Contents lists available at ScienceDirect

Integration, the VLSI Journal

journal homepage: www.elsevier.com/locate/vlsi

Ensemble learning model for effective thermal simulation of multi-core CPUs☆

Lin Jiang[1], Anthony Dowling, Yu Liu, Ming-C. Cheng*

Department of Electrical and Computer Engineering, Clarkson University, Potsdam, 13699-5720, NY, United States

ARTICLE INFO

Keywords:
Chip-level thermal simulation
Data-learning approach
Multi-core processors
Proper orthogonal decomposition
Galerkin projection

ABSTRACT

An ensemble data-learning approach based on proper orthogonal decomposition (POD) and Galerkin projection (EnPOD-GP) is proposed for thermal simulations of multi-core CPUs to improve training efficiency and the model accuracy for a previously developed global POD-GP method (GPOD-GP). GPOD-GP generates one set of basis functions (or POD modes) to account for thermal behavior in response to variations in dynamic power maps (PMs) in the entire chip, which is computationally intensive to cover possible variations of all power sources. EnPOD-GP however acquires multiple sets of POD modes to significantly improve training efficiency and effectiveness, and its simulation accuracy is independent of any dynamic PM. Compared to finite element simulation, both GPOD-GP and EnPOD-GP offer a computational speedup over 3 orders of magnitude. For a processor with a small number of cores, GPOD-GP provides a more efficient approach. When high accuracy is desired and/or a processor with more cores is involved, EnPOD-GP is more preferable in terms of training effort and simulation accuracy and efficiency. Additionally, the error resulting from EnPOD-GP can be precisely predicted for any random spatiotemporal power excitation.

1. Introduction

Associated with aggressively downscaled technology nodes, the power density of semiconductor chips is continuously increasing [1], which therefore results in higher thermal gradients and more and higher-temperature hot spots in the semiconductor chips and degrades their performance and reliability [2,3]. In addition, higher heat dissipation induced by the increased power density imposes a significant challenge to the cooling system of semiconductor chips [4], which in turn makes the thermal issues more severe. To maintain high performance and prolong their lifespan, dynamic thermal management has been implemented to reduce chip temperature and suppress hot spots [5,6]. This however requires an accurate and efficient prediction of the dynamic thermal profile in the chips with high resolution to capture all crucial hot spots. Many approaches have been employed for thermal simulations of semiconductor chips. For instance, direct numerical simulations (DNSs) based on the finite element method (FEM), finite difference method (FDM) or finite volume method (FVM) offer temperature solutions of semiconductor chips accurately. Due to a required large number of degrees of freedom (DoF), DNSs are computationally intensive and prohibitive for dynamic thermal management for large-scale semiconductor chips. To satisfy the demand of efficiency, several other approaches are developed including thermal circuit models [7–10], the Green's function method [11], machine learning based approaches [12,13], etc., by sacrificing accuracy and/or resolution of temperature solutions.

The block model of HotSpot [9], one of the popular thermal circuit models, realizes high efficiency by using large lumped thermal elements, which consequently results in low resolution and inaccurate temperature solutions. For some floorplans, the block model of HotSpot even leads to a 200% error, compared with DNSs [14]. The grid model of HotSpot [7] was thereby developed to improve the accuracy through dividing each functional unit (FU) of semiconductor chips into much smaller elements, and it inherently becomes computationally time-consuming similar to an FDM. For the Green's function method, temperature solutions are obtained through the convolution of the power map (PM) with the Green's function. The Green's function is the spatial impulse response to a unit point power source with an assumption that the chip is infinitely large. It is therefore difficult to take into account boundary conditions (BCs) for a realistic semiconductor chip

☆ This work is supported by National Science Foundation, United States under Grant Nos. ECCS-2003307 and OAC-2118079.

* Corresponding author.
E-mail address: mcheng@clarkson.edu (M.-C. Cheng).

[1] Current affiliation: Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China.

https://doi.org/10.1016/j.vlsi.2024.102201
Received 6 October 2023; Received in revised form 13 April 2024; Accepted 21 April 2024
Available online 24 April 2024

with a finite dimension [13,15]. Although the method of image has been used to address the issue, it is only valid for adiabatic BCs [11]. Moreover, it is hard to implement the Green's function method for transient thermal simulations [13,16]. To overcome this limitation, effort has been made such as the power blurring method [11]. In addition, applying the Green's function method to 3D dynamic thermal simulations still remains challenging. Due to the simplicity, machine learning approaches have become popular in recent years for thermal simulation of semiconductor chips [12,13,17]. It is however difficult to implement high spatial resolution in these approaches that would require extremely intensive training [17], especially in large-scale chips. Furthermore, due to the absence of any physical guidance, machine learning based approaches usually perform poorly in extrapolation cases.

Due to the ability to accomplish high accuracy, efficiency and resolution simultaneously, data-learning approaches enabled by proper orthogonal decomposition (POD) [18,19] have been attracting more attentions in recent years, especially in large-scale numerical simulations [20–22]. Using the POD, the problem of interest is projected from the physical space onto a POD space that is represented by a finite set of basis functions (or POD modes). The POD modes are extracted/trained by solution data collected from DNSs subjected to variations of spatiotemporal heat excitations and BCs. The heat transfer equation is then projected onto the POD modes to close the model using the Galerkin projection (GP) that also incorporates the fundamental physical principles of heat transfer into the model. This rigorous POD-GP simulation methodology generates an optimal set of POD modes that are tailored to capture essential thermal behavior induced by variations of spatiotemporal heat excitations and BCs. Together with the physics-based guidance enforced by the GP, the POD-GP methodology thus offers a very accurate and efficient prediction of dynamic thermal simulations of semiconductor chips at different levels if the quality of the solution data is adequate. POD-GP based approaches have been successfully applied to steady-state and dynamic thermal simulations of semiconductor devices [23,24], integrated circuits [25–27], interconnects [28] and microprocessors [29–31].

A global POD-GP thermal model (hereafter GPOD-GP) using the POD modes trained for an entire chip has been demonstrated in previous studies [29–31]. Using 5 modes in GPOD-GP, a speedup over 17,000 times was achieved in a multi-core processor with a high accuracy [29], compared to DNS via FEniCS using the FEM (FEniCS-FEM) [32]. However, an extensive training effort is needed for GPOD-GP to ensure good data quality and to cover enough spatial variation of dynamic PMs to maintain its accuracy, which becomes prohibitive for a large-scale chip. In this study, accuracy and robustness of GPOD-GP is first examined in a quad-core CPU, AMD ATHLON II X4 610e [33]. Based on the findings, GPOD-GP is revised and the ensemble POD-GP model (EnPOD-GP) is proposed to improve the training efficiency and simulation accuracy.

Fig. 1(a) depicts the workflow of EnPOD-GP training, including data collection from DNSs, generation of model parameters and development of EnPOD-GP. EnPOD-GP trains an individual POD-GP model (IPOD-GP) for the power source provided by each FU of a multi-core CPU using the temperature solution data of the entire CPU induced by the FU. Thus, there are a total of N_{FU} sets of POD modes, each of which is trained independently to construct EnPOD-GP, and there is no need for a training PM of the entire CPU. In contrast, GPOD-GP generates one set of global POD modes accounting for power sources in all FUs in the entire CPU to capture spatial variation of the power density provided by several dynamic PMs. With N_{FU} IPOD-GP models generated, an EnPOD-GP model is constructed for the multi-core CPU, as shown in Fig. 1(a). The temperature solution of the multi-core CPU can then be obtained by EnPOD-GP via the solution ($T_n(\vec{r},t)$) predicted by each of the N_{FU} IPOD-GP models, as shown in Fig. 1(b). To be more specific, $T_n(\vec{r},t)$ of the nth IPOD-GP model induced by the nth power source P_n is solved first for the entire CPU, and the dynamic temperature in the entire space $T(\vec{r},t)$ is then the sum of all $T_n(\vec{r},t)$ based on the superposition principle. Note that either GPOD-GP or EnPOD-GP solves the dynamic solution in its POD space first before calculating the spatiotemporal temperature in the entire chip. The details described in Fig. 1 will be presented in the following sections.

