PISOV: Physics-Informed Separation of Variables Solvers for Full-Chip Thermal Analysis

Liang Chen, *Member, IEEE*, Wenxing Zhu, Min Tang, *Senior Member, IEEE*, Sheldon X.-D. Tan, *Senior Member, IEEE*, Jun-Fa Mao, *Fellow, IEEE*, and Jianhua Zhang

***Abstract*—Thermal issues are becoming increasingly critical due to rising power densities in high-performance chip design. The need for fast and precise full-chip thermal analysis is evident. Although machine learning (ML)-based methods have been widely used in thermal simulation, their training time remains a challenge. In this article, we proposed a novel physics-informed separation of variables solver (PISOV) to significantly reduce training time for fast full-chip thermal analysis. Inspired by the recently proposed ThermPINN, we employ a least-square regression method to calculate the unknown coefficients of the cosine series. The proposed PISOV method combines physics-informed neural network (PINN) and separation of variables (SOVs) methods. Due to the matrix-solving method of PISOV, its speed is much faster than that of ThermPINN. On top of PISOV, we parameterize effective convection coefficients and power values for surrogate model-based uncertainty quantification (UQ) analysis by using neural networks, a task that cannot be accomplished by the SOV method. In the parameterized PISOV, we only need to calculate once to obtain all parameterized results of the hyperdimensional partial differential equations. Additionally, we study the impact of sampling methods (such as grid, uniform, Sobol, Latin hypercube sampling (LHS), Halton, and Hammersly) and hybrid sampling methods on the accuracy of PISOV and parameterized PISOV. Numerical results show that PISOV can achieve a speedup of 245×, and 10^4× over ThermPINN, and PINN, respectively. Among different sampling methods, the Hammersley sampling method yields the best accuracy.**

***Index Terms*—Full-chip, parameterization technique, physics-informed separation of variables method, thermal analysis, training time.**

Received 21 March 2024; revised 12 October 2024; accepted 21 November 2024. Date of publication 27 November 2024; date of current version 23 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62188102 and Grant 92473105; in part by the State Key Laboratory of Radio Frequency Heterogeneous Integration (Open Scientific Research Program) under Grant KF2024005; and in part by the Joint Foundation of Key Laboratory of Shanghai Jiao Tong University-Xidian University, Ministry of Education. This article was recommended by Associate Editor I. Vatajelu. *(Corresponding authors: Jianhua Zhang; Min Tang.)*

Liang Chen and Jianhua Zhang are with the School of Microelectronics, Shanghai University, Shanghai 201800, China (e-mail: lchenshu@shu.edu.cn; jhzhang@shu.edu.cn).

Wenxing Zhu is with the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China.

Min Tang is with the State Key Laboratory of Radio Frequency Heterogeneous Integration, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: tm222@sjtu.edu.cn).

Sheldon X.-D. Tan is with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA.

Jun-Fa Mao is with the State Key Laboratory of Radio Frequency Heterogeneous Integration, Shenzhen University, Shenzhen 518060, Guangdong, China.

Digital Object Identifier 10.1109/TCAD.2024.3506867

I. Introduction

THERMAL issues are becoming increasingly critical due to the rising power densities driven by the demands of high-performance computing, artificial intelligence (AI), and other computational applications [1]. Elevated temperatures not only degrade VLSI chip performance but also diminish its lifespan. Several strategies have been proposed to enhance thermal reliability, such as multi/many-core techniques, task migration methods, dynamic voltage and frequency scaling (DVFS) techniques, and thermal cooling techniques [2], [3], [4], [5]. To address these temperature-related challenges, efficient and precise full-chip thermal estimation methods are essential for the thermal design and management of VLSI Chip.

Typically, commercial software relies on numerical methods, such as finite element [6], finite difference [7], [8], [9], and finite volume methods [10], due to their high accuracy and ability to handle complex structures. However, numerical methods require significant computational costs, including large memory requirements and the use of numerous computing cores. Consequently, there is a growing interest in analytical methods for VLSI chip thermal analysis, which offer advantages in terms of speed and precision. The separation of variables (SOVs) method [11] is one such analytical approach used to solve heat conduction and provide thermal profiles. In this method, temperature is represented as a series of cosine functions, and the coefficients are determined through integration operations. Another analytical method is the use of Green's functions [12], [13], which are determined based on boundary conditions. Once Green's functions are obtained, thermal maps can be predicted multiple times without the need for recalculating Green's functions.

The performance of VLSI chips is increasingly affected by fabrication process variations, particularly as technology nodes advance to 5 nm [14]. The effective convection coefficient m varies with the fabrication process, as it is determined by materials and dimensions. Accounting for these variation effects is crucial for precise full-chip thermal estimation in a thermal-aware design flow, which involves solving hyperdimensional partial differential equations (PDEs). To evaluate the impact of process variations, it is a common practice to generate a set of random variation parameters and conduct thermal simulations, which is uncertainty quantification (UQ) analysis [15] by using Monte Carlo (MC) simulation [16]. However, both numerical and analytical methods require recalculating the temperature distribution each time the parameters change due to the fabrication process, resulting in significant time consumption.

1937-4151 © 2024 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 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

In recent years, neural networks have been utilized to address highperdimensional and nonlinear PDEs, leveraging the computational power of high-performance GPUs and the flexibility of deep learning frameworks. Scientific machine learning (SciML) has emerged as a promising PDE solver, revolutionizing thermal simulations. Two categories of methods have gained prominence: 1) data-driven methods and 2) physics-informed neural networks (PINNs) [17], [18], [19], [20], [21], [22]. Data-driven methods rely solely on labeled data to train machine learning (ML) models [17], [18], [19]. However, generating labeled data can be challenging or expensive, limiting the practical applicability of data-driven approaches. To overcome this limitation, PINN was proposed, integrating domain knowledge and physical laws into the loss function to solve nonlinear PDEs through unsupervised learning, without the need for labeled data [20], [21], [22]. However, plain PINN often demands substantial training time to achieve a specific level of accuracy, particularly for complex engineering problems. Convergence can also be challenging when dealing with intricate structures. To address these issues, the novel ThermPINN method combines the SOVs method with PINN using back-propagation techniques [23]. ThermPINN significantly reduces the number of learning parameters, leading to substantial acceleration. Nonetheless, it still requires a significant amount of time to determine the coefficients.

In this work, we present a novel physics-informed SOVs (PISOVs) framework to perform fast thermal simulations. Unlike the conventional back-propagation method, we employ the Moore–Penrose generalized inverse algorithm to obtain results quickly. Furthermore, the parameterized PISOV method enables the consideration of variations in effective convection coefficients m. Our key contributions are summarized as follows.

1) We propose a novel PISOV framework for fast full-chip thermal simulations based on the previous ThermPINN approach and the Moore–Penrose generalized inverse algorithm instead of the back-propagation method, which can significantly reduce the training time. To further enhance the efficiency of matrix assembly, we develop a hierarchical method for cosine operations, eliminating redundant calculations and thereby significantly improving computational efficiency.
2) To account for fabrication process variations, we introduce a parameterized technique into the PISOV framework for solving hyperdimensional PDEs, a task that is beyond the capability of the SOV method. Notably, this marks the first implementation of parameterized techniques within PINN utilizing the Moore–Penrose generalized inverse algorithm. Also, we develop a fast matrix assembly method.
3) Finally, we investigate the impact of sampling methods on the accuracy of the proposed PISOV method. The examined sampling methods include Grid, Uniform, Sobol, Latin hypercube sampling (LHS), Halton, and Hammersley distributions. Then, a hybrid sampling technique that integrates the aforementioned examined sampling approaches with an adaptive sampling method is investigated for the meshless PISOV method.
4) Numerical results demonstrate that the PISOV method can achieve speedup factors of 245×, and $10^4\times$ over ThermPINN, and PINN approaches, respectively. Furthermore, the parameterized PISOV exhibits remarkable speed enhancements of 230× and 5000× over parameterized ThermPINN and parameterized PINN, respectively. Among the various sampling methods investigated, the Hammersley sampling method exhibits the highest accuracy.

This article is structured as follows. Section II provides an overview of relevant prior research. In Section III, we delve into the SOV method for full-chip thermal simulation. Section IV introduces the novel PISOV framework for solving the heat conduction equation. Section V details the parameterization of effective convection coefficients and power values based on PISOV. Experimental results are presented in Section VI. Finally, Section VII offers concluding remarks for this article.

II. Related Prior Works of Machine Learning Approaches

Recently, ML methods have witnessed remarkable advancements in computer vision and natural language processing, providing inspiration for novel approaches to solving PDEs in various scientific applications [24]. In the domain of thermal analysis, several ML techniques have been employed to address the heat conduction equation, and they can be categorized into two primary strategies: 1) data-driven methods and 2) PINNs.

