Full-Chip Thermal Map Estimation for Commercial Multi-Core CPUs with Generative Adversarial Learning* (invited paper)

Wentian Jin[1], Sheriff Sadiqbatcha[1], Jinwei Zhang[1], and Sheldon X.-D. Tan[1]
[1]Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
wjin018@ucr.edu,ssadi003@ucr.edu,jzhan319@ucr.edu,stan@ece.ucr.edu

ABSTRACT

In this paper, we propose a novel transient full-chip thermal map estimation method for multi-core commercial CPU based on the data-driven generative adversarial learning method. We treat the thermal modeling problem as an image-generation problem using the generative neural networks. In stead of using traditional functional unit powers as input, the new models are directly based on the measurable real-time high level chip utilizations and thermal sensor information of commercial chips without any assumption of additional physical sensors requirement. The resulting thermal map estimation method, called *ThermGAN* can provide tool-accurate full-chip *transient* thermal maps from the given performance monitor traces of commercial off-the-shelf multi-core processors. In our work, both generator and discriminator are composed of simple convolutional layers with Wasserstein distance as loss function. *ThermGAN* can provide the transient and real-time thermal map without using any historical data for training and inferences, which is contrast with a recent RNN-based thermal map estimation method in which historical data is needed. Experimental results show the trained model is very accurate in thermal estimation with an average RMSE of 0.47°C, namely, 0.63% of the full-scale error. Our data further show that the speed of the model is faster than *7.5ms* per inference, which is two orders of magnitude faster than the traditional finite element based thermal analysis. Furthermore, the new method is ∼4x more accurate than recently proposed LSTM-based thermal map estimation method and has faster inference speed. It also achieves ∼2x accuracy with much less computational cost than a state-of-the-art pre-silicon based estimation method.

KEYWORDS

Thermal Modeling, Temperature Estimation, Processor Thermal Maps, Generative Adversarial Learning, Machine Learning

ACM Reference Format:
Wentian Jin[1], Sheriff Sadiqbatcha[1], Jinwei Zhang[1], and Sheldon X.-D. Tan[1]. 2020. Full-Chip Thermal Map Estimation for Commercial Multi-Core CPUs with Generative Adversarial Learning (invited paper). In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '20), November 2–5, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3400302.3415764

*This work is supported in part by NSF grants under No. CCF-1816361, in part by NSF grant under No. CCF-2007135 and No. OISE-1854276.

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11.
https://doi.org/10.1145/3400302.3415764

1 INTRODUCTION

As technology advances, today's high-performance microprocessors are becoming more thermally constrained due to steadily increasing power densities [1]. To enhance reliability, many system-level thermal/power regulation techniques such as clock gating, power gating, dynamic voltage and frequency scaling (DVFS) and task migration have been proposed in the past [2–4]. One critical aspect of the algorithms mentioned above is correctly estimating the full-chip temperature profile to properly guide the online thermal management schemes [5, 6]. However, accurate thermal estimation is a difficult task, especially for commercial off-the-shelf multi-core processors.

Some of the existing methods depend on the on-chip temperature sensors. However, very few physical sensors are typically available, and they may not be located in close proximity to the true hot-spots on the chip, consequently misleading the temperature regulation decision [7]. Hence, the more popular solution is to supplement the data from the few on-chip sensors with estimated temperatures of all the prominent hot-spots on the chip via thermal models based on estimated power-traces [8]. These methods offer higher spatial resolution as they allow for the temperature of all the hot-spots on the chip to be monitored in real-time [9–12].

However, the existing thermal modeling methods still suffer a few drawbacks. First, they need accurate power-traces as inputs; but estimating the power of each functional unit (FU) of a real processor under varying workloads is not a trivial task, if not infeasible [13, 14]. On the other hand, from the system-level thermal or power management perspective, the parameters that can be easily accessed are core frequency, voltage, and many utilization or performance metrics natively supported by most commercial processors [15]. Examples include Intel's Performance Counter Monitor (PCM) [16] and AMD's uProf [17]. Thermal models parameterized by these parameters will be more desirable and practical. Second, it is difficult to calibrate these models for practical use due to simplified modeling, boundary conditions, and the lack of sufficient accuracy. Lastly, most models such as HotSpot [10] still employ expensive numerical methods to find temperature solutions, which may not be fast enough for real-time use.

On the other hand, estimating the full-chip 2D thermal map of multi-core CPUs from given performance monitor parameters can be viewed as imaging synthesis problem. We can treat the performance monitor parameters as extracted latent features for power

information of the chip. Then we can synthesize the 2D thermal maps once the neural network are trained for the utilization to temperature transformation. Such training and image generation process can be carried out using generative adversarial networks (GAN), which is a popular generative deep neural networks for imaging synthesis, semantic imaging editing, style transfer, image superresolution etc [18, 19].