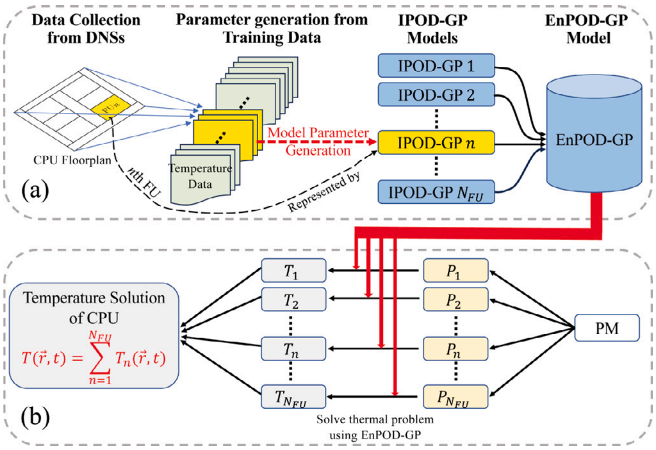
Data Collection from DNSs
Parameter generation from Training Data
IPOD-GP Models
EnPOD-GP Model
CPU Floorplan
FU n
nth FU
Temperature Data
Model Parameter Generation
Represented by
IPOD-GP 1
IPOD-GP 2
IPOD-GP n
IPOD-GP N_{FU}
EnPOD-GP
(a)
Temperature Solution of CPU
$T(\vec{r},t) = \sum_{n=1}^{N_{FU}} T_n(\vec{r},t)$
T_1
T_2
T_n
$T_{N_{FU}}$
P_1
P_2
P_n
$P_{N_{FU}}$
PM
Solve thermal problem using EnPOD-GP
(b)

Fig. 1. Workflow of (a) training for each of IPOD-GPs to construct the EnPOD-GP model, where no PM is needed, and (b) temperature prediction using EnPOD-GP for a multi-core CPU subjected to a dynamic PM with the spatial distribution, $P_1 - P_{N_{FU}}$. n denotes the nth FU and N_{FU} is the total number of FUs.

2. Global POD-Galerkin-Projection model

For the POD-based approach, the temperature solution is given by a linear combination of the POD modes φ_j,

$$T(\vec{r},t) = \sum_{j=1}^{M} a_j(t)\varphi_j(\vec{r}), \tag{1}$$

where a_j is the weighting coefficient and M is the selected number of modes. The POD mode is generated by maximizing the mean square inner product of the thermal solution with the POD mode, which thereby leads to an eigenvalue problem [18]

$$\int_{\Omega'} \langle T(\vec{r},t) \otimes T(\vec{r}',t)\rangle \varphi(\vec{r}')d\Omega' = \lambda\varphi(\vec{r}), \tag{2}$$

where $\langle\cdot\rangle$ denotes an average process over the sampled solution data, $\otimes$ is the tensor operator and λ is the eigenvalue representing the mean squared temperature captured by $\varphi(\vec{r})$. To close the model, a set of equations for a_i need to be derived. This can be achieved by the GP of the heat transfer equation onto each of the generated POD modes

$$\begin{gathered}\int_{\Omega}(\varphi_j(\vec{r})\frac{\partial\rho CT}{\partial t} + \nabla\varphi_j(\vec{r})\cdot k\nabla T)d\Omega = \\ \int_{\Omega}\varphi_j(\vec{r})P_d(\vec{r},t)d\Omega - \int_{S}\varphi_j(\vec{r})(-k\nabla T\cdot\vec{n})dS,\end{gathered} \tag{3}$$

where $j = 1$ to M, and k, ρ and C are material properties of chips (i.e., the thermal conductivity, density and specific heat, respectively), $P_d(\vec{r},t)$ is the interior power density, S is the boundary surface and $\vec{n}$ is its outward normal vector.

By incorporating (1), (3) can be expressed as a set of ordinary differential equations (ODEs) for a_j,

$$\sum_{j=1}^{M} c_{i,j}\frac{da_j}{dt} + \sum_{j=1}^{M} g_{i,j}a_j = P_i,\ i = 1 \text{ to } M, \tag{4}$$

where $c_{i,j}$, $g_{i,j}$ and P_i are the elements of the thermal capacitance matrix, thermal conductance matrix and power vector in the POD space, respectively. $c_{i,j}$ is defined as

$$c_{i,j} = \int_{\Omega}\rho C\varphi_i\varphi_j d\Omega. \tag{5}$$

Table 1
Percentage distribution of power consumption.

PM	Core 1	Core 2	Core 3	Core 4	Other units
1	16.2%	17.5%	16.2%	31.8%	18.3%
2	16.9%	18.3%	16.2%	29.5%	19.1%
3	15.6%	16.7%	35.7%	14.5%	17.5%
4	16.0%	17.0%	39.0%	10.0%	18.0%

As indicated by (5), $c_{i,j}$ are independent of BCs while $g_{i,j}$ and P_i are dependent on BCs. In this work, adiabatic and convective BCs are implemented to the boundary surfaces of the chip. For adiabatic boundary surfaces where the heat flux is zero, $g_{i,j}$ and P_i can be given as

$$g_{i,j} = \int_\Omega k\nabla\varphi_i \cdot \nabla\varphi_j d\Omega,\ P_i = \int_\Omega \varphi_i P_d(\vec{r},t) d\Omega. \tag{6}$$

As to convective boundary surfaces, the heat flux on the surfaces is described by

$$-k\nabla T \cdot \vec{n} = -k\frac{\partial T}{\partial n} = h(T - T_{amb}), \tag{7}$$

where h and T_{amb} are the heat transfer coefficient and ambient temperature, respectively. Plugging (7) into (3), $g_{i,j}$ and P_i for the convective BC become

$$g_{i,j} = \int_\Omega k\nabla\varphi_i \cdot \nabla\varphi_j d\Omega + \int_S h\varphi_i\varphi_j dS, \tag{8}$$

and

$$P_i = \int_\Omega \varphi_i P_d(\vec{r},t) d\Omega + \int_S h\varphi_i T_{amb} dS. \tag{9}$$

For GPOD-GP, the modes are trained by thermal data collected from FEniCS-FEM responding to applied PMs and BCs in the entire chip using a mesh with a grid number of 382,500 (i.e., 150 × 150 × 17 in x, y and z directions, respectively) and the grid size of $0.094 \times 0.081 \times 0.046$ mm^3. To examine the model, four dynamic PMs with different spatial distributions given in Table 1 are generated for the selected multi-core CPU. PMs 1 to 3 are generated by gem5 [34] and McPAT [35] using four selected benchmarks (Radix Soft, Monte Carlo, 2D heat and Advection Diffusion) [36]. The power distributions of PMs 1–3 are determined by the assignment of the benchmarks, as shown in Table 2. Notes that uniform power consumption is generated in each FU labeled in Fig. 2 for Cores 1–4, L2 Caches 1–4 and Northbridge only. McPAT does not generate power dissipation in I/O or DDR3 Channels due to gem5 limitations. There are thus 9 FUs in total. PM 1 is used to train one set of POD modes for the entire multi-core CPU. PMs 2 and 3 are test PMs, where PM 2 is slightly deviated from PM 1 but power applied to Cores 3 and 4 of PM 3 are very different from those in PM 1. As given in Table 1, the Core 3 power consumption in PM 3 is more than double of that in PM 1, and the Core 4 power consumption in PM 3 is less than half of that in PM 1. Two GPOD-GP models are developed in this work:

- GPOD-A: using data generated from PM 1
- GPOD-B: using data generated from both PMs 1 and 4.

PM 4 is created manually as an additional GPOD-B training PM, whose power in each of Cores 3 and 4 is selected such that power percentages of Cores 3 and 4 in PM 3 fall between those of the GPOD-B training PMs (i.e., PMs 1 and 4), as seen in Table 1. The power percentages in other cores/FUs PM 4 are similar to those in PM 1. This setting creates a simple scenario to examine the leaning ability of GPOD-B influenced by PM 3 that leads to dynamic thermal behavior bounded by the training data resulting from PMs 1 and 4.

The training procedure for obtaining POD modes and eigenvalues was detailed in the first two paragraphs of Section 2 and presented [30]. The theoretical least square (LS) error of GPOD-GP against DNS can be estimated by [30]

Table 2
Benchmark assignments for the generation of PMs [36].