One approach entails data-driven methods, which involve supervised learning. Zhang et al. [25] utilized fully convolutional networks (FCNs) to predict the thermal response of numerous temperature sensors on processors. Sadiqbatcha et al. [26] applied long-short-term-memory (LSTM) networks to capture dynamic temperature profiles acquired through infrared thermal imaging. Jin et al. [27] used generative adversarial networks to generate full-chip thermal maps, taking performance metrics as input. Chhabria et al. [28] conducted a thermal analysis using a convolutional neural network (CNN) with an encoder–decoder architecture. Lu et al. [29] employed transformers to predict real-time thermal profiles for AMD multicore CPUs with high accuracy. However, these models require retraining when there are significant changes in the floorplan of the target chip. To address this issue, Wen et al. [18] divided the entire chip into smaller regions (tiles) where deep neural network (DNN)-based solvers are applied to enhance model transferability. Chen et al. [19] applied graph convolution networks (GCNs) to represent compact thermal models and generated thermal maps for chips of varying sizes, capitalizing on the transferability of GCN. Nevertheless, data-driven methods necessitate a database with ground truth for model training, which limits their practical utility, as data generation may not always be feasible or may incur substantial costs.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

On the other hand, unsupervised learning methods, particularly the recently proposed PINN concept [20], have emerged as a compelling solution to address the challenge of data generation. Similar to analytical methods, neural networks are inherently differentiable, allowing them to directly calculate derivatives without discretization errors [30], [31]. Leveraging automatic differentiation, PINN integrates physics laws, including governing equations, boundary conditions, and initial conditions, into loss functions. These augmented loss functions are then utilized in backpropagation to train the neural network without requiring PDE simulation data. However, upon completing the training process, conventional PINN models obtain PDE results with specific parameters and boundary/initial conditions. This outcome deviates from the original objective of ML models in data-driven methods, which aim to solve hyperdimensional PDEs encompassing various scenarios with diverse parameters and boundary/initial conditions.

To address this challenge, various parameterized PINN methods have been developed to handle parameterized variations, optimization, and UQ. Sun et al. [32] employed the PINN concept to address UQ issues in fluid flows described by Navier–Stokes PDEs. This approach introduced additional parameters into neural networks to model parameter variations, demonstrating that PINN-based models can offer significant speed improvements over traditional numerical approaches. The parameterized PINN paradigm is well demonstrated in tackling electrostatics problems [33]. Cai et al. [34] have successfully applied the PINN methodology to a diverse range of prototype heat transfer scenarios, encompassing forced and mixed convection, as well as the two-phase Stefan problem. Notably, NVIDIA contributes to this trajectory by introducing a PINN-based PDE solver named Modulus, which aids in the simulation and optimization of heat sink designs using parameterized techniques [35]. More recently, building on DeepONet [36], Liu et al. [22] proposed the DeepOHeat framework for fast thermal analysis of 3-D ICs, transitioning from the function space of several PDE configurations to the function space of the solution, considering different initial/boundary conditions. However, these methods often exhibit slow training speeds and poor convergence, particularly when applied to large-scale problems.

To deal with these problems, Chen et al. [23] proposed a novel parameterized framework known as ThermPINN for performing full-chip thermal analysis with varying effective convection coefficients and ambient temperatures. ThermPINN leverages discrete cosine neural networks (DCNs) to represent temperature distributions. Derived from the SOV method, DCN automatically enforces boundary conditions, with the loss function exclusively encompassing the governing equation. ThermPINN eliminates the need to determine optimal penalty coefficients for balancing the governing equation and boundary conditions, a task previously addressed by Sun et al. [32] through the introduction of penalty coefficients λ. However, the penalty coefficient method alone cannot guarantee complete initial conditions/boundary conditions (IC/BC) compliance. Subsequently, the concept of hard constraints was introduced to consolidate multiple losses into a single loss function using the augmented Lagrangian method [37], [38]. Nonetheless, this approach still requires explicit consideration of boundary and initial conditions in the loss functions. ThermPINN effectively resolves these issues, leading to enhanced accuracy and convergence for large-scale problems. Additionally, DCN has a smaller number of learning weights compared to FCN, with the majority of these weights being very small. This characteristic contributes to faster training speeds.

ThermPINN, however, relies on training through backpropagation methods, which can be highly time-consuming. A recent approach called the physics-informed extreme learning machine (PIELM) was introduced to accelerate the PINN method using the Moore–Penrose generalized inverse algorithm [39]. Nevertheless, PIELM utilizes a neural network to approximate temperature distributions, resulting in a higher number of learning parameters compared to DCN. Therefore, in this article, we propose a novel PISOV to significantly improve solving speed by combining ThermPINN with the Moore–Penrose generalized inverse algorithm within the framework of PIELM while maintaining high accuracy. Then, we extend PISOV to a parameterized version to account for parameter variations. Furthermore, we investigate the influence of different sampling methods on the method's accuracy.

III. Thermal Modeling for Full Chip

The chip's functional units produce substantial amounts of heat, resulting in nonuniform temperature distributions. The steady-state temperature profile of the full chip is determined by the thermal equation [11], [12]

$$\nabla \cdot (-\kappa \nabla T(\mathbf{r})) = g(\mathbf{r}) \tag{1}$$

where $\mathbf{r}$ represents the position (x, y, z), $T(\mathbf{r})$ denotes the temperature, $g(\mathbf{r})$ is the power density, and κ stands for the thermal conductivity. As for the boundary conditions, the lower surface of the chip is defined as a convection boundary condition with the convection coefficient h, while the remaining surfaces are assigned adiabatic boundary conditions, expressed as

$$\left.\frac{\partial T}{\partial x}\right|_{x=0,a} = \left.\frac{\partial T}{\partial y}\right|_{y=0,b} = \left.\frac{\partial T}{\partial z}\right|_{z=0} = 0$$
$$\left.\kappa \frac{\partial T}{\partial z}\right|_{z=c} = h(T|_{z=c} - T_0) \tag{2}$$

where T_0 is the ambient temperature. The chip system comprises multiple layers, including the thermal interface material (TIM) layer, through-silicon via (TSV) layer, chip layer, and so on. Due to the multilayer structure, we can simplify the 3-D problem to a 2-D problem. The reduced thermal equations are expressed as [40], [41]

$$\frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa \frac{\partial T}{\partial y}\right) - m^2(T - T_0) = -g(x, y),$$
$$\text{BC: } \left.\frac{\partial T}{\partial x}\right|_{x=0,a} = \left.\frac{\partial T}{\partial y}\right|_{y=0,b} = 0 \tag{3}$$

where m is the effective convection coefficient, which describes the vertical heat transfer behavior.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

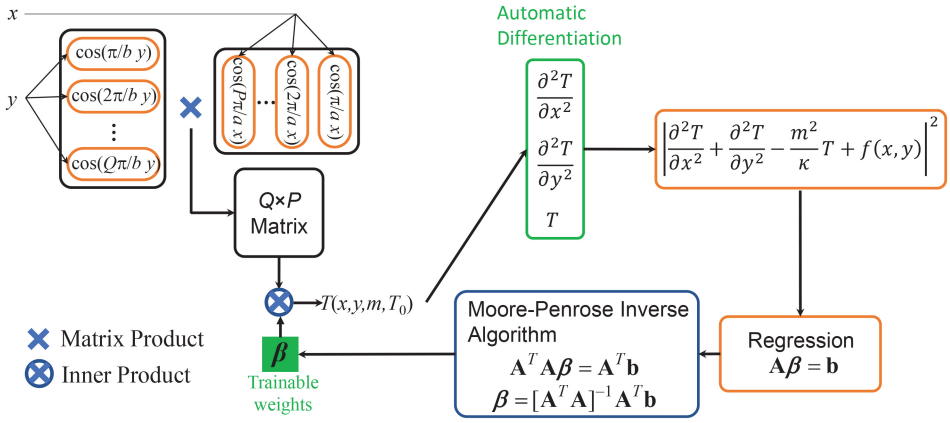


Fig. 1. Overall framework of the PISOV method.

The SOV method is a highly efficient and precise method to solve the thermal equation (3) with constant thermal conductivity and power density [41]. Based on the boundary conditions, the temperature distribution can be expressed as

$$T(x,y)=\sum_{q=0}^{Q}\sum_{p=0}^{P}C_{pq}\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right) \tag{4}$$

where P and Q represent the truncated number of series along x- and y-directions, respectively, a and b are the widths of the chip along x- and y-directions, respectively, and C_{pq} is the coefficients to be determined by governing Equation of (3). The C_{pq} can be calculated analytically by using the formula [41]

$$C_{pq}=\begin{cases}\frac{1}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy,\\ (p=0 \text{ and } q=0);\\ \frac{2}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy,\\ (p>0 \text{ and } q=0 \;||\; p=0 \text{ and } q>0);\\ \frac{4}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy,\\ (p>0 \text{ and } q>0)\end{cases} \tag{5}$$

where

$$f(x,y)=\frac{g(x,y)+m^2T_0}{\kappa}. \tag{6}$$

The detailed derivation process of the SOV method is provided in the Appendix. We have developed a fast SOV algorithm utilizing grid sampling points, which is outlined in Algorithm 1. This method employs an $M\times P$ matrix $\mathbf{F}$ to store computed results, effectively preventing redundant calculations and significantly enhancing the algorithm's speed.

Algorithm 1: SOVs Method

Data: $M\times N$ grid sampling points (x_i, y_i) with intervals Δx and Δy, power density $g(x, y)$.
Result: Coefficients C_{pq}.