Inspired by this observation, In this work, we propose a novel data-driven fast transient full-chip thermal map estimation method for multi-core commercial CPU by exploiting the conditional generative adversarial learning. The new contributions are as follows:

1. First, *ThermGAN* can be implemented on most, if not all, existing commercial multi-core microprocessors as it only uses the existing temperature sensors and workload independent utilization information. In other words, our strictly post-silicon approach does not require any modifications to the chip's design.
2. We propose to treat this existing thermal modeling problem as the image generation problem conditioned on high-level performance monitors, which are available in most, if not all, commercial microprocessors. Then we propose to explore the conditional generative neural network structure in which the input high-level performance data are treated as categorical conditions.
3. In our work, we use simple memory-less convolutional neural network for both generator and discriminator with Wasserstein distance as loss function. We demonstrate that the proposed *ThermGAN* can estimate transient and real-time thermal map without using any historical data for training and inferences, which is contrast with a recent LSTM-based thermal map estimation method in which historical data is needed [20].
4. We use an advanced infrared thermography setup system, that enables lucid heatmaps to be recorded directly from commercial microprocessors while they are under load. A total number of 257400 pairs of PCM data and thermal maps were collected and 75% were used for training.
5. The resulting *ThermGAN* can provide tool-accurate full-chip *transient* thermal maps from the given performance monitor traces of commercial off-the-shelf multi-core processors.

Experimental results show the trained model is very accurate in thermal estimation with an average RMSE of 0.47°C, namely, 0.63% of the full-scale error. Our data further show that the speed of the model is less than *7.5ms* per inference, which is two orders of magnitude faster than the traditional finite element based thermal analysis and is suitable for real-time thermal estimation. Furthermore, the new method is ~4x more accurate than recently proposed LSTM-based thermal estimation method [20] and has faster inference speed. It also achieves ~2x accuracy with much less computational cost than the EigenMaps method [21], which is a state-of-the-art pre-silicon method.

2 RELATED WORK

To estimate the on-chip temperature maps, there are two general strategies. The first is to estimate the full-chip heatmaps from physics-based thermal models and power related information [11, 12]. Such *bottom-up* numerical methods such as HotSpot [10] based simplified finite difference methods, finite element methods [22], equivalent thermal RC networks [23], and the recently proposed top-down behavioral thermal models based on matrix pencil method [24] and subspace identification method [25, 26]. In general, full-chip thermal analysis from given power information requires expensive numerical analysis such as finite difference or finite element based approaches, which are very expensive for on-line applications [27]. Second method is to use an interpolation based approach to estimate the full-chip heatmaps from the embedded sensor readings [8, 28]. Since the number of sensors and their placement have a significant impact on the accuracy of the aforementioned interpolation, smart sensor placement algorithms have also been proposed that can be used during design time to find the optimal placement for the given budget of embedded temperature sensors [21, 28–32]. Work in [28] exploits Fourier analysis techniques to fully recover the thermal map. But the accuracy is limited by the nonband-limited nature of the temperature signals and approximations required for nonuniform placement of the thermal sensors, which is common in heterogeneous multi-core processors. Nowroz *et al.* [29, 30] tried to minimize the number of thermal sensors in the sensor placement to recover thermal maps (or some key locations) based on interpolation of hard sensor information in frequency domain and DC domain respectively. Such strategy was further improved by using Eigen decomposing of the interpolation matrix, which leads to near optimal sensor number and placement [21]. Zhang *et al* [31, 33] proposes a statistical method for both power and thermal maps estimation, in which the correlations of power dissipation of different modules of a chip were exploited to recover the power map from sensor readings first and temperature was estimated once power map is obtained. However, the estimation based on the power correlation information. Recently Ziabari *et al* [34] introduced the power blurring method for fast 2-D temperature map computation, which essentially is the Green's function based method in which temperature response to unit power impulses have to be computed first from FEM thermal analysis. This make this method difficult to be applied practically as accurate thermal models are not always available first.

However, the aforementioned methods either require design-time hardware changes (inserting or relocating sensors) or at the very least require detailed knowledge of the chip's floorplan, correlations among functional unit power sources, and constants specific to the technology-node which are not disclosed by the original chip manufacturer. An exclusively *post-silicon* approach to real-time transient estimation of the spatial temperature distribution across the entire chip area (i.e. at time t, estimate the full-chip spatial heatmap $T(x, y)_t$) remains a challenge for existing commercial microprocessors.