PM	Core 1	Core 2	Core 3	Core 4
1	Radix Soft	Monte Carlo	2D heat	Adv. Diff.
2	Monte Carlo	2D heat	Adv. Diff.	Radix Soft
3	2D heat	Adv. Diff.	Radix Soft	Monte Carlo

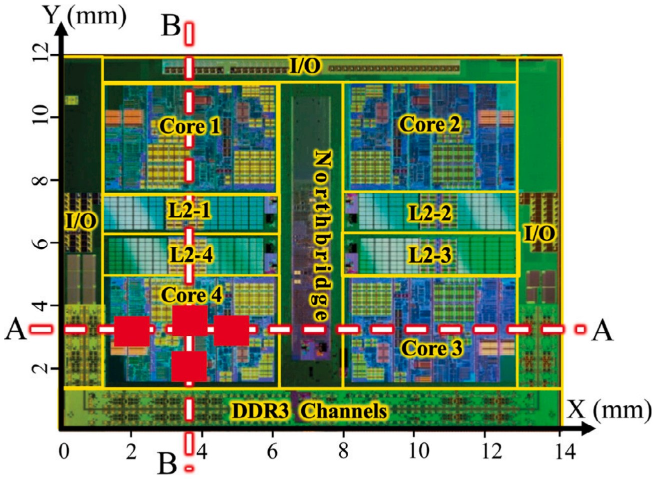

Fig. 2. Floorplan of AMD ATHLON II X4 610e CPU including localized high-power densities (red boxes with a size of 0.7×0.75 mm^2).

$$Err_{theo} = \sqrt{\sum_{j=M+1}^{N_s} \lambda_j \Big/ \sum_{j=1}^{N_s} \lambda_j}, \tag{10}$$

where N_s is the number of data samples. As a counterpart to the theoretical LS error, the numerical LS error is estimated by

$$Err_{num} = \sqrt{\frac{\sum_{i=1}^{N_t} \int_\Omega e^2(\vec{r},t_i) d\Omega}{\sum_{i=1}^{N_t} \int_\Omega \left[T(\vec{r},t_i) - T_{amb}\right]^2 d\Omega}}, \tag{11}$$

where N_t is the number of selected time steps in the thermal prediction, $T(\vec{r},t_i)$ is the temperature solution from FEM which is used as the baseline in this work, $e(\vec{r},t_i)$ is the temperature difference between FEM and GPOD-GP (or EnPOD-GP in Section 3) at the ith time step. Fig. 3 shows that the ideal LS error of GPOD-A is approximately half of GPOD-B's error until it is near 0.00003% beyond 80 modes. This is because the eigenvalue of GPOD-A decreases faster, as shown in Fig. 4, due to less information provided by the training data for GPOD-A.

It should be noted that the error predicted by (10) is accurate only if the quality of the training data is sufficient. The training data quality is predominantly influenced by (i) numerical accuracy of training temperature data collected from DNSs and (ii) the deviation between the training and testing simulations. The former is influenced by the numerical method and grid resolution implemented in DNSs; the latter in this work is determined by how well the PM in simulation is covered by the training PMs. With inaccurate numerical training data, the POD parameters in (4) cannot be estimated accurately. Accuracy of GPOD-GP is thereby deteriorated, which is extensively studied in [30,31]. When the simulation setting deviates from the training condition (influenced by PMs), the training is incomplete or inadequate and the error is always large with a small number of modes, which is investigated below.

To observe the learning ability of GPOD-GP influenced by the training data quality, GPOD-A simulations are demonstrated below using PMs 1 to 3. Fig. 3 shows that the LS training error of GPOD-A (i.e., using the training PM, PM 1, in simulation) is nearly identical

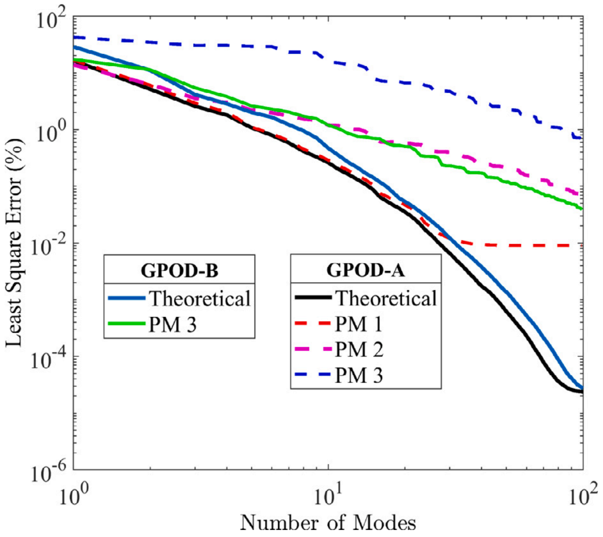

Fig. 3. Numerical and theoretical LS errors for the entire chip from GPOD-A and GPOD-B.

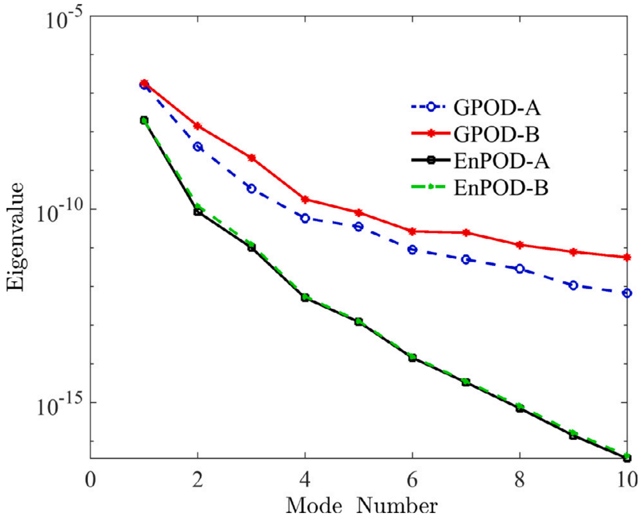

Fig. 4. Eigenvalues derived from GPOD-A, GPOD-B, EnPOD-A and EnPOD-B.

to Err_{theo} and stays near 0.01% beyond 35 modes due to computer precision. When applying PM 2 (whose power distribution is slightly different from PM 1; see Table 1), use of 3, 5, 11 or 15 modes leads to an LS error near 3.5%, 2.2%, 1.1% or 0.72%, respectively. The error reaches 0.069% with 99 modes and continues declining. When using PM 3 (very different power consumptions in Cores 3 and 4 from PM 1), its LS error becomes nearly an order larger, and it needs 52 modes to reach 2.2% (only 5 modes for the case using PM 2). The error from PM 3 reaches 0.69% with 99 modes and still continues decreasing. Even with inadequate training for GPOD-A subjected by PM 3, GPOD-A clearly demonstrates its learning ability in this case to perform accurate extrapolation when more modes are included.

The above findings resulting from the GPOD-A demonstration indicate that the training data quality is the key to offer both high accuracy and efficiency for GPOD-GP. When using the training data (collected from the stimulation subjected to PM 1) that provides nearly perfect data quality for GPOD-A, the LS error is as small as the ideal error given in (10), as depicted in Fig. 3. For the simulation using PM 2, the training data quality resulting from PM 1 is still sufficient to offer an accurate prediction with just around 5 modes. The quality of the same data from PM 1 is however insufficient for the simulation using PM 3, which thus leads to an LS error near 29% with 5 modes. However, even with an incomplete training leading to a poor data quality, GPOD-A using PM 3 still demonstrates its learning ability to perform accurate extrapolation far beyond its training setting when 50 or more modes are included. Such a remarkable learning ability stems from the GP in (3) that enforces physical principles of heat transfer in (4). To further validate the learning ability associated with the data quality, PM 3 is implemented in GPOD-B simulation, where PM 3 is within the bounds of PMs 1 and 4 that are both used to train GPOD-B. Fig. 3 illustrates the LS error induced by GPOD-B is greatly improved, which reaches 2.6% with 5 modes and 2.2% with 6 modes. It becomes even smaller than GPOD-A's error with PM 2 beyond 9 modes.

As demonstrated in Fig. 3, although GPOD-GP thermal model is able to reach a high accuracy and efficiency, multiple dynamic PMs are needed in the training to cover enough variation of the power distributions over all the cores. For CPUs/GPUs with a large number of cores, the training then becomes computationally prohibitive.