Calculate $f(x,y)=\frac{g(x_i,y_i)+m^2T_0}{\kappa}$;
for $j\leftarrow 1$ **to** M **do**
  $F[j][0]=\frac{1}{a}\sum_{i=0}^{N}f(x_i,y_j)\Delta x$;
end
for $p\leftarrow 1$ **to** $P-1$ **do**
  for $j\leftarrow 1$ **to** M **do**
    $F[j][p]=\frac{2}{a}\sum_{i=0}^{N}f(x_i,y_j)\cos(\frac{p\pi}{a}x_i)\Delta x$;
  end
end
for $p\leftarrow 0$ **to** $Q-1$ **do**
  $C_{p0}=\frac{1}{b}\sum_{j=0}^{M}\frac{F[j][p]}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}\Delta y$;
end
for $p\leftarrow 0$ **to** $P-1$ **do**
  for $q\leftarrow 1$ **to** $Q-1$ **do**
    $C_{pq}=\frac{2}{b}\sum_{j=0}^{M}\frac{F[j][p]\cos(\frac{q\pi}{b}y_j)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}\Delta y$;
  end
end

IV. Proposed PISOV Framework

The temperature distribution within the VLSI chip system is represented as a series of cosine functions, with the coefficients being initially unknown. While analytical solutions for these coefficients are efficient and accurate, they involve a 2-D integration operation, which can be time-consuming. Taking inspiration from the principles of PINN, we treat the coefficients as learning weights and incorporate the governing equation into the loss functions for coefficient training. To further enhance solving speed, we employ the generalized Moore–Penrose inverse algorithm to compute these coefficients. The comprehensive framework is referred to as PISOV, as illustrated in Fig. 1. Based on the cosine series (4), a novel neural network called the DCN is constructed, as shown in the left part of Fig. 1. With the automatic differentiation of the DCN, the derivatives $(\partial^2T/\partial x^2)$ and $(\partial^2T/\partial y^2)$ are obtained to form loss functions. Subsequently, a least-squared regression problem is solved using the generalized Moore–Penrose inverse algorithm to determine the coefficients $\boldsymbol{\beta}$.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

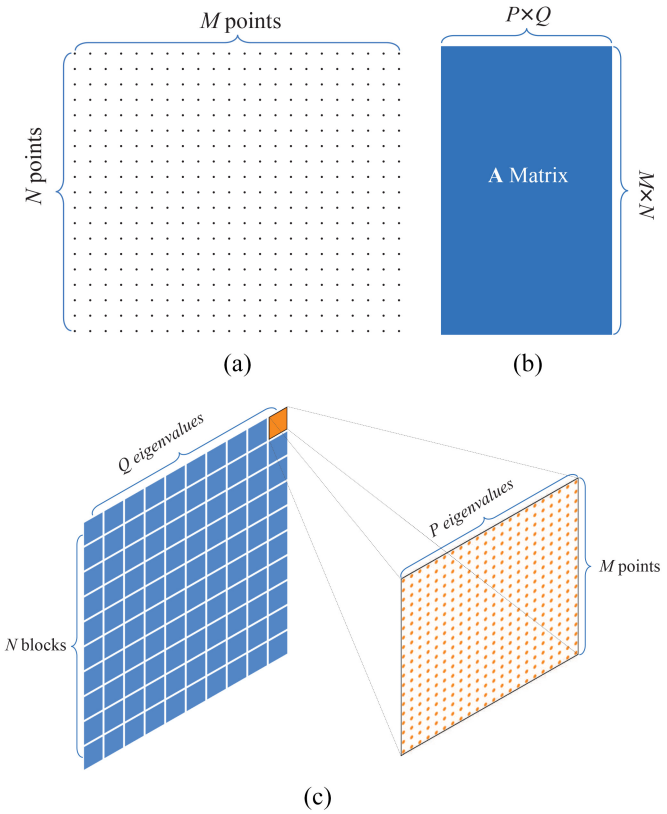


Fig. 2. (a) Grid distributions for $M \times N$ sampling points. (b) Matrix $\mathbf{A}$ with $MN \times PQ$ dimensions. (c) Two-layer hierarchical computational technique to prevent repeated cosine calculations.

Specifically, the cosine series (4) is introduced into the governing (3), allowing us to derive the loss functions, which form the fundamental concept behind PINN. The loss function is mathematically expressed as

$$\mathcal{L}_{\text{PISOV}} = \left| \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} C_{pq} \left[\frac{p^2\pi^2}{a^2} + \frac{q^2\pi^2}{b^2} + \frac{m^2}{\kappa} \right] \cdot \cos\left(\frac{p\pi}{a}x\right) \cos\left(\frac{q\pi}{b}y\right) - f(x,y) \right|^2. \tag{7}$$

Based on the loss functions, a least squares method is used to optimize the learning coefficients C_{pq}. The spatial sampling data is fed into the proposed PISOV framework, leading to the formation of linear matrix equations as follows:

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{b} \tag{8}$$

where $\mathbf{A}$ is an $L \times W$ matrix

$$A_{ij} = \left[\frac{p^2\pi^2}{a^2} + \frac{q^2\pi^2}{b^2} + \frac{m^2}{\kappa} \right] \cos\left(\frac{p\pi}{a}x_i\right) \cos\left(\frac{q\pi}{b}y_i\right) \tag{9}$$

where x_i and y_i are the coordinates of the ith sampling point, $j = p \times q$ is the jth coefficients C_{pq}, L is the number of sampling points, $W = P \times Q$ is the number of the coefficients, $\boldsymbol{\beta}$ is the coefficients to be solved

$$\beta_j = C_{pq} \tag{10}$$

and $\mathbf{b}$ is the power density related item

$$b_i = f(x_i, y_i). \tag{11}$$

As L is not equal to W, we employ the generalized Moore–Penrose inverse algorithm to address the least-square problem. First, we multiply both sides of (8) with the transpose of $\mathbf{A}$ matrix, which is expressed as

$$\mathbf{A}^T\mathbf{A}\boldsymbol{\beta} = \mathbf{A}^T\mathbf{b}. \tag{12}$$

After that, the coefficients can be computed as

$$\boldsymbol{\beta} = \left[\mathbf{A}^T\mathbf{A}\right]^{-1}\mathbf{A}^T\mathbf{b}. \tag{13}$$

The proposed PISOV framework efficiently calculates the coefficients. The majority of the algorithm's time is dedicated to constructing the matrix, as illustrated in Fig. 2(b). We assume that the sampling points form a grid distribution ($L = M \times N$), as shown in Fig. 2(a). We only need to consider M x-points and N y-points since other points can be represented by the combinations of M x-points and N y-points. Consequently, we introduce a two-hierarchical computational technique aiming at preventing redundant cosine calculations, as described in Fig. 2(c). First, we compute MP cosine computations and store them. Then, we perform NQ cosine operations using MP blocks. The number of $\cos(p\pi x/a)$ and $\cos(q\pi y/b)$ computations is MP and NQ, respectively. The total number of computations is $MP + NQ$, a significantly reduced figure compared to $MN \times PQ$. If the sampling points are not distributed in a grid pattern, the number of cosine computations becomes $MN \times PQ$, which is very time-consuming.

V. Parameterized PISOV

As we know, m is an effective convection coefficient, which varies with vertical heat dissipation. ML method can parameterize this coefficient m, which is a hyperdimensional problem. Parameterization of m can be used powerfully for facilitating fast UQ analysis. However, it is very difficult for traditional methods to solve hyperdimensional problems. To overcome this issue, the PISOV method is extended to solve hyperdimensional PDEs with parameterization of coefficient m, as depicted in Fig. 3. To consider the effective convection coefficient m, a neural network is employed as the basis function. With the automatic differentiation of the DCN and neural networks, the derivatives $(\partial^2 T/\partial x^2)$ and $(\partial^2 T/\partial y^2)$ are obtained to form loss functions. Subsequently, a least-squared regression problem is solved using the generalized Moore–Penrose inverse algorithm to determine the coefficients $\boldsymbol{\beta}$.