On the other hand, recently, machine-learning (especially deep-learning) is gaining much attention due to the breakthrough performance in various cognitive applications such as visual object recognition, object detection, speech recognition, natural language understanding, etc., due to dramatic accuracy improvements in their time-series or sequential modeling capabilities [18]. Machine-learning for electronic design automation (EDA) is also gaining significant traction as it provides new computing and optimization paradigms for many of the challenging design automation

problems that are complex in nature. For instance, machine learning methods have been applied to power modeling [35] and design space exploration [36]. Additionally, machine-learning based schemes have recently been explored to build a workload-dependent thermal prediction model [15], where the future steady-state temperature of the chip can be predicted by application characteristics and physical features.

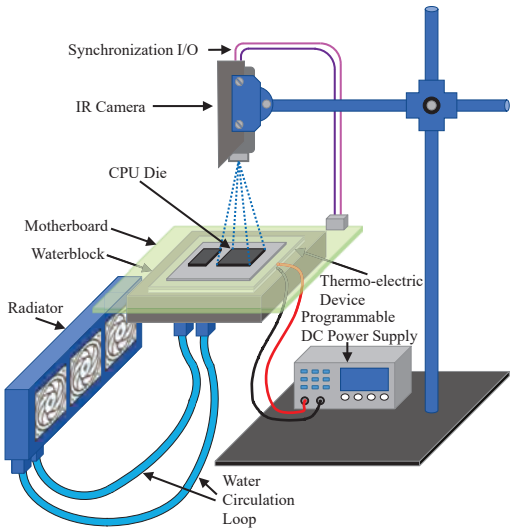
Recently long-short-term memory (LSTM) based machine learning approach based on Intel Performance Counter Monitor (PCM) metrics has been proposed for hot spot detection [37**?**] and for full-chip thermal map estimation [20] of commercial off-the shelf multi-core processors. To improve the efficiency, 2D discrete cosine transformation (DCT) is used to compress the thermal images for the learning process [20]. But this method needs to know the historical data of both PCM and temperatures for the training, which can be expensive. Furthermore, the accuracy of this approach is still less than expected due to the data compression process.

Recently GAN-based methods have been applied for VLSI physical designs such as generation of the various noise maps to facility the IR-drop noise sensor placement [38], for layout lithography analysis [39] and sub-resolution assist feature generation [40], for analog layout well generation [41]. But less studies have been investigated for data-driven circuit level and thermal analysis to model the dynamic systems described by the partial differential equations.

3 TRAINING DATA PREPARATION

Sufficient data is always vital for machine learning methods. To let the proposed model learn the distribution of PCM data and map it to correct thermal distribution map, sufficient training data is a must for it. In this work, a large amount of thermal distribution data of the CPU (called *thermal map* in this work) and corresponding real-time PCM data is required and from which the model can learn the transformation scheme in between. In what follows, we will present the setup used to acquire the training data. Some necessary pre-processing methods performed on the training set prior to feeding them to the model will also be discussed.

To externally acquire accurate thermal maps of a working CPU, we propose to use a measurement system based on an infrared (IR) camera. Fig. 1 illustrates the overall setup of our thermography system. The IR camera over the chip is a FLIR A325sc (16-bit 320×240 pixels, 60Hz). The camera is rated for the temperature range of $0°C$ to $328°C$, and spectral range of $7.5\mu m$ to $13\mu m$. A microscope lens is used to provide a finer spatial resolution of $50\mu m$/px. The CPU used in our test is an Intel i7-8650U working on an Intel ® NUC7i7DNHE motherboard with the stock CPU cooler removed. The distance between the camera and the chip is approximately $70mm$. When the CPU is running, the thermo-electric device mounted at the back of the chip transfers heat from its upper side to the other. The water block and circulation loop attached below further dissipates the heat into the radiator where the heat finally radiates to the air. With such setup, we are able to maintain the temperature of the CPU within its specified range as the stock cooler between the IR camera and the chip is removed. To synchronize the captured thermal map with its corresponding PCM data, we connect the IR camera and the CPU through a synchronization I/O. Each thermal map and PCM data that were collected in the same time instant are paired and saved together as one sample.



(a)

(b)

Figure 1: IR thermography setup used to collect training data in this work

PCM is a tool from Intel which monitors performance and energy metrics of all series of Intel processors. The monitored metrics range widely from basic processor monitoring utilities, such as instructions per cycle (IPC) and core frequency, to sleep and energy states of processor, and to peripheral memory bandwidth and cache miss. A number of APIs are provided for real-time monitoring which is highly suitable for our real-time full chip thermal modeling application. The complete list of all 170 PCM metrics that we collect and employ for the thermal modeling of Intel i7-8650U is shown in Table 1.