3. Ensemble POD-Galerkin-Projection model

To improve the GPOD-GP accuracy and training efficiency, the EnPOD-GP thermal simulation methodology is proposed in this work. Unlike GPOD-GP that trains one global set of modes for the entire chip, EnPOD-GP trains N_{FU} sets of POD modes, and each represents an IPOD-GP model for the entire processor excited only by the power source(s) in an FU. Similar to GPOD-GP, the modes of IPOD-GP for each power source are generated and optimized by maximizing the mean square inner product of the mode with the temperature solution data collected from DNS of the entire chip subjected to the corresponding power source. Therefore, for the nth IPOD-GP of the nth power source, the eigenvalue problem becomes

$$\int_{\Omega'} \langle T_n(\vec{r},t) \otimes T_n(\vec{r}',t)\rangle \varphi_n(\vec{r}')d\Omega' = \lambda_n \varphi_n(\vec{r}), \tag{12}$$

where $\varphi_n(\vec{r})$ and λ_n are the modes and eigenvalue of nth IPOD-GP. Once $\varphi_n(\vec{r})$ are determined, the temperature induced by the power consumption in the nth power source can be given by

$$T_n(\vec{r},t) = \sum_{j=1}^{M_n} a_{j,n}(t)\varphi_{j,n}(\vec{r}), \tag{13}$$

where M_n is the number of modes of the nth power source.

To obtain $a_{j,n}(t)$, as GPOD-GP does, the heat transfer equation is projected onto the POD space represented by the modes of the IPOD-GP for the nth power source

$$\begin{aligned}\int_{\Omega} (\varphi_{j,n}(\vec{r})\frac{\partial \rho C T_n}{\partial t} + \nabla\varphi_{j,n}(\vec{r}) \cdot k\nabla T_n)d\Omega = \\ \int_{\Omega} \varphi_{j,n}(\vec{r})P_{d,n}(\vec{r},t)d\Omega - \int_{S} \varphi_{j,n}(\vec{r})(-k\nabla T_n \cdot \vec{n})dS,\end{aligned} \tag{14}$$

where $P_{d,n}(\vec{r},t)$ is the interior power density in the nth power source. Together with (13), (14) can be rewritten as a set of ODEs

$$\sum_{j=1}^{M_n} c_{i,j}^{n}\frac{da_{j,n}}{dt} + \sum_{j=1}^{M_n} g_{i,j}^{n} a_{j,n} = P_{i,n},\ i = 1 \text{ to } M_n, \tag{15}$$

where $c_{i,j}^n$, $g_{i,j}^n$ and $P_{i,n}$ are the parameters of the IPOD-GP for the nth power source, and their expressions are identical to those given in (5)–(9). By solving (15), $a_{j,n}(t)$ are determined, and temperature caused by the nth power source is given by (13). Based on the superposition principle, the dynamic temperature solution of the entire processor is then the sum of temperatures derived from all the IPOD-GP models, i.e.,

$$T(\vec{r},t) = \sum_{n=1}^{N_{FU}} T_n(\vec{r},t) = \sum_{n=1}^{N_{FU}}\sum_{j=1}^{M_n} a_{j,n}(t)\varphi_{j,n}(\vec{r}), \tag{16}$$

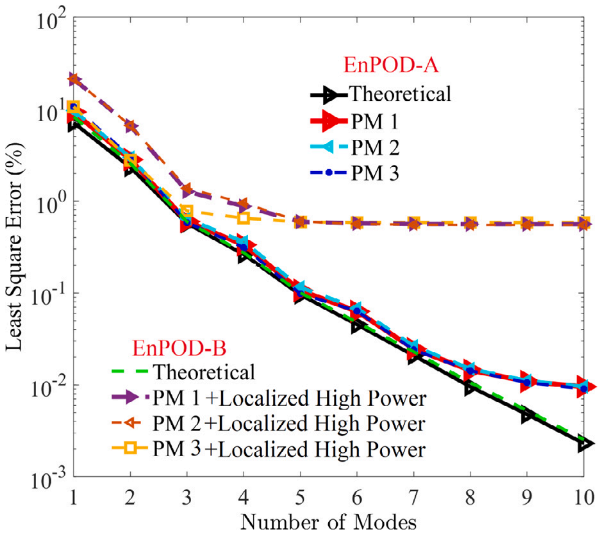

Fig. 5. LS errors of EnPOD-A and EnPOD-B vs. number of modes per FU.

where the indices denote the jth mode of the nth FU. In this study, $M_n = M$ for all units. The equivalent jth-mode eigenvalue considering all FUs in EnPOD-GP is defined by

$$\lambda_j^{eq} = \sum_{n=1}^{N_{FU}} \lambda_{j,n} w_n, \tag{17}$$

where w_n is the area fraction for the nth FU over the chip area. Using λ_j^{eq}, the theoretical LS error of EnPOD-GP can be estimated from (10).

Since each FU in EnPOD-GP is trained by the power source in the corresponding FU independently of other FUs, there is no training PM needed for entire processor and the training becomes more effective. To demonstrate this, 2 EnPOD-GP models are constructed. EnPOD-A is trained by a random uniform power density in each FU and EnPOD-B by the same except for Core 4 where 4 localized high-power densities (1 to 3 times higher than the uniform level) shown in Fig. 2 are added to the uniform power. Note that, since EnPOD-A is not exposed to localized high power density in the training, it only works well for the cases where power source in each FU is uniform. EnPOD-B is generated to examine the EnPOD-GP's ability to capture accurate hot spots induced by localized high-power injections. Moderate variation of the high-power injections in one core is implemented in the training. The same mesh size for GPOD-GP is used for training EnPOD-A but a finer mesh of 256 × 256 × 17 with a grid size of $0.055 \times 0.047 \times 0.046\ \text{mm}^3$ is applied for EnPOD-B with localized high power densities. Also, since only moderate variation of the high-power injections is implemented in one core during the training, Fig. 4 shows that EnPOD-B's eigenvalue declines only slightly slower than that of EnPOD-A. Compared with the eigenvalues of GPOD-A and GPOD-B, due to more effective training, eigenvalues of both EnPOD-A and EnPOD-B shown in Fig. 4 decrease significantly faster than those of GPOD-A and GPOD-B. Thus, Err_{theo} of both EnPOD-GP models illustrated in Fig. 5 decline relatively fast compared to both GPOD-GP models in Fig. 3.

Note that the LS error of EnPOD-A shown in Fig. 5 is nearly independent of the PMs because each of its IPOD-GP models is trained independently without the need of training PM. In addition, the LS error is accurately predicted in terms of the number of modes (i.e., the DoF that determines the computational time) by (10) using λ_j^{eq} until 8 modes and stays near 0.009% beyond 10 modes due to computer precision. This is very different from thermal circuit models or machine learning methods where the error cannot be predicted. Because of localized high power densities included in EnPOD-B, Fig. 5 reveals that its Err_{theo} is slightly larger than EnPOD-A's. To examine EnPOD-B, PMs 1–3 are applied, together with additional localized high-power densities injected at 4 locations in Core 4, as shown in Fig. 2. The localized high power densities increase local thermal gradients, which thus deteriorates the prediction accuracy. When they are applied to higher power regions, thermal gradients are further enhanced. The uniform power of Core 4 (where the localized high power density is applied) in PMs 1 and 2 are more than double of that in PM 3 (see Table 1). As a result, EnPOD-B induced by PMs 1 or 2 using 1 to 3 modes/FU leads to an LS error nearly double of that resulting from PM 3. However, because of the learning ability guided by physical principles incorporated via the GP given in (14), the LS errors of EnPOD-B resulting from all these 3 PMs, together with the localized high power densities, reduce drastically and merge gradually beyond 3 modes/FU. Their LS errors become less than 1% with 4 modes/FU and all equal 0.58% with 5 or more modes/FU.

The LS error resulting from EnPOD-B induced by localized high-power densities is however relatively large, compared to those from EnPOD-A. For example, EnPOD-A reaches an LS error of 0.31% with just 4 modes/FU. The relatively large LS error from EnPOD-B mainly arises from the less accurate numerical training data due to the high thermal gradients induced by localized high-power excitations. This is similar to the observation in [30] that demonstrated a significant improvement in the POD-GP accuracy by replacing a coarser-mesh training in DNS with a finer mesh in a situation with high thermal gradients. This is however very different from the incomplete training presented in Fig. 3, where the numerical accuracy in training data is high enough and physical principles of heat transfer enforced by the GP allow the incompletely trained POD modes to perform sophisticate extrapolation to reach high accuracy. On the contrary, less-accurate numerical training data for EnPOD-B resulting from high thermal gradients leads to a constant LS error, as observed for EnPOD-B in Fig. 5. This constant LS error is induced by the less-accurate POD modes generated by less-accurate numerical training data. Nevertheless, LS errors of EnPOD-B subjected to various PMs all merger to a very small value with a handful of modes. This demonstrates that EnPOD-GP is resilient to various PMs even in situations where localized high-power densities are applied. It should be noted that implementation of a finer mesh in EnPOD-B will further improve the accuracy without increasing its computing time, as suggested by the study in [30]; however more training effort will be needed.