Specifically, the temperature distribution can be represented by

$$T(x,y,m) = \sum_{p=0}^{P} \sum_{q=0}^{Q} \sum_{h=0}^{H} C_{pqh} \text{NN}_h(m) \cdot \cos\left(\frac{p\pi}{a}x\right) \cos\left(\frac{q\pi}{b}y\right) \tag{14}$$

where $\text{NN}_h(m)$ is the hth output of the neural network $\text{NN}(\cdot)$. To utilize the efficient Moore-Penrose generalized inverse algorithm, it is necessary to fix the weights of $\text{NN}(\cdot)$, which are randomly generated within the range of $[-1, 1]$. C_{pqh} is

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

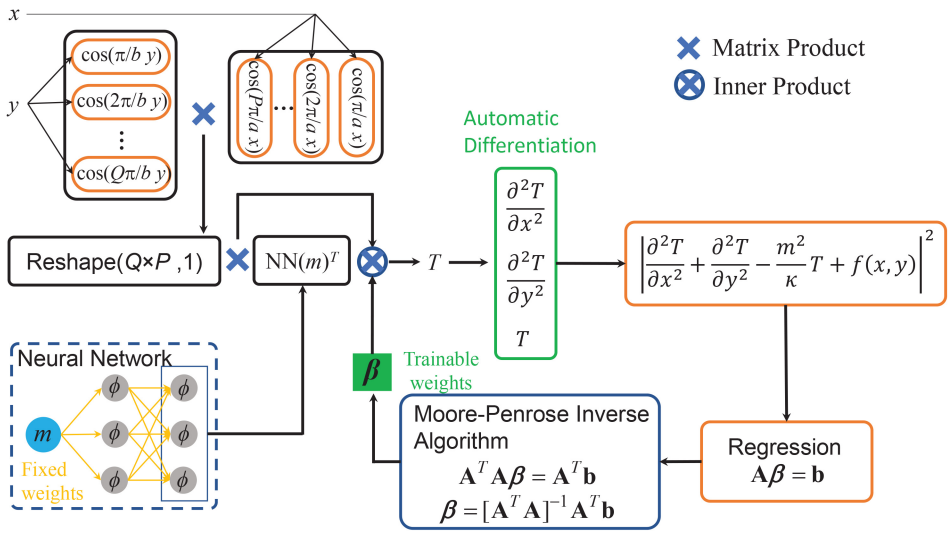


Fig. 3. Overall framework of the PISOV method with parameterization of coefficients m.

the learning parameter. By submitting (14) into the governing (3), we obtain the loss function

$$\mathcal{L}_{\text{P-PISOV}} = \Bigg| \sum_{p=0}^{P}\sum_{q=0}^{Q}\sum_{h=0}^{H} C_{pqh}\left[\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}\right]\cdot \text{NN}_h(m)\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right) - f(x,y)\Bigg|^2. \quad (15)$$

Based on the loss functions, a least squares method is employed to optimize the learning coefficients C_{pqh}. Specifically, L sampling points are fed into the proposed parameterized PISOV framework, resulting in the formation of linear matrix equations, which are then used to solve for the optimal coefficients

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{b} \quad (16)$$

where $\mathbf{A}$ is an $L \times W$ matrix

$$A_{ij} = \left[\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}\right]\text{NN}_h(m_i) \cdot \cos\left(\frac{p\pi}{a}x_i\right)\cos\left(\frac{q\pi}{b}y_i\right) \quad (17)$$

where x_i, y_i and m_i are the coordinates of the ith sampling point, $j = p \times q \times h$ is the jth coefficients C_{pqh}, $\text{NN}_h(m)$ is the hth output of neural network $\text{NN}(m)$, L is the number of sampling points, $W = P \times Q \times H$ is the number of the coefficients, $\boldsymbol{\beta}$ is the coefficients to be solved

$$\beta_j = C_{pqh} \quad (18)$$

and $\mathbf{b}$ is the power density related items

$$b_i = f(x_i, y_i). \quad (19)$$

Since L is not equal to W, the generalized Moore-Penrose inverse algorithm is used to solve the least square problem, which is the same as Section IV.

The parameterized PISOV framework can calculate the coefficients only once to obtain all results for parameter variations. Fig. 4(b) depicts the $\mathbf{A}$ matrix generated by the parameterized PISOV method. The number of coefficients $\boldsymbol{\beta}$ is $P \times Q \times H$. We assume that the sampling points form a grid distribution ($L = M \times N \times R$), as shown in Fig. 4(a). We only need to consider M x-points, N y-points, and R m-points as other points can be represented by the combinations of M x-points, N y-points, and R m-points. Therefore, we propose a three-layer computational skill to avoid redundant cosine and neural network calculations, as outlined in Fig. 4(c). First, we compute MP cosine computations and store them. Then, we perform NQ cosine operations using MP blocks to store them. Third, we obtain RH feedforward neural network operations using $MP \times NQ$ blocks to store them. The number of $\cos(p\pi x/a)$, $\cos(q\pi y/b)$ and $\text{NN}_h(m)$ computations is $M \times P$, $N \times Q$ and $R \times H$, respectively. The total number of function computations is $MP + NQ + RH$, which is much less than $MNR \times PQH$. In the event that the sampling points are not distributed in a grid pattern, the number of function computations increases to $MNR \times PQH$, which is time-consuming.

Although this method specifically addresses three variables (x, y, and m), it can be readily extended to accommodate an arbitrary number of variables by employing neural networks. When additional parameters are introduced, the input size of the neural network increases accordingly, leading to an increase in the sampling dimension as well. For instance, we can utilize a neural network $\text{NN}(P1, P2, P3, P4)$ to simultaneously account for power values for four cores in the chip which is demonstrated in Section VI-B, and temperature is represented by

$$T(x, y, P1, P2, P3, P4) = \sum_{p=0}^{P}\sum_{q=0}^{Q}\sum_{h=0}^{H} C_{pqh} \cdot \text{NN}_h(P1, P2, P3, P4) \cdot \cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right) \quad (20)$$

where $P1$, $P2$, $P3$, and $P4$ are the power values of four cores, respectively.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

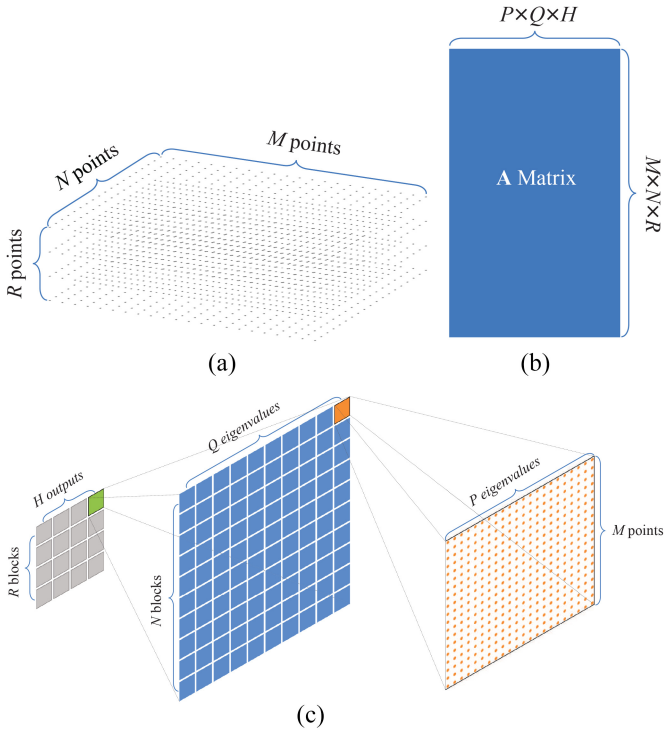


Fig. 4. (a) Grid distributions for $M \times N \times R$ sampling points. (b) Matrix **A** with $MNR \times PQH$ dimensions. (c) Three-layer hierarchical computational technique to prevent repeated cosine calculations.

VI. Numerical Results and Discussion

In this section, we demonstrate the accuracy and efficiency of the proposed PISOV method through comparisons with ThermPINN, and PINN techniques. First, we assess the performance of PISOV without parameterization on the Alpha21264 microprocessor, using seven benchmarks randomly generated with varying complexities, as well as six real-world industrial cases. Second, using the same set of benchmarks, we parameterize the effective convection coefficient m and power values of four cores to investigate PISOV with parameterization. Lastly, we explore the impact of six different sampling methods (Grid, Uniform, Sobol, LHS, Halton, and Hammersley sampling methods) and hybrid sampling methods to further enhance the PISOV framework.

The PISOV method without parameterization is implemented in the Python language with the Numpy package. The parameterized PISOV, parameterized ThermPINN, parameterized PINN, ThermPINN, and PINN methods are developed using the PyTorch platform. To obtain ground truth results for all benchmarks, COMSOL commercial software is employed. All the programs are executed on a Linux server equipped with an Intel Xeon W9-3495X 1.9 GHz CPU, two NVIDIA RTX A6000 48G GPUs and two NVIDIA RTX 4090 24G GPUs.

A. PISOV Without Parameterization

We conduct testing on the alpha21264 processor, seven randomly generated floorplans and six real-world industrial cases to evaluate the proposed PISOV method without parameterization by using grid sampling points. In the PISOV approach, the number of sampling points and eigenvalues significantly impact accuracy. To illustrate this, we employ the Alpha21264 processor as an example and perform PISOV with sampling points ranging from 50×50 to 600×600 and eigenvalues from 10×10 to 70×70. Accuracy is quantified using the mean absolute error (MAE) and maximum absolute error (MAX). The results are presented in Tables I and II. Table II highlights the effect of eigenvalues on the accuracy of PISOV and SOV methods when using 200×200 sampling points. It becomes evident that 50×50 eigenvalues are sufficient for PISOV to achieve high accuracy. Here, 200×200 sampling points are capable of achieving good accuracy. With an increase in the number of sampling points, accuracy improves, and the speedup of PISOV is reduced.

