{COMSOL}}(y, z)$, (d) absolute error $|T_{\text{FENN}} - T_{\text{COMSOL}}|$. ($x = 0.15$ mm).

TABLE IV
SPEED COMPARISON OF THE PROPOSED FENN AND COMSOL

	Time	Number of degrees of freedom
FENN	10 s	65083
COMSOL	51 s	351773
	5.1×	5.4×

E. Accuracy and Efficiency of the Proposed Finite Element Neural Network

We proposed a novel FENN to perform thermal simulations of the VLSI chip system with the TEC device. This method combines the FE method and the ML-based model through heat flux boundary conditions (convection boundary condition). First, we employed the proposed IPCNN with the boundary conditions parameterization to obtain the 1-D TEC ML-based model. Then, based on the heat flux calculated by the neural network, the proposed FENN method only simulates the chip layer with the boundary conditions, which significantly reduces the number of degrees of freedom. We use the example shown in Fig. 6(a) to demonstrate the accuracy and efficiency of the proposed FENN method. Fig. 15(a) shows the power density that we generate randomly in the chip layer. Note that the proposed FENN method is not limited to this case and can be also used to perform thermal analysis for power densities of some benchmarks and industrial cases. Fig. 15 shows the chip thermal maps in the chip layer ($x = 0.15$ mm) obtained by the proposed FENN method and COMSOL. As we can see, the results from FENN and COMSOL match each other perfectly. Therefore, the TEC reduction from 3-D to 1-D we used is reasonable and the proposed FENN is very accurate compared with ground truth. Table IV shows the speed comparison of the FENN and COMSOL. With high accuracy, the FENN can achieve 5.1× speedup and 5.4× memory reduction over the COMSOL. Our proposed ML-based model can improve the efficiency of traditional numerical methods with high accuracy.

VII. CONCLUSION

In this article, we proposed a novel IPCNN to develop a parameterized surrogate model for nonlinear 1-D PDEs describing the single TEC device by using the two cascaded neural network architecture. The PDE surrogate model parameterizes current density, length, and thermal boundary conditions of the TEC. Among them, the current density is the key design parameter in our TEC device optimization. The loss function for existing PINN converges to a large error since it deals with large numbers of learnable parameters at the same time. To mitigate the problem, we proposed IPCNN concepts in which the two-stage training method for two separate neural networks are carried out sequentially. The two-stage training method leads to much faster training speed and much better accuracy for the unsupervised training. On top of this, benefiting from the automatic differentiation of the ML-based model and parameterization of current density, we calculated the derivative of the cooling heat flux with respect to current density. Then, the derivative-based Newton's method is employed to find the optimal current density to achieve maximum cooling heat flux efficiently. We also proposed a novel hybrid FENN method to perform the thermal analysis of the VLSI chip system with TEC devices. Experimental results show that the ML-based method can achieve 8.5× speedup with good accuracy in the simulation of the single TEC device compared with the conventional numerical method. Our proposed two-step training method is more stable and accurate than the one-step training method. The proposed FENN can have a 5.1× speedup and 5.4× memory reduction over the traditional numerical method. In addition, it is very important to consider the temperature-dependent parameters because they impact the maximum cooling heat flux significantly.

REFERENCES

[1] I. Chowdhury et al., "On-chip cooling by superlattice-based thin-film thermoelectrics," *Nat. Nanotechnol.*, vol. 4, no. 4, pp. 235–238, Jan. 2009.
[2] G. Bulman et al., "Superlattice-based thin-film thermoelectric modules with high cooling fluxes," *Nat. Commun.*, vol. 7, no. 1, pp. 1–13, Jan. 2016.
[3] R. A. Kishore, A. Nozariasbmarz, B. Poudel, M. Sanghadasa, and S. Priya, "Ultra-high performance wearable thermoelectric coolers with less materials," *Nat. Commun.*, vol. 10, no. 1, pp. 1–13, Apr. 2019.
[4] D. M. Rowe, *Thermoelectrics Handbook: Macro to Nano*. Boca Raton, FL, USA: CRC Press, 2005.
[5] E. E. Antonova and D. C. Looman, "Finite elements for thermoelectric device analysis in ANSYS," in *Proc. 24th Int. Conf. Thermoelectrics (ICT)*, 2005, pp. 215–218.
[6] M. Chen, L. A. Rosendahl, and T. Condra, "A three-dimensional numerical model of thermoelectric generators in fluid power systems," *Int. J. Heat Mass Transf.*, vol. 54, nos. 1–3, pp. 345–355, 2011.
[7] H. Fateh, C. A. Baker, M. J. Hall, and L. Shi, "High fidelity finite difference model for exploring multi-parameter thermoelectric generator design space," *Appl. Energy*, vol. 129, pp. 373–383, Sep. 2014.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.