More detailed solutions from GPOD-A, GPOD-B, EnPOD-A and EnPOD-B subjected to PM 3 are compared to results from FEniCS-FEM in Figs. 6 and 7. In Fig. 6, the temperature evolution at the intersection of Path A and B indicated in Fig. 2 shows that all of POD-GP models with 3 or more modes provide accurate dynamic temperature. Note that the dynamic temperature predicted by GPOD-A subjected to PM 3 happens to be reasonably accurate at this intersection located at $(3.61\ \text{mm}, 3.16\ \text{mm})$ even though the overall LS error for this case is close to 25% with 3 modes, as shown in Fig. 3, due to inadequate training data quality for GPOD-A subjected to PM 3. The large error for GPOD-A subjected to PM 3 is clearly observed in Fig. 7(a) and (b), where the LS error is as large as 20% when 7 modes are included (see Fig. 3). Compared with GPOD-A that is trained only from temperature data induced by PM 1, GPOD-B is generated via a more thorough training using temperature data influenced by PMs 1 and 4. GPOD-B with 3 or more modes therefore reaches a very good agreement with FEniCS-FEM, as shown in Figs. 6(b) and 7(c)–(d), although very small deviations are still observed in some locations, for instance, near $5.0\ \text{mm} < y < 6.5\ \text{mm}$ in Fig. 7(d). The substantial improvement for GPOD-B is due to the better data quality for PM 3. Fig. 7(e)–(h) also confirm the excellent agreement given in Fig. 5 between EnPOD-A/EnPOD-B using 3 or more modes per FU and FEniCS-FEM. Fig. 5 shows that use of 3 modes per FU in EnPOD-A or EnPOD-B reaches an LS error near 0.58% or 0.78%, respectively. For GPOD-B, 13 or 17 modes are needed, respectively, as indicated in Fig. 3.

To offer better visualization of the thermal profiles presented in Fig. 7, thermal maps for these profiles resulting from POD-GP models

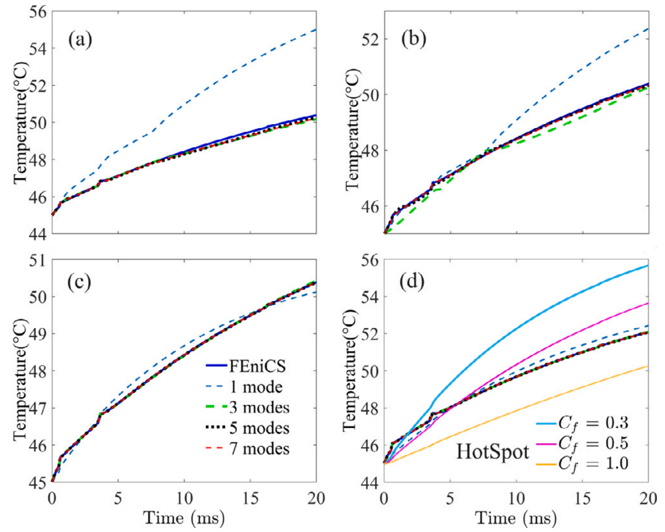

Fig. 6. Dynamic temperature at the intersection of Paths A and B given in Fig. 2 predicted by (a) GPOD-A, (b) GPOD-B and (c) EnPOD-A and (d) EnPOD-B and HotSpot, compared with FEniCS-FEM. All simulations here are performed subjected to PM 3, and EnPOD-B simulation includes additional localized high power densities.

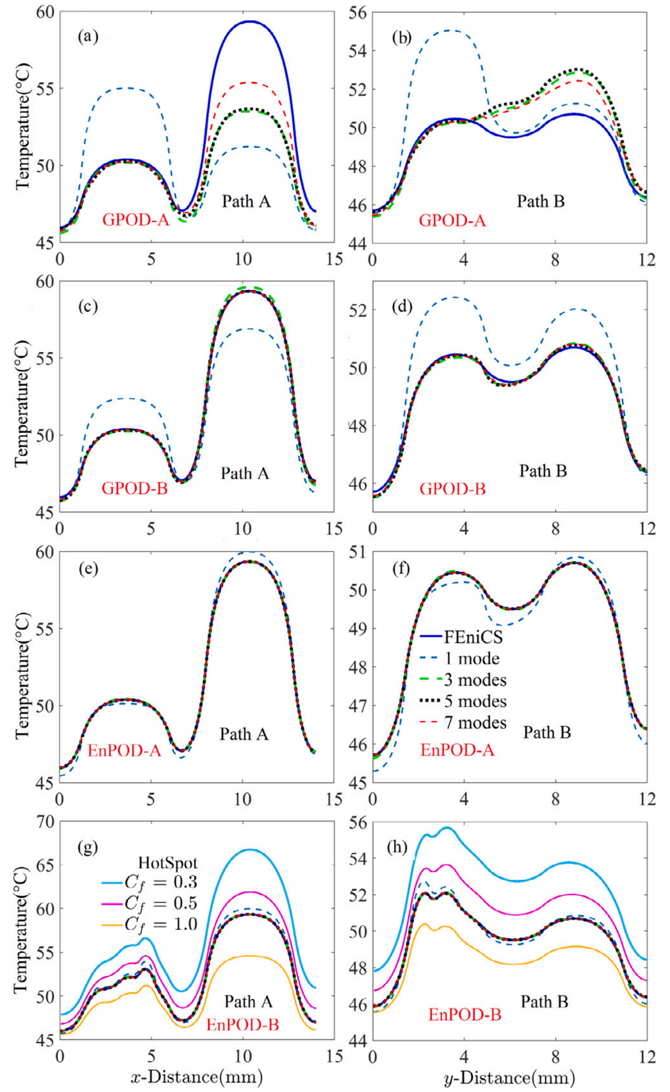

Fig. 7. Temperature profiles at $t = 20$ ms from GPOD-A, GPOD-B, EnPOD-A, EnPOD-B, FEniCS and HotSpot along Paths A (First column) and B (Second Column), respectively. Note that in EnPOD-A and EnPOD-B the number of modes is for each FU. The test PM, PM 3, is applied to all simulations, and additional localized high power densities are included in both HotSpot and EnPOD-B.

and FniCS-FEM at $t = 20$ ms are illustrated in Fig. 8. Similar to Fig. 7(a) and (b), the thermal map in Fig. 8(b) resulting from GPOD-A deviates significantly from Fig. 8(a) predicted by FEniCS-FEM due to inadequate training for GPOD-A in this case. Other POD-GP models offer very accurate thermal maps compared to FEniCS-FEM results. Alongside the LS error over the entire simulation time and space presented in Figs. 3 and 5, the absolute temperature error distribution with the maximum absolute error for each model over the entire simulation time is presented in Fig. 9. It can be seen in Fig. 9(b)–(d) that using only 3 modes GPOD-B leads to a maximum error near 0.5 °C and for both EnPOD-A and EnPOD-B the maximum error near 0.1 °C is achieved. In contrast, large absolute errors are observed in Fig. 9(a) for GPOD-A even with 7 modes, where the maximum is as high as 2.1 °C in Core 3.

For thermal simulations of large-scale semiconductor chips, thermal circuit models [7,8,11,14,37–39], are usually used because of their computing efficiency compared to DNSs. However, it has been shown in many studies that, even though their large time-scale (on the order of seconds) or steady-state thermal responses are close to the rigorous FEM results [7,8,11,14,37–39], large deviations from the FEM are always observed in smaller time scales [8,11,14,37]. A scaling factor C_f for capacitance elements is thus included in HotSpot [7] for users to fit the transient response for a certain period of time to obtain a reasonable agreement with FEM simulation [40]. One of the most popular state-of-the-art simulators, the grid model of HotSpot, is also applied to thermal simulation of the selected quad-core CPU using the same numerical settings used in the training of EnPOD-B. The spatiotemporal temperature solution derived from HotSpot is included in Figs. 6(d) and 7(g)–(h), compared to FEniCS-FEM and EnPOD-B. C_f is utilized to adjust its transient response, as seen in Figs. 6(d) and Fig. 7(g) and (h) for a thermal response to continuous random power excitations. Although a careful selection of C_f could approximately fit FEM results around several millisecond, it is not likely to work for a longer period unless large-timescale (in seconds) or steady-state results are of interest.