We perform a comparison of the proposed PISOV, ThermPINN, PINN, and COMSOL by using seven randomly generated benchmarks and six real-world industrial cases

TABLE I
Accuracy and Speed Comparison for alpha21264 Processor With 200×200 Sampling Points

Eigenvalues	PISOV		
	MAE	MAX	Time (s)
10×10	0.55	2.67	0.11
20×20	0.19	1.02	0.19
30×30	0.14	0.62	0.35
40×40	0.14	0.56	0.63
50×50	0.14	0.59	0.85
60×60	0.14	0.57	1.52
70×70	0.14	0.59	1.98

TABLE II
Accuracy and Speed Comparison for alpha21264 Processor With 50×50 Eigenvalues

Sampling points	PISOV		
	MAE	MAX	Time (s)
50×50	0.51	1.71	0.28
100×100	0.34	1.23	0.44
200×200	0.14	0.59	0.85
300×300	0.11	0.52	1.73
400×400	0.053	0.30	3.22
500×500	0.065	0.37	4.65
600×600	0.061	0.30	6.26

TABLE III
Accuracy and Speed Comparison of PISOV for Several Benchmarks With 50×50 Eigenvalues and 200×200 Sampling Points, COMSOL With 40020 Elements Mesh

	Units	PISOV		ThermPINN [23]		PINN [34]		COMSOL
		MAE (K)	Time (s)	MAE (K)	Time (s)	MAE (K)	Time (s)	Time (s)
Design1	14	0.015	**0.94**	0.020	249.27	0.03	10426	1.55
Design2	66	0.15	**1.04**	0.084	245.37	0.14	10422	1.67
Design3	132	0.14	**0.86**	0.093	245.73	0.16	10420	1.66
Design4	229	0.25	**0.98**	0.16	245.03	0.18	10423	1.54
Design5	350	0.12	**0.95**	0.19	245.37	0.15	10433	1.63
Design6	512	0.18	**1.03**	0.15	246.42	0.15	10396	1.59
Design7	700	0.29	**0.98**	0.15	252.16	0.20	10438	1.54
Sparc95	16384	0.019	**0.78**	0.32	225.86	0.31	6348	1.69
Swerv95	16384	0.39	**0.88**	0.71	303.14	0.77	6235	1.58
Black95	16384	0.079	**0.81**	0.32	226.99	0.33	6054	1.51
Pico95	16384	0.007	**0.91**	0.27	303.48	0.35	6279	1.73
Scc	517	0.053	**0.84**	0.18	152.21	0.026	3374	1.63
Intel-i7	27	0.016	**0.80**	0.085	151.37	0.034	3499	1.71

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

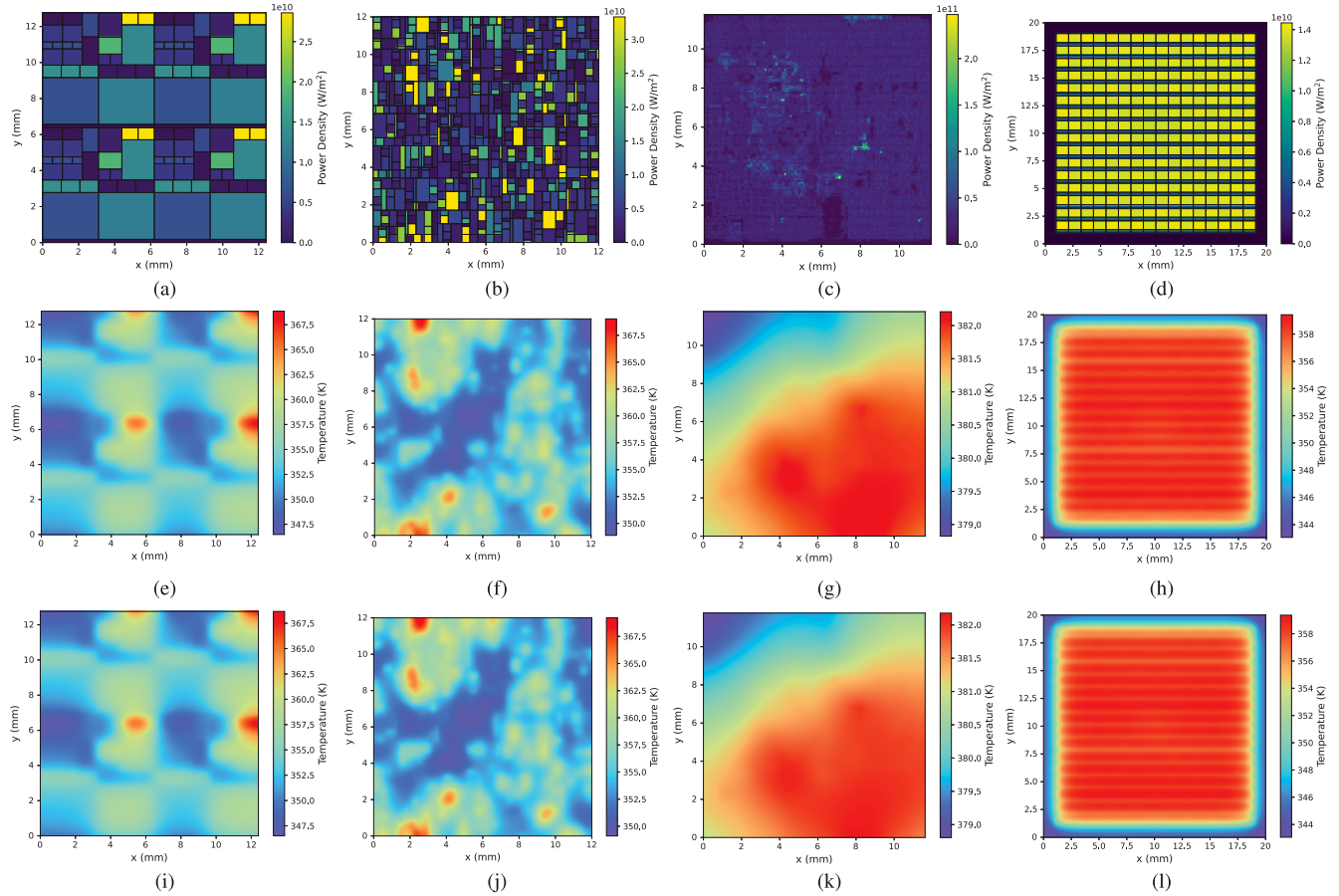


Fig. 5. (a) Alpha21264 microprocessor floorplan. (b) Design 7. (c) Sparc circuits from OpenROAD. (d) SCC-based Chip. (e)–(h) Thermal maps obtained by the PISOV method. (i)–(l) Thermal maps obtained by COMSOL.

obtained from OpenROAD, SCC-based chip, and Inteli7 CPU, as summarized in Table III. The real power maps can be accessed at https://github.com/peaclab/PACT. The PINN is configured as a FCN architecture. Without the parameterization of m, the FCN consists of six layers with the following neuron configuration: [2, 256, 256, 256, 256, 256, 1]. In contrast, the ThermPINN employs a DCN with 900 neurons. In COMSOL, the model has been configured with a mesh consisting of 40 020 elements. The number of sampling points and eigenvalues are set to 200×200 and 50×50, respectively. Across different designs, they exhibited similar accuracy and running times, dependent on the number of sampling points and eigenvalues, rather than the number of functional units in the VLSI chip. For a given level of accuracy, PISOV achieves speedups of $245\times$, $10^4\times$ and $1.6\times$ over ThermPINN, PINN, and COMSOL, respectively. From a PINN perspective, PISOV stands out by offering the fastest speed coupled with good accuracy when compared to ThermPINN and traditional PINNs. This improvement is primarily attributed to the utilization of the Moore–Penrose generalized inverse algorithm and a hierarchical computational technique. The Moore–Penrose generalized inverse algorithm excels in training learning parameters through matrix solving, a process that is significantly faster than the stochastic gradient descent algorithm commonly used in PINNs. Additionally, the hierarchical computational technique employed in PISOV helps to avoid redundant computations of cosine functions by leveraging a lookup table method, further enhancing computational efficiency.

To provide a clear visualization of the results, we present thermal maps generated by PISOV and COMSOL in Fig. 5. For all benchmarks, PISOV has a good agreement with COMSOL. Even as the floorplans become increasingly complex, PISOV maintains high accuracy, as evident in Fig. 5(g) and (k).