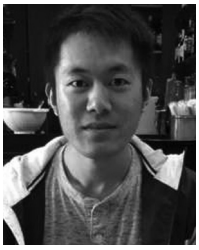
[8] Y. Sheikhnejad, R. Bastos, Z. Vujicic, A. Shahpari, and A. Teixeira, "Laser thermal tuning by transient analytical analysis of peltier device," *IEEE Photon. J.*, vol. 9, no. 3, pp. 1–13, Jun. 2017.
[9] L. Chen, S. Sadiqbatcha, H. Amrouch, and S. X.-D. Tan, "Electrothermal simulation and optimal design of thermoelectric cooler using analytical approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 9, pp. 3066–3077, Sep. 2022.
[10] F. Hao et al., "High efficiency Bi_2Te_3-based materials and devices for thermoelectric power generation between 100 and 300° C," *Energy Environ. Sci.*, vol. 9, pp. 3120–3127, Aug. 2016.
[11] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.
[12] Z. Li et al., "Fourier neural operator for parametric partial differential equations," 2020, *arXiv:2010.08895*.
[13] Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," *J. Comput. Phys.*, vol. 399, Dec. 2019, Art. no. 108925.
[14] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nat. Mach. Intell.*, vol. 3, no. 3, pp. 218–229, 2021.
[15] 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," *J. Computat. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
[16] L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Comput. Methods Appl. Mech. Eng.*, vol. 361, Apr. 2020, Art. no. 112732.
[17] O. Hennigh et al., "NVIDIA SimNet^{TM}: An AI-accelerated multi-physics simulation framework," in *Computational Science (ICCS)*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds. Cham, Switzerland: Springer Int., 2021, pp. 447–461.
[18] W. Jin, S. Peng, and S. X.-D. Tan, "Data-driven electrostatics analysis based on physics-constrained deep learning," in *Proc. Des. Autom. Test Eur. Conf. (DATE)*, Feb. 2021, pp. 1–6.
[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–538, 1989.
[21] H. Zhou, W. Jin, and S. X.-D. Tan, "GridNet: Fast data-driven EM-induced IR drop prediction and localized fixing for on-chip power grid networks," in *Proc. 39th Int. Conf. Comput.-Aided Des.*, Nov. 2020, pp. 1–9.
[22] K. Zhang et al., "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 2, pp. 405–419, Feb. 2018.
[23] S. Sadiqbatcha, Y. Zhao, J. Zhang, H. Amrouch, J. Henkel, and S. X.-D. Tan, "Machine learning based online full-chip heatmap estimation," in *Proc. 25th Asia South Pac. Des. Autom. Conf. (ASP-DAC)*, 2020, pp. 229–234.
[24] W. Jin, S. Sadiqbatcha, J. Zhang, and S. X.-D. Tan, "Full-chip thermal map estimation for commercial multi-core CPUs with generative adversarial learning," in *Proc. Int. Conf. Comput.-Aided Des. (ICCAD)*, Nov. 2020, pp. 1–9.
[25] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and IR drop analysis using convolutional encoder–decoder networks," in *Proc. Asia South Pac. Des. Autom. Conf. (ASPDAC)*, 2021, pp. 690–696.
[26] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *J. Heat Transfer*, vol. 143, no. 6, pp. 1–15, Apr. 2021.
[27] H. He and J. Pathak, "An unsupervised learning approach to solving heat equations on chip based on auto encoder and image gradient," 2020, *arXiv:2007.09684*.
[28] K. Shukla, A. D. Jagtap, J. L. Blackshire, D. Sparkman, and G. Em Karniadakis, "A physics-informed neural network for quantifying the microstructural properties of polycrystalline nickel using ultrasound data: A promising approach for solving inverse problems," *IEEE Signal Process. Mag.*, vol. 39, no. 1, pp. 68–77, Jan. 2022.
[29] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
[30] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

Liang Chen (Member, IEEE) was born in 1992. He received the B.E. degree in electromagnetic field and wireless technology from Northwestern Polytechnical University, Xi'an, China, in 2015, and the Ph.D. degree in electronic science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2020.

From 2020 to 2022, he was a Postdoctoral Researcher with the VLSI System and Computation Laboratory, Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His current research interests include machine learning for VLSI reliability analysis, signal integrity of high-speed interconnects, electrothermal co-simulation of 3-D integrated packages, and electromigration reliability.

Wentian Jin (Student Member, IEEE) received the B.S. and M.S. degrees in instrument science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA.

He is currently a Student Researcher with the VLSI Systems and Computation Laboratory, University of California at Riverside. His current research interests are electronic design automation and applied machine learning in VLSI reliability analysis.



Jinwei Zhang (Student Member, IEEE) received the B.S. degree in electrical and control engineering from the Department of Mechatronics, Beijing Institute of Technology, Beijing, China, in 2014, the M.S. degree in electrical engineering from Washington University in Saint Louis, Saint Louis, MI, USA, in 2016, and the Ph.D. degree from the VLSI System and Computation Lab, Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA, in Winter 2023.

His research interests are aligned with electronic design automation, system-level dynamic thermal/reliability management, power modeling and optimization for VLSI and commercial multicore processors with machine learning techniques at the system level.

Sheldon X.-D. Tan (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, IA, USA, in 1999.

He is a Professor with the Department of Electrical Engineering, University of California at Riverside, Riverside, CA, USA, where he also is a cooperative Faculty Member with the Department of Computer Science and Engineering. He was a Visiting Professor with Kyoto University, Kyoto, Japan, as a JSPS Fellow from December 2017 to January 2018. His research interests include machine and deep learning for VLSI reliability modeling and optimization at circuit and system levels, machine learning for circuit and thermal simulation, thermal modeling, optimization and dynamic thermal management for many-core processors, efficient hardware for machine learning and AI, and parallel computing and simulation based on GPU and multicore systems. He has published more than 330 technical papers and has coauthored six books on those areas.

Dr. Tan received the NSF CAREER Award in 2004. He received the Best Paper Awards from ICSICT'18, ASICON'17, ICCD'07, and DAC'09. He also received the Honorable Mention Best Paper Award from SMACD'18. He is serving as the TPC Chair for ASPDAC 2021 and the TPC Vice Chair for ASPDAC 2020. He is serving or served as the Editor-in-Chief for *Integration, the VLSI Journal* (Elsevier), the Associate Editor for four journals: IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, *ACM Transactions on Design Automation of Electronic Systems* (TODAES), *Microelectronics Reliability* (Elsevier), and *Electronics Microelectronics and Optoelectronics Section* (MDPI).

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 20,2024 at 05:15:44 UTC from IEEE Xplore. Restrictions apply.