The temperatures in each thermal map vary widely from $25°C$ to $100°C$ while the values of the metrics in PCM data have all kinds of scales. Some metrics only changes in a small range around zero while others range widely with several orders of magnitude. Such inconsistencies in data scales may cause severe instability and accuracy degeneration in neural networks. Before feeding them to

Table 1: Performance metrics (Intel PCM)

Pkg.	Socket	Socket	Core 1 to 8
INST	EXEC	C6res%	EXEC
ACYC	IPC	C7res%	IPC
TIME	FREQ	C2res%	FREQ
PhysIPC	AFREQ	C3res	AFREQ
PhysIPC%	L3MISS	C6res	L3MISS
INSTnom	L2MISS	C7res	L2MISS
INSTnom%	L3HIT	C8res%	L3HIT
C0res%	L2HIT	C9res%	L2HIT
C2res%	L3MPI	C10res%	L3MPI
C3res%	L2MPI	SKT0	L2MPI
C6res%	READ		C0res%
C7res%	WRITE		C1res%
C8res%	TEMP		C3res%
C9res%	C0res%		C6res%
C10res%	C1res%		C7res%
Energy	C3res%		TEMP

the machine learning model, all data must be rescaled to comparable ranges. In this work, to accommodate to the *tanh* activation function employed in our model, as detailed in Section 4, we rescale all thermal maps to the range of [-1,1] using min-max normalization scheme as is given in (1). For PCM data, we rescale all metrics to mean of 0 and standard deviation of 1 using data standardization method.

$$Data'_{ij} = \left(\frac{Data_{ij} - min(Data)}{max(Data) - min(Data)} \times 2\right) - 1 \quad (1)$$

Fig. 2 illustrates the flow of conventional thermal modeling for full-chip estimation and our proposed ThermGAN method. There are multiple stages in the conventional flow. First, only thermal related metrics are extracted from the PCM data while the exact locations of the thermal sensors are unknown. The thermal model should predict the sensor locations prior to perform the actual thermal estimation. As the final estimation is based only on the sensor data, the accuracy of full-chip thermal modeling is inherently limited. As is shown in the lower flow in Fig. 2, our proposed GAN based method takes all PCM data as input and is trained on measured thermal maps. The unknown physics-law governing the transmission between them is automatically learned by the model which makes it possible for high-accuracy full-chip thermal modeling.

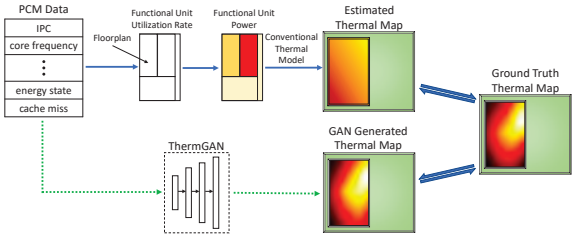


Figure 2: Conventional thermal modeling flow and the proposed ThermGAN flow.

We remark that the proposed thermal modeling technique is orthogonal to specific CPU being modeled and the way thermal maps are obtained. It can work for any real-time monitoring metrics to full-chip thermal modeling of commercial multi-processor chips. The CPU we choose in this work is only for illustration purpose. Further more, the thermal maps obtained in this work is from the set up without heat sinks due to the imaging measurement requirement. But the proposed method can work for any obtained or computed thermal maps. Research is under way to obtain accurate transient thermal maps from CPUs running in the practical setup with heat sinks.

4 CGAN-BASED PCM TO TEMPERATURE TRANSFORMATION

4.1 From PCM to thermal image transformation

We first show that we can view the full-chip thermal map estimation process for a multi-core processor as image synthesis process, in which the DNN can convert the features (PCMs) and continuous time variable into an image.

4.2 Review of GANs

Generative Adversarial Net (GAN) was first introduced by Ian Goodfellow in 2014 [42] and has drawn tremendous attention during the past few years. A typical GAN consists of two networks known as discriminator **D** and generator **G**. The generator takes a random vector **z**, usually normally distributed, as its input and maps it to an output image as close to those in the training dataset as possible. Images in the training set are labeled as 'real' images, and the ones produced by the generator are noted as 'fake'. The discriminator takes either real or fake image as its input and discriminates them from each other. Both **D** and **G** are trained simultaneously, and such process is a contest between these two networks. The generator keeps optimizing itself to fool the discriminator with fake images while the discriminator also strives to increase its classification accuracy. Once the GAN is trained, the generator should be able to generate real-like images by mapping its random input to the learned distribution of real images. The discriminator, on the other hand, will classify all its input images to be "real" or "fake" with the same possibility of 50%, which indicates that fake and real images look pretty much alike and are no longer distinguishable by the discriminator.

The training of GAN is usually a tricky process and may never converge due to gradient vanishing problem. Wasserstein GAN (WGAN) was introduced by Martin Arjovsky in [43] to mitigate this issue. Wasserstein Distance, rather than