B. PISOV With Parameterization

To demonstrate the effectiveness of the parameterization technique in PISOV, we apply the parameterized PISOV to the same set of benchmarks, as detailed in Tables IV and V. Several parameters require determination in PISOV, such as the number of eigenvalues, the number of sampling points in the (x, y) and m domains, and the number of outputs for neural networks. To identify the optimal parameters, we use the Alpha21264 processor, and the results are summarized in Table IV. The parameterization of effective convection coefficient m ranges from 1500 to 1600. The accuracy significantly improved with increasing values of P, Q, M, and N, while accuracy remained constant with increasing R and H, suggesting that small values of R and H were sufficient for achieving the desired accuracy. The optimal parameters,

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
ACCURACY AND SPEED COMPARISON FOR ALPHA21264 PROCESSOR WITH DIFFERENT PARAMETERS IN THE PARAMETERIZED PISOV

R=4,P=Q=30, H=3	M=N=20		M=N=60		M=N=100		M=N=140		M=N=180	
	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)
MAE (K)	4.71e30	2.04e30	0.33	0.33	0.27	0.27	0.16	0.16	0.15	0.15
MAX (K)	1.16e31	5.05e30	1.35	1.35	1.30	1.30	1.16	1.16	0.82	0.82
Time (s)	**0.15**	0.44	**0.38**	1.18	**1.10**	2.63	**2.20**	4.27	**3.76**	6.42
M=N=100, P=Q=30, H=3	R=2		R=3		R=4		R=5		R=6	
	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)
MAE (K)	0.46	0.36	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
MAX (K)	2.77	1.79	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
Time (s)	**0.60**	1.53	**0.87**	1.97	**1.10**	2.69	**1.40**	3.12	**1.56**	3.53
M=N=100, R=4, H=3	P=Q=10		P=Q=20		P=Q=30		P=Q=40		P=Q=50	
	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)
MAE (K)	0.70	0.70	0.30	0.30	0.27	0.27	0.27	0.27	0.27	0.27
MAX (K)	2.65	2.65	1.61	1.61	1.25	1.30	1.21	1.21	1.23	1.23
Time (s)	**0.10**	0.66	**0.40**	1.31	**1.10**	2.62	**2.38**	4.74	**4.29**	8.73
M=N=100, R=4,P=Q=30	H=2		H=3		H=4		H=5		H=6	
	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)	PISOV (Optimized)	PISOV (Normal)
MAE (K)	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
MAX (K)	1.29	1.29	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
Time (s)	**0.69**	2.33	**1.10**	2.43	**1.55**	2.66	**1.94**	2.85	**2.52**	3.24

$M = N = 100, R = 4, P = Q = 30$, and $H = 3$, strike a balance between accuracy and speed. In addition, we introduced a fast matrix assembly method based on grid sampling to avoid redundant computation of cosine functions, which is also validated in Table IV. The optimized PISOV represents our proposed hierarchical computational technique, while the regular PISOV does not include the fast matrix assembly technique. The optimized PISOV achieves a 2× speedup over the regular PISOV. However, SOV is unable to parameterize the effective convection coefficient and requires multiple runs to obtain the parameterized results (m with the range of 1500–1600).

To further emphasize the accuracy and efficiency of the parameterized PISOV, we conducted a comparison with parameterized ThermPINN and the parameterized PINN method using seven randomly generated benchmarks, as described in Table V. When the parameterization of m is applied, the FCN maintains the same six-layer structure with [2, 256, 256, 256, 256, 256, 1] neurons. Additionally, the ThermPINN model is extended to include both a DCN with 900 neurons and a fully connected network with a configuration of [1, 100, 200, 300, 600, 900] neurons. In COMSOL, the model has been configured with a mesh consisting of elements. As observed, the parameterized PISOV produces results for m values in the range of 1500-1600 within just 3.5 s, which is exceptionally fast compared to COMSOL, which requires 176 s to calculate 200 times ($\Delta m = 0.5$). The parameterized PISOV method achieves a remarkable speedup of 230× and 5000× over parameterized ThermPINN and parameterized PINN. Similar to PISOV, the parameterized PISOV exhibited similar accuracy and running times across different designs.

TABLE V
ACCURACY AND SPEED COMPARISON FOR SEVEN RANDOMLY GENERATED BENCHMARKS WITH $M = N = 100$, $R = 4$, $P = Q = 30$, AND $H = 3$, COMSOL WITH 10546 ELEMENTS MESH AND Δm=0.5

	Units	Parameterized PISOV		Parameterized ThermPINN [23]		Parameterized PINN [34]		COMSOL
		MAE (K)	Time (s)	MAE (K)	Time (s)	MAE (K)	Time (s)	Time (s)
Design1	14	0.019	**3.46**	0.021	531	0.025	19735	177
Design2	66	0.15	**3.55**	0.15	735	0.13	19738	180
Design3	132	0.16	**3.41**	0.16	841	0.16	19732	179
Design4	229	0.23	**3.55**	0.23	823	0.23	19744	181
Design5	350	0.11	**3.50**	0.11	525	0.08	19737	182
Design6	512	0.19	**3.56**	0.19	739	0.17	19752	177
Design7	700	0.29	**3.57**	0.29	831	0.30	19742	175

To further validate the effectiveness of the parameterization technique, we applied it to a more complex scenario by parameterizing the power values (P1, P2, P3, and P4) of four cores, each ranging from 0 to 40 W, as depicted in Fig. 6(a). In the Parameterized PISOV method, we set the parameters M=N=70, R=5, P=Q=30, and H=45. The training process took 2333 s, enabling us to obtain results for any combination of

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

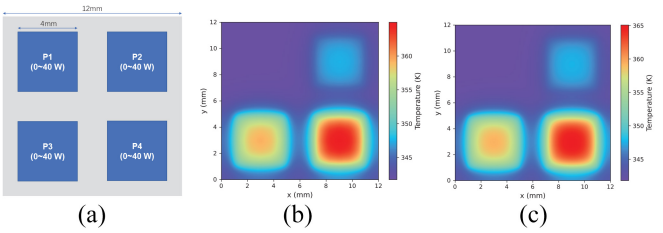


Fig. 6. (a) Power parameterization of four cores for a chip with the range of 0–40 W. Thermal maps obtained by (b) Parameterized PISOV method and (c) COMSOL with the P1=0, P2=10, P3=30, and P4=40.

TABLE VI
ACCURACY AND SPEED COMPARISON FOR SEVERAL POWER MAPS WITH $M = N = 70$, $R = 5$, $P = Q = 30$, AND $H = 45$, COMSOL WITH 10546 ELEMENTS MESH AND ΔP=1 W

Power Sampling (W)	Parameterized PISOV			COMSOL
	MAE (K)	MAX (K)	Time (s)	Time (s)
P1 = 10, P2 = 20, P3 = 30, P4 = 40	0.43	0.99	2333	9.26×10^7
P1 = 0, P2 = 10, P3 = 30, P4 = 40	0.23	0.85		
P1 = 10, P2 = 10, P3 = 40, P4 = 40	0.13	0.66		
P1 = 30, P2 = 30, P3 = 30, P4 = 30	0.26	0.61		
P1 = 30, P2 = 30, P3 = 30, P4 = 40	0.34	0.87		
P1 = 30, P2 = 30, P3 = 40, P4 = 40	0.12	0.62		
P1 = 30, P2 = 33, P3 = 37, P4 = 40	0.22	0.73		
P1 = 30, P2 = 40, P3 = 40, P4 = 40	0.15	0.61		
P1 = 40, P2 = 40, P3 = 40, P4 = 40	0.42	0.90		

P1, P2, P3, and P4 within the specified range. Subsequently, we used COMSOL to perform a parameter sweep with a step size of $\Delta P = 1W$, resulting in 2 560 000 evaluations. The time consumption for this process, using a mesh with 10 546 elements, was 9.26×10^7 s, significantly slower than the PISOV method. Table VI presents some of our findings, demonstrating an MAE of less than 0.5 K, indicating high accuracy. For a visual comparison, we provide thermal maps for a specific power configuration (P1 = 0, P2 = 10, P3 = 30, P4 = 40) obtained using both the PISOV and COMSOL methods, as shown in Fig. 6(b) and (c), respectively. These figures intuitively illustrate the close agreement between the two methods.

C. Impact of Sampling Methods

Fig. 7 illustrates six different sampling methods, including Grid, Uniform, Sobol, LHS, Halton, and Hammersley distributions. We employ the Alpha21264 processor with both PISOV and parameterized PISOV to investigate the impact of these sampling methods on accuracy. The results are presented in Table VII, indicating varying levels of sampling efficiency that directly affect accuracy. Among the sampling methods, Hammersley sampling demonstrates the highest accuracy. The Sobol and Halton sampling methods demonstrate impressive accuracy. Conversely, the grid sampling method exhibits the highest error among the tested methods. LHS and uniform sampling methods are influenced by random, their errors are larger than Sobol and Halton. Furthermore, it is evident that the Grid sampling method offers the fastest computational speed compared to the others. For grid sampling points, the computational effort is significantly reduced, as it only requires

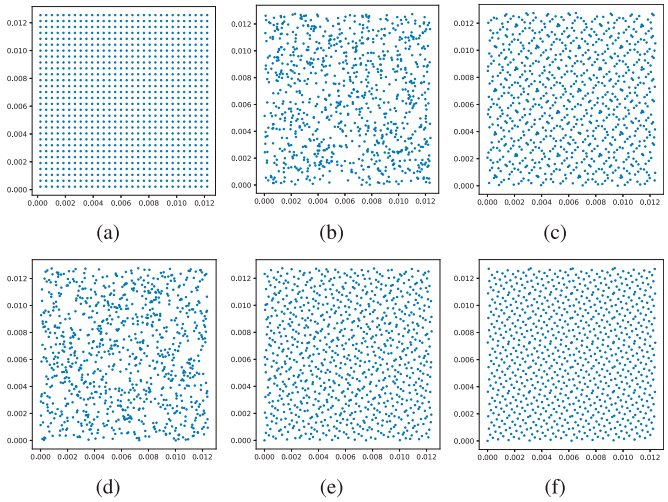


Fig. 7. (a) Grid sampling method. (b) Sampling method with uniform distribution. (c) Sobol sampling method. (d) LHS sampling method. (e) Halton sampling method. (f) Hammersly sampling method.