There are 9 IPOD-GP models in each of EnPOD-A and EnPOD-B. As shown in Table 3 for EnPOD-A, 9 × 3 ODEs (i.e., 3 modes for each IPOD-GP) given in (4) are needed to reach an LS error near 0.58% and $|\mathrm{Err}|_{max} = 0.088$ °C while only 17 ODEs (17 modes) in GPOD-B are needed to reach similar accuracy ($\mathrm{Err}_{\mathrm{LS}} = 0.57\%$ and $|\mathrm{Err}|_{max} = 0.086$ °C), as given in Table 4. It is noted that Table 4 includes the total number of modes (i.e., the dimension of the ODE system matrix) for GPOD-B. Table 3 however lists the number of modes per FU (i.e., for each IPOD-GP of EnPOD-A), where the number of modes for each FU is taken to be identical in this study. However, the 9 sets of ODEs in EnPOD-GP are independent and the ODE system matrix is thereby highly sparse. Specifically, the ODE system matrix for EnPOD-A is block-diagonal with very small block matrices. For instance, as shown in Table 3, use of 2 modes per IPOD-GP for EnPOD-A renders a set of ODEs with 9 small block matrices each with a size as small as 2 × 2, which leads to an LS error of 3%. When using 3 modes per IPOD-GP, the size of each block matrix becomes 3 × 3 and EnPOD-A reaches an LS error of 0.58%. For GPOD-B with 17 modes, all $g_{i,j}$ elements of the 17 × 17 ODE system matrix are non-zero. Thus, the computational time for solving the 17 ODEs using GPOD-B is not much different from what is needed for 27 ODEs needed in EnPOD-A (16.7 s vs. 18.8 s). Furthermore, based on the findings in Figs. 3–5, to reach a higher accuracy, the number of modes needed in GPOD-B will be larger than that needed in EnPOD-A, as also shown in Tables 3 and 4. This becomes more obvious for microprocessors with more cores since considerably more PMs are needed in the GPOD-GP training to account for power

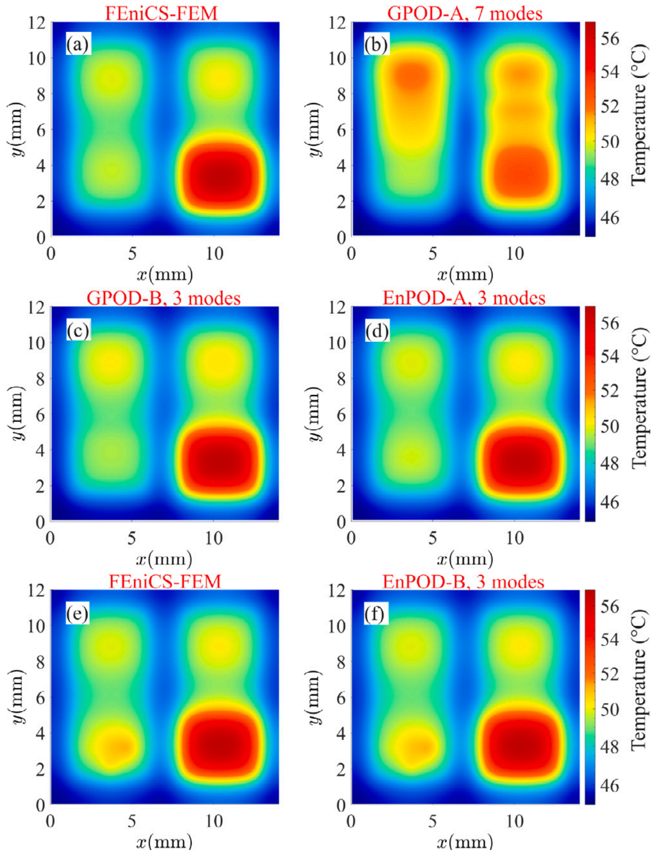

Fig. 8. Thermal maps at t = 20 ms in the device layer of the multi-core CPU subjected to PM 3 predicted by (a) FEniCS-FEM, (b) GPOD-A, (c) GPOD-B and (d) EnPOD-A. Similar thermal maps at t = 20 ms induced by PM 3 with localized high-power densities in Core 4 predicted by (e) FEniCS-FEM and (f) EnPOD-B.

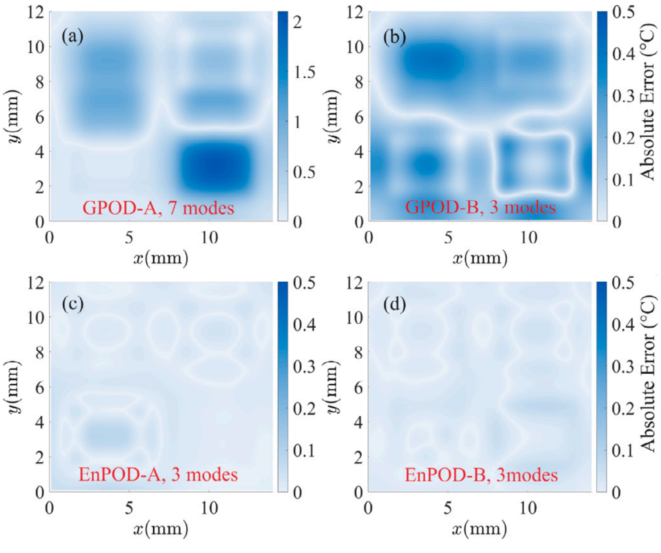

Fig. 9. Absolute error distributions induced by PM 3, where the maximum absolute error occurs, predicted by (a) GPOD-A, (b) GPOD-B, (c) EnPOD-A and (d) EnPOD-B, against FEniCS-FEM. EnPOD-B simulation includes additional localized high power densities. Note that the error scale in (a) is different from those in (b)–(d).

density variation among the cores. In contrast, an arbitrary random power excitation applied to each FU will be sufficient for EnPOD-GP without the need of a PM. Much more information provided by more training PMs will inevitably lead to a slower decrease in the eigenvalue.

Table 3
LS and maximum absolute errors and computational time for EnPOD-A associated with the number of modes. The number inside parentheses indicates the total number of POD modes.

Mode No./FU	Time (s)	Err_{LS} (%)	$\lvert\text{Err}\rvert_{\max}$ (°C)
1(9)	11.8	10.6	0.815
2(18)	15.2	3.00	0.594
3(27)	18.8	0.58	0.088
4(36)	23.0	0.31	0.061
5(45)	28.1	0.098	0.037
6(54)	32.9	0.063	0.029
7(63)	38.2	0.024	0.012

Table 4
LS and maximum absolute errors and computational time for GPOD-B associated with the number of modes.

Mode No.	Time (s)	Err_{LS} (%)	$\lvert\text{Err}\rvert_{\max}$ (°C)
2	6.5	11.4	1.075
5	7.9	2.6	0.340
11	12.3	1.03	0.153
17	16.7	0.57	0.086
27	25.0	0.32	0.046
55	48.9	0.098	0.022

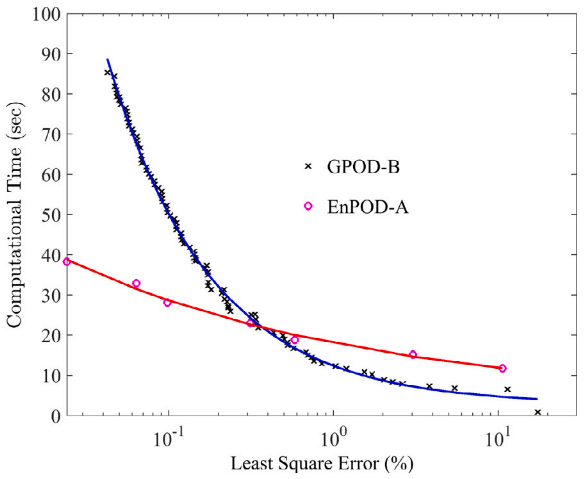

Fig. 10. Computational time vs. LS error of the entire chip for the thermal simulation with PM 3 using GPOD-B and EnPOD-A. The computational time for FEM is more than 23 h.

As shown in Fig. 4, the eigenvalue of GPOD-A declines faster than that of GPOD-B even though only one additional training PM is included in GPOD-B. Contrarily, the decrease in the eigenvalue for EnPOD-A is considerably faster (see Fig. 4) and its LS error is basically independent of the PMs (see Fig. 5).

As shown in Fig. 10 and Tables 3 and 4, when 18 ODEs with a diagonal block system matrix in EnPOD-A or 5 ODEs with a non-sparse system matrix in GPOD-B are used, the LS errors for both models are near 2.6%–3%. In this case with the similar errors, GPOD-B is around twice as fast as EnPOD-A. To reach an LS error below 2%, the GPOD-B computational time drastically increases while the time needed for EnPOD-A just increases slowly. As clearly illustrated in Fig. 10, the computing speed for EnPOD-A becomes considerably faster as higher accuracy is desired.

Results presented in Figs. 3–10 suggest that for a processor with many more cores EnPOD-GP is much preferable to GPOD-GP in terms of training effort and simulation efficiency and accuracy.

Training effort: As shown above, to ensure good data quality for the training, thermal data collected from DNSs needs to accounting for enough variations of PMs. For a large number of FUs, an enormous number of training PMs are needed for GPOD-GP and becomes computationally intolerable and prohibitive. Conversely, only one simple dynamic random PM is needed for an effective training of IPOD-GP to construct EnPOD-GP.

Simulation efficiency and accuracy: Even if the computationally intensive training of POD modes can be somehow achieved to construct GPOD-GP for a processor with many cores, the slowly decrease in the eigenvalue would require an very larger number of modes to reach reasonable accuracy. Contrarily, in EnPOD-GP, each FU (or each IPOD-GP) only needs 2 or 3 ODEs (i.e., 2 or 3 modes) to reach high accuracy; there are N_{FU} sets of ODEs and each set is independent of others. Thus, computational time of EnPOD-GP only increases linearly with N_{FU}; however, for GPOD-GP the number of modes increases drastically as more cores (more FUs) are involved in order to reach a reasonable accuracy.

For a small number of cores, GPOD-GP is faster unless extremely high accuracy is needed, as demonstrated in Fig. 10. As the number of cores is scaled up, the overhead of the increased DoF for EnPOD-GP is eased by its highly sparse system matrix, and EnPOD-GP will become considerably more efficient than GPOD-GP to reach good accuracy. One of the major overhead for POD-GP is the slow post processing calculations for temperature using (16), especially in a large-scale chip with fine spatial resolution. Practically for most chip-level applications relevant to thermal issues, only temperatures near hot spots and at some time instants are of interest. The times included in Fig. 10 and Tables 3 and 4 are estimated for evaluating temperature using (16) in the whole chip over the entire simulation time. If only the temperature at selected location near hot spots are of interest, the computational time and memory space will drastically reduce.

Regarding the computational time, it takes more than 30 h for FEniCS-FEM and 3.53 h for HotSpot to perform thermal simulation of the quad-core CPU subject to the localized high power injections. For EnPOD-B (instead of EnPOD-A given in Table 3), 32.5 s is needed using 2 modes per FU to reach $\mathrm{Err}_{\mathrm{LS}} = 2.8\%$ or $|\mathrm{Err}|_{max} = 0.59$ °C and 43 s is needed using 3 modes per FU to reach $\mathrm{Err}_{\mathrm{LS}} = 0.78\%$ or $|\mathrm{Err}|_{max} = 0.14$ °C for the entire CPU. As mentioned above, the computing speed can be significantly improved for EnPOD-B, if temperature is only calculated at certain points in time and/or space from (16).

4. Discussions

The demonstrations of GPOD-A and GPOD-B have illustrated the importance of the quality of training thermal data to reach both high efficiency and accuracy. When using poor-quality data due to incomplete training (e.g., the GPOD-A simulation with PM 3), Fig. 3 shows that GPOD-GP is still able to reach a high accuracy with many more modes. This is very different from most machine learning based methods, where the prediction far beyond the training setting in general fails. The simulation with PM 3 using GPOD-B whose training PMs, PMs 1 and 4, cover the variation in PM 3 demonstrates that an appropriate training setting in GPOD-GP can achieve both high accuracy and efficiency. Similar training settings can be performed using many PMs to cover a range of variation for each of major heating sources, Cores 1–4, without great difficulty. However, using more data from a larger number of training PMs, the eigenvalue will decrease more slowly, as observed in Fig. 4 for GPOD-B and GPOD-A. In addition, for a processor with considerably more cores, the computing effort to accommodate a huge number of PMs for training a set of POD modes to account for possible variations of all heat sources would become practically prohibitive.

On the other hand, the EnPOD-GP methodology trains IPOD-GP models independently, which offers simpler training settings, minimal training effort, fast declining eigenvalues, and extremely high accuracy. Compared to GPOD-GP, more computational time is needed for EnPOD-GP in a processor with a small number of cores unless a very high accuracy is needed, as shown in Fig. 10. In a processor with more cores, if one can manage to achieve a computationally extensive training for a well-trained GPOD-GP, its eigenvalue would decline considerably mores slowly and thus more modes (longer computational time) are needed to reach a good accuracy. Since the IPOD-GP models of EnPOD-GP are all independent, the large block-diagonal matrix with very small block matrices in EnPOD-GP will be more efficient than GPOD-GP. Moreover, when implementing a parallel computing environment (such as MPI) in the sparse EnPOD-GP system, EnPOD-GP will offer a superior advantage over the dense system matrix of GPOD-GP. EnPOD-GP will be further investigated in the near future on CPUs/GPUs with considerably more cores in parallel computing environments.

Compared to FEniCS-FEM simulation, a reduction in the DoF near 4 or 5 orders of magnitude is observed to achieve an LS error around 0.6%–3% for EnPOD-A or GPOD-GP, respectively. To reach an LS error near 3%, the computing speedup over FEniCS-FEM by 5459 and 10,504 times for EnPOD-A and GPOD-B is observed, respectively, and to reach an error near 0.58%, the speedup becomes 4413 and 4969 times. For the case with localized high power, to reach an LS error near 2.8% or 0.78%, EnPOD-B computing speed is 3355 or 2517 times faster than FEniCS-FEM and 390 or 295 times faster than HotSpot, respectively. It is noted that, after solving the ODEs in (4) in the POD space, the post processing for GPOD-GP or EnPOD-GP using (1) or (16), respectively, is needed to obtain spatiotemporal temperature. The post process takes 50 to 100 times longer than solving the ODEs [30]. Unlike thermal circuit models (e.g., HotSpot [7] or PACT [8]) or DNSs where solution over the entire time and space must be solved all together, for the POD-GP based approaches one can post-process temperature only at selected locations or time instants to significantly improve the computing efficiency and memory space.

5. Conclusions

The EnPOD-GP simulation methodology has been proposed for thermal simulation of multi-core CPUs. The pros and cons of EnPOD-GP and GPOD-GP have been investigated in terms of accuracy and efficiency, compared to FEniCS-FEM. GPOD-GP suffers from intensive training, and for a processor with a large number of cores, the training effort becomes computationally intolerable. However, the training for EnPOD-GP is very simple and effective, and the trained modes are resilient to any dynamic PM. EnPOD-GP thus offers very accurate spatiotemporal thermal solution in the selected multi-core processor subjected to different PMs. EnPOD-GP is however less efficient than GPOD-GP for processors with only a handful of cores unless a very high accuracy is desired. For a processor with more cores, EnPOD-GP will become relatively more efficient due to its highly sparse system matrix, and its LS error influenced by computational DoF can be accurately predicted. Compared to FEniCS-FEM to obtain the temperature profile in the whole multi-core CPU for the entire simulation time, a 3-order simulation speedup can be achieved with a high accuracy for both EnPOD-GP and GPOD-GP for the selected multi-core CPU. Compared to HotSpot, a 2-order speedup is observed for EnPOD-GP in the high resolution case. For most applications of chip-level thermal simulations, only temperature near hot spots at certain time instants are needed from (1) or (16), and computational times for EnPOD-GP and GPOD-GP would be one or two order shorter.

It is noted that construction of EnPOD-GP models solely relies on the quality of the training thermal data from DNSs for the structures of interest regardless of materials or complexity of the structures. The approach can be applied to semiconductor structures at different levels, including flip-chip and 2.5D/3D packaging technologies. Additionally, owing to the high efficiency and accuracy, the proposed EnPOD-GP thermal simulation methodology can be implemented for thermal predictions in runtime applications, such as true real-time thermal management (e.g., thermal-aware task scheduling), runtime reliability management, power management, etc. Based on the ability to predict accurate hot spots efficiently in microprocessors with a large number of cores, the developed EnPOD-GP simulation methodology will soon be applied to dynamic thermal analysis for modern GPUs.