TABLE VII
ACCURACY AND SPEED COMPARISON FOR ALPHA21264 PROCESSOR WITH DIFFERENT SAMPLING METHODS WITH SAME NUMBER OF POINTS

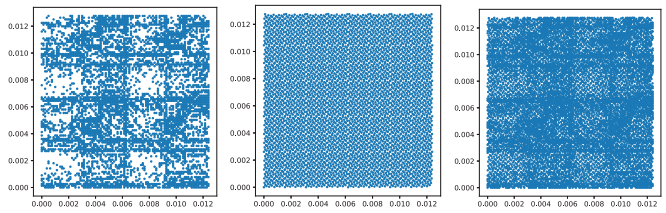
		Grid	Uniform	Sobol	LHS	Halton	Hammersly
PISOV	MAE (K)	0.32	0.16	0.088	0.19	0.093	**0.066**
	MAX (K)	1.34	0.99	0.73	1.60	0.68	**0.51**
	Time (s)	**0.13**	0.73	0.83	0.72	0.59	0.62
Parameterized PISOV	MAE (K)	0.27	0.22	0.11	0.21	0.11	**0.08**
	MAX (K)	1.30	2.12	1.07	1.77	1.05	**0.92**
	Time (s)	**1.07**	3.39	3.53	3.65	2.76	3.05

$MP+NQ$ cosine operations, thereby avoiding redundant cosine calculations compared to the other sampling methods.

To further illustrate the performance of Harmmersley sampling, we investigate the efficacy of hybrid sampling methods. Specifically, we combine different sampling techniques. For instance, we utilize a blend of 5000 Harmmersley sampling points and 5000 Halton sampling points as an alternative to using 10 000 Harmmersley sampling points alone. The results of this comparison are presented in Table VIII. Incorporating Harmmersley sampling points with Grid, uniform, Sobol, and LHS points enhances the overall accuracy. However, combining Halton and Sobol sampling methods yields inferior results compared to using Sobol or Halton sampling points individually.

Next, we examine the impact of adaptive sampling methods on accuracy. Adaptive sampling points are determined based on the residual error of the equation; generally, more sampling points can lead to a larger residual error. Initially, we employ 10 000 Hammersley sampling points to compute the results. Subsequently, based on these results, we estimate 5000 adaptive sampling points using the equation residual error, as depicted in Fig. 8(a). To ensure solver stability, we retain 5000 Hammersley sampling points, as illustrated in Fig. 8(b). Finally, we combine both sets of sampling points, as shown in Fig. 8(c). However, the accuracy is not improved compared with Harmmersley sampling methods. Based on our observations, the Hammersley sampling method is highly suitable for the proposed PISOV framework, as demonstrated in Table VIII.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 8. (a) Adaptive sampling points based on equation residuals. (b) Hammersly sampling points. (c) Hybrid sampling points from (a) and (b).

TABLE VIII
ACCURACY COMPARISON FOR ALPHA21264 PROCESSOR WITH HYBRID SAMPLING METHODS WITH SAME NUMBER OF POINTS

		Grid+ Hammersly	Uniform+ Hammersly	Sobol+ Hammersly	LHS+ Hammersly	Halton+ Hammersly	Sobol+ Halton	Adaptive+ Hammersly
PISOV	MAE (K)	0.17	0.14	0.085	0.14	0.10	0.12	0.27
	MAX (K)	1.16	1.15	1.13	0.85	0.74	1.32	0.89

VII. CONCLUSION

This article proposed a novel PISOV for fast full-chip thermal analysis. We employed the Moore–Penrose generalized inverse algorithm instead of the back-propagation method to speedup ThermPINN. Additionally, we implemented a hierarchical matrix assembly technique to accelerate the training process of ThermPINN. The proposed PISOV, which combines principles from PINN and SOV, outperforms ThermPINN, PINN, and COMSOL in terms of computational speed. Furthermore, we extended PISOV to parameterize the effective convection coefficient m and power values of four cores using neural networks, a capability beyond the reach of SOV. The parameterized PISOV calculates the coefficients just once, providing all parameterized results for hyperdimensional PDEs. We also investigated the impact of various sampling methods, including grid, uniform, Sobol, LHS, Halton, Hammersley, and hybrid sampling methods on the accuracy of PISOV and parameterized PISOV. The numerical results demonstrate that PISOV achieves remarkable speed improvements of 245×, and $10^4\times$ compared to ThermPINN, and PINN, respectively. Additionally, the parameterized PISOV exhibits substantial speed enhancements of 230× and 5000× over parameterized ThermPINN and parameterized PINN, respectively. Among the various sampling methods studied, Hammersley sampling exhibits the highest accuracy. Notably, the Grid method offers the highest computational efficiency by avoiding redundant cosine calculations.

APPENDIX

Under the framework of the SOVs method, the temperature distribution is initially represented as a product of two separate functions as shown in

$$T(x,y) = X(x)Y(y). \tag{21}$$

By substituting (21) into (3), a PDE is obtained as follows:

$$\frac{\partial^2 XY}{\partial x^2} + \frac{\partial^2 XY}{\partial y^2} - \frac{m^2}{\kappa}XY = -f(x,y),$$
$$\text{BC: } \left.\frac{\partial XY}{\partial x}\right|_{x=0,a} = \left.\frac{\partial XY}{\partial y}\right|_{y=0,b} = 0 \tag{22}$$

where

$$f(x,y) = \frac{g(x,y) + m^2 T_0}{\kappa}. \tag{23}$$

We reorganize the formulas (22) as

$$-\lambda_x - \lambda_y - \frac{m^2}{\kappa} = -f(x,y)/(XY)$$
$$\frac{\partial^2 X}{\partial x^2}/X = -\lambda_x, \quad \frac{\partial^2 Y}{\partial y^2}/Y = -\lambda_y$$
$$\text{BC: } \left.\frac{\partial X}{\partial x}\right|_{x=0,a} = \left.\frac{\partial Y}{\partial y}\right|_{y=0,b} = 0. \tag{24}$$

This PDE (24) is separated into two ordinary differential equations (ODEs) using the SOV method. Subsequently, boundary conditions are applied to determine the eigenvalues and eigenfunctions of the two ODEs

$$\lambda_x = \frac{p\pi}{a}, \quad X_p \sim \cos\left(\frac{p\pi}{a}x\right)$$
$$\lambda_y = \frac{q\pi}{b}, \quad Y_q \sim \cos\left(\frac{q\pi}{b}y\right). \tag{25}$$

Therefore, based on (21) and (25), the temperature can be expressed as

$$T(x,y) = \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} C_{pq}\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right). \tag{26}$$

By submitting (26) to (3), we obtain

$$\sum_{p=0}^{\infty}\sum_{q=0}^{\infty} C_{pq}\left[\frac{p^2\pi^2}{a^2} + \frac{q^2\pi^2}{b^2} + \frac{m^2}{\kappa}\right]$$
$$\cdot\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right) = f(x,y). \tag{27}$$

We multiply both sides of (27) by $\cos((p\pi/a)x)\cos((q\pi/b)y)$, and perform double integration, resulting in

$$\int_0^b\int_0^a \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} C_{pq}\left[\frac{p^2\pi^2}{a^2} + \frac{q^2\pi^2}{b^2} + \frac{m^2}{\kappa}\right]$$
$$\cdot\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)dxdy$$
$$= \int_0^b\int_0^a f(x,y)\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)dxdy. \tag{28}$$

Based on the orthogonality of basis functions, we can determine the coefficients C_{pq} as expressed in

$$C_{pq} = \begin{cases} \frac{1}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy, \\ (p=0 \text{ and } q=0); \\ \frac{2}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy, \\ (p>0 \text{ and } q=0 \;||\; p=0 \text{ and } q>0); \\ \frac{4}{ab}\int_0^b\int_0^a f(x,y)\frac{\cos\left(\frac{p\pi}{a}x\right)\cos\left(\frac{q\pi}{b}y\right)}{\frac{p^2\pi^2}{a^2}+\frac{q^2\pi^2}{b^2}+\frac{m^2}{\kappa}}dxdy, \\ (p>0 \text{ and } q>0). \end{cases} \tag{29}$$

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] S. X.-D. Tan, M. Tahoori, T. Kim, S. Wang, Z. Sun, and S. Kiamehr, *Long-Term Reliability of Nanometer VLSI Systems: Modeling, Analysis and Optimization*. Cham, Switzerland: Springer Publ., 2019.

[2] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 349–360, Feb. 2014.

[3] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, "Task migrations for distributed thermal management considering transient effects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 397–401, Feb. 2015.

[4] H. Wang et al., "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, no. 1, pp. 1–21, Aug. 2016.

[5] J. Zhang, S. Sadiqbatcha, and S. X.-D. Tan, "Hot-trim: Thermal and reliability management for commercial multicore processors considering workload dependent hot spots," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 7, pp. 2290–2302, Jul. 2023.

[6] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, 2003, pp. 86–89.

[7] C.-H. Tsai and S.-M. Kang "Cell-level placement for improving substrate thermal distribution," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 2, pp. 253–266, Feb. 2000.

[8] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "Efficient full-chip thermal modeling and analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2004, pp. 319–326.