CRediT authorship contribution statement

Lin Jiang: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Visualization. **Anthony Dowling:** Software, Data curation. **Yu Liu:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Ming-C. Cheng:** Conceptualization, Methodology, Formal analysis, Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S.I. Guggari, Analysis of thermal performance metrics—application to CPU cooling in HPC servers, IEEE Trans. Compon. Packag. Manuf. Technol. 11 (2) (2021) 222–232.

[2] A. Heinig, R. Fischbach, M. Dittrich, Thermal analysis and optimization of 2.5 D and 3D integrated systems with Wide I/O memory, in: Proc. ITHERM, IEEE, 2014, pp. 86–91.

[3] J. Zhou, J. Yan, K. Cao, Y. Tan, T. Wei, M. Chen, G. Zhang, X. Chen, S. Hu, Thermal-aware correlated two-level scheduling of real-time tasks with reduced processor energy on heterogeneous MPSoCs, J. Syst. Archit. 82 (2018) 1–11.

[4] D. Ansari, K.-Y. Kim, Hotspot thermal management using a microchannel-pinfin hybrid heat sink, Int. J. Therm. Sci. 134 (2018) 27–39.

[5] M.S. Mohammed, A.K. Al-Dhamari, A.A.-H. Ab Rahman, N. Paraman, A.A. Al-Kubati, M. Marsono, Temperature-aware task scheduling for dark silicon many-core system-on-chip, in: Proc. ICMSAO, IEEE, 2019, pp. 1–5.

[6] Y.G. Kim, M. Kim, J. Kong, S.W. Chung, An adaptive thermal management framework for heterogeneous multi-core processors, IEEE Trans. Comput. 69 (6) (2020) 894–906.

[7] W. Huang, K. Sankaranarayanan, R.J. Ribando, M.R. Stan, K. Skadron, An improved block-based thermal model in HotSpot 4.0 with granularity considerations, in: Proc. WDDD, 2007.

[8] Z. Yuan, P. Shukla, S. Chetoui, S. Nemtzow, S. Reda, A.K. Coskun, PACT: An extensible parallel thermal simulator for emerging integration and cooling technologies, IEEE Trans. CAD ICs Syst. 41 (4) (2021) 1048–1061.

[9] HotSpot 6.0 temperature modeling tool, 2021, http://lava.cs.virginia.edu/HotSpot/versions.htm (accessed on 1, October 2021).

[10] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, D. Atienza, 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling, in: Proc. ICCAD, IEEE, 2010, pp. 463–470.

[11] A. Ziabari, J.-H. Park, E.K. Ardestani, J. Renau, S.-M. Kang, A. Shakouri, Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices, IEEE Trans. VLSI Syst. 22 (11) (2014) 2366–2379.

[12] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, P. Beckman, Machine learning-based temperature prediction for runtime thermal management across system components, IEEE Trans. Parallel Distrib. Sys. 29 (2) (2017) 405–419.

[13] H. Sultan, A. Chauhan, S.R. Sarangi, A survey of chip-level thermal simulators, ACM Comput. Surv. 52 (2) (2019) 1–35.

[14] D. Fetis, P. Michaud, An evaluation of HotSpot-3.0 block-based temperature model, in: Proc. WDDD, 2006.

[15] Y. Zhan, S.S. Sapatnekar, High-efficiency green function-based thermal simulation algorithms, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 26 (9) (2007) 1661–1675.

[16] S. Varshney, H. Sultan, P. Jain, S.R. Sarangi, Nanotherm: An analytical fourier-boltzmann framework for full chip thermal simulations, in: Proc. ICCAD, IEEE, 2019, pp. 1–8.

[17] H. Sultan, S.R. Sarangi, Variability-aware thermal simulation using CNNs, in: Proc. VLSID, IEEE, 2021, pp. 65–70.

[18] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1) (1993) 539–575.

[19] J.L. Lumley, The structure of inhomogeneous turbulent flows, Atmosp. Turbul. Radio Wave Propag. (1967) 166–178.

[20] M. Rovira, K. Engvall, C. Duwig, Proper orthogonal decomposition analysis of the large-scale dynamics of a round turbulent jet in counterflow, Phys. Rev. Fluids 6 (1) (2021) 014701.

[21] G. Jiang, H. Liu, K. Yang, X. Gao, A fast reduced-order model for radial integral boundary element method based on proper orthogonal decomposition in nonlinear transient heat conduction problems, Comput. Methods Appl. Mech. Engrg. 368 (2020) 113190.

[22] S. Fresca, A. Manzoni, POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114181.

[23] W. Jia, B.T. Helenbrook, M.-C. Cheng, Thermal modeling of multi-fin field effect transistor structure using proper orthogonal decomposition, IEEE Trans. Electron Devices 61 (8) (2014) 2752–2759.

[24] R. Venters, B.T. Helenbrook, K. Zhang, M.-C. Cheng, Proper-orthogonal-decomposition based thermal modeling of semiconductor structures, IEEE Trans. Electron Devices 59 (11) (2012) 2924–2931.

[25] W. Jia, B.T. Helenbrook, M.-C. Cheng, Fast thermal simulation of FinFET circuits based on a multiblock reduced-order model, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 35 (7) (2016) 1114–1124.

[26] D.S. Meyer, B.T. Helenbrook, M.-C. Cheng, Proper orthogonal decomposition-based reduced basis element thermal modeling of integrated circuits, Internat. J. Numer. Methods Engrg. 112 (5) (2017) 479–500.

[27] M.-C. Cheng, W. Jia, B.T. Helenbrook, Thermal modeling for FinFET NAND gate circuits using a multi-block reduced-order model, in: Proc. THERMINIC, IEEE, 2015, pp. 1–4.

[28] W. Jia, M.-C. Cheng, A methodology for thermal simulation of interconnects enabled by model reduction with material property variation, J. Comput. Sci. 61 (2022) 101665.

[29] L. Jiang, M. Veresko, Y. Liu, M.-C. Cheng, An effective physics simulation methodology based on a data-driven learning algorithm, in: Proc. PASC, 2022, pp. 1–10.

[30] L. Jiang, Y. Liu, M.-C. Cheng, Fast-accurate full-chip dynamic thermal simulation with fine resolution enabled by a learning method, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 42 (8) (2023) 2675–2688.

[31] L. Jiang, A. Dowling, M.-C. Cheng, Y. Liu, PODTherm-GP: A physics-based data-driven approach for effective architecture-level thermal simulation of multi-core CPUs, IEEE Trans. Comput. 72 (10) (2023) 2951–2962.

[32] FEniCS project, 2023, https://fenicsproject.org/ (accessed on 1, October 2023).

[33] K. Dev, A.N. Nowroz, S. Reda, Power mapping and modeling of multi-core processors, in: Proc. ISLPED, IEEE, 2013, pp. 39–44.

[34] N. Binkert, B. Beckmann, G. Black, S.K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D.R. Hower, T. Krishna, S. Sardashti, et al., The gem5 simulator, ACM SIGARCH Comput. Archit. News 39 (2) (2011) 1–7.

[35] S. Li, J.H. Ahn, R.D. Strong, J.B. Brockman, D.M. Tullsen, N.P. Jouppi, McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures, in: Proc. MICRO, 2009, pp. 469–480.

[36] A. Dowling, F. Swiatowicz, Y. Liu, A.J. Tolnai, F.H. Engel, COMBS: First open-source based benchmark suite for multi-physics simulation relevant HPC research, in: Proc. ICA3PP, Springer, 2020, pp. 3–14.

[37] W. Huang, K. Skadron, S. Gurumurthi, R.J. Ribando, M.R. Stan, Differentiating the roles of IR measurement and simulation for power and temperature-aware design, in: Proc. ISPASS, IEEE, 2009, pp. 1–10.

[38] H.-H. Chu, Y.-C. Kao, Y.-S. Chen, Adaptive thermal-aware task scheduling for multi-core systems, J. Syst. Softw. 99 (2015) 155–174.

[39] J.-H. Han, X. Guo, K. Skadron, M.R. Stan, From 2.5 D to 3D chiplet systems: Investigation of thermal implications with HotSpot 7.0, in: Proc. ITHERM, IEEE, 2022, pp. 1–6.

[40] W. Huang, K. Sankaranarayanan, K. Skadron, R.J. Ribando, M.R. Stan, Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model, IEEE Trans. Comput. 57 (9) (2008) 1277–1288.