[9] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[10] J. Xie and M. Swaminathan, "Electrical–thermal cosimulation with nonconformal domain decomposition method for multiscale 3-D integrated systems," *IEEE Trans. Compon., Packag. Manuf. Technol.*, vol. 4, no. 4, pp. 588–601, Apr. 2014.

[11] P.-Y. Huang and Y.-M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 5, pp. 613–626, May 2009.

[12] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–1675, Sep. 2007.

[13] B. Wang and P. Mazumder, "Accelerated chip-level thermal analysis using multilayer Green's function," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 325–344, Feb. 2007.

[14] H. Sultan and S. R. Sarangi, "VarSim: A fast and accurate variability and leakage aware thermal simulator," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, 2020, pp. 1–6.

[15] N. Soleimani, P. Manfredi, and R. Trinchero, "Efficient implementation of the vector-valued kernel ridge regression for the uncertainty quantification of the scattering parameters of a 2-GHz low-noise amplifier," in *Proc. IEEE MTT-S Int. Conf. Numer. Electromagn. Multiphysics Model. Optim. (NEMO)*, 2023, pp. 143–146.

[16] N. Metropolis and S. Ulam, "The monte carlo method," *J. Amer. Stat. Assoc.*, vol. 44, no. 247, pp. 335–341, 1949.

[17] Z. Li et al., "Fourier neural operator for parametric partial differential equations," 2021, *arXiv:2010.08895*.

[18] J. Wen et al., "DNN-based fast static on-chip thermal solver," in *Proc. 36th Semicond. Therm. Meas., Model. Manage. Symp. (SEMI-THERM)*, 2020, pp. 65–75.

[19] L. Chen, W. Jin, and S. X.-D. Tan, "Fast thermal analysis for chiplet design based on graph convolution networks," in *Proc. 27th Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2022, pp. 485–492.

[20] 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. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[21] 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.

[22] Z. Liu et al., "DeepOHeat: Operator learning-based ultra-fast thermal simulation in 3D-IC design," in *Proc. 60th ACM/IEEE Design Autom. Conf. (DAC)*, 2023, pp. 1–6.

[23] L. Chen, J. Lu, W. Jin, and S. X.-D. Tan, "Fast full-chip parametric thermal analysis based on enhanced physics enforced neural network," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2023, pp. 1–8.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[25] 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.

[26] 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. Design Autom. Conf. (ASP-DAC)*, 2020, pp. 229–234.

[27] 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. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2020, pp. 1–9.

[28] 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. 26th Asia South Pac. Design Autom. Conf. (ASPDAC)*, 2021, pp. 690–696.

[29] J. Lu, J. Zhang, and S. X.-D. Tan, "Real-time thermal map estimation for AMD multi-core cpus using transformer," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2023, pp. 1–7.

[30] 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.

[31] 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 Design*, 2020, pp. 1–9.

[32] 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.

[33] W. Jin, S. Peng, and S. X.-D. Tan, "Data-driven electrostatics analysis based on physics-constrained deep learning," in *Proc. Design, Autom. Test Europe Conf. (DATE)*, 2021, pp. 1382–1387.

[34] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *J. Heat Transf.*, vol. 143, no. 6, Apr. 2021, Art. no. 60801.

[35] O. Hennigh et al., "NVIDIA SimNet^{TM}: An AI-accelerated multi-physics simulation framework," in *Proc. 21st Int. Conf. Comput. Sci. (ICCS)*, 2021, pp. 447–461.

[36] 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.

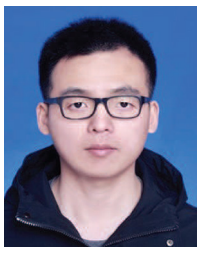
[37] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-informed neural networks with hard constraints for inverse design," *SIAM J. Sci. Comput.*, vol. 43, no. 6, pp. B1105–B1132, 2021.

[38] E. Schiassi, R. Furfaro, C. Leake, M. De Florio, H. Johnston, and D. Mortari, "Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations," *Neurocomputing*, vol. 457, pp. 334–356, Oct. 2021.

[39] V. Dwivedi and B. Srinivasan, "Physics informed extreme learning machine (PIELM)–A rapid method for the numerical solution of partial differential equations," *Neurocomputing*, vol. 391, pp. 96–118, May 2020.

[40] C. Xu, S. K. Kolluri, K. Endo, and K. Banerjee, "Analytical thermal model for self-heating in advanced FinFET devices with implications for design and reliability," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 7, pp. 1045–1058, Jul. 2013.

[41] D. Sarkar, A. Haji-Sheikh, and A. Jain, "Temperature distribution in multi-layer skin tissue in presence of a tumor," *Int. J. Heat Mass Transf.*, vol. 91, pp. 602–610, Dec. 2015.



Liang Chen (Member, IEEE) 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 University of California at Riverside, Riverside, CA, USA. Since 2023, he has been an Associate Professor with the School of Microelectronics, Shanghai University, Shanghai, China. His current research interests include electronic design automation and machine learning for VLSI reliability analysis.

Dr. Chen has been an Associate Editor for *Integration* (Elsevier).

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.

Wenxing Zhu received the Ph.D. degree from Shanghai University, Shanghai, China, in 1996.

He is a "Jiaxi" Distinguished Professor with Fuzhou University, Fuzhou, China. His research interests include VLSI physical design, optimization theory, and algorithms.

Dr. Zhu has received several prestigious awards, including the First Prize of Natural Sciences from the Ministry of Education of China in 2022 and the Application Award of Operations Research from the Operations Research Society of China in 2020. He received Best Paper Awards from DAC 2017 and ISEDA 2023, and was a Best Paper Award nominee at ICCAD 2018 and CCF-DAC 2023. Additionally, he was honored with the Second Prize for National Teaching Achievement in 2009.

Min Tang (Senior Member, IEEE) received the B.S. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 2001, the M.S. degree in electrical engineering from Xi'an Jiao Tong University, Xi'an, China, in 2004, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007.

Since 2007, he has been a Faculty Member of Shanghai Jiao Tong University, where he is currently a Professor. His research interest is multiphysics modeling of integrated systems.

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 is also a Cooperative Faculty Member with the Department of Computer Science and Engineering. He has published more than 330 technical papers and has co-authored six books on those areas. 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.

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 was a Visiting Professor of Kyoto University as a JSPS Fellow from December 2017 to January 2018. He is serving as the TPC Chair for ASPDAC 2021, and the TPC Vice Chair for ASPDAC 2020. He is serving or served as Editor-in-Chief for *Integration* (Elsevier), the an Associate Editor for four journals, including IEEE TRANSACTION ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, *ACM Transaction on Design Automation of Electronic Systems*, *Microelectronics Reliability* (Elsevier), and *Electronics* (MDPI), *Microelectronics*, and *Optoelectronics Section*.

Jun-Fa Mao (Fellow, IEEE) was born in 1965. He received the B.S. degree in radiation physics from the National University of Defense Technology, Changsha, China, in 1985, the M.S. degree in experimental nuclear physics from the Shanghai Institute of Nuclear Research, Chinese Academy of Sciences, Beijing, China, in 1988, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992.

From July 1992 to January 2022, he was a Faculty Member with Shanghai Jiao Tong University. He was the Dean of the School of Electronic Information and Electrical Engineering and the Vice President of Shanghai Jiao Tong University. He was a Visiting Scholar with the Chinese University of Hong Kong, Hong Kong, from 1994 to 1995, and a Postdoctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA, from 1995 to 1996. Since 2022, he has been the President and a Chair Professor of Shenzhen University, Shenzhen, China. He has authored or co-authored 600 papers (including 200 IEEE journal papers). His research interests include the interconnect, components and package problem of high speed, and radio frequency integrated circuits and systems.

Dr. Mao was a recipient of the National Natural Science Award of China in 2004, the National Technology Invention Award of China in 2008, the National Science and Technology Advancement Award of China in 2012 and 2023, the National Award of Teaching in 2005 and 2018, the Top Ten Scientific and Technological Progress of Chinese Universities in 2020, and many best paper awards of international conferences. He is an Academician of the Chinese Academy of Science. He was the Founder and 2007–2009 Chair of the IEEE Shanghai Section, the 2009–2018 Chair of IEEE MTT-S Shanghai Chapter, a member of the 2012–2014 IEEE Microwave Theory and Techniques Society Fellow Evaluation Committee and the 2015 IEEE Fellow Committee. He is a member of the State Council Academic Degrees Committee, a Fellow and the Vice Chairman of Chinese Institute of Electronics (CIE), and the Director of the Microwave Society of CIE.

Jianhua Zhang received the Ph.D. degree in mechanical engineering from Shanghai University, Shanghai, China, in 1999.

She is currently a Professor with the School of Microelectronics, Shanghai University, Shanghai, where she serves as the Director of the Key Laboratory of Advanced Display and System Applications, Ministry of Education. Her research interests include new optoelectronic materials and device technologies, display chip design and system integration, digital video and stereoscopic display technologies, and AMOLED device engineering for future display applications.

Authorized licensed use limited to: Univ of Calif Riverside. Downloaded on September 24,2025 at 04:33:32 UTC from IEEE Xplore. Restrictions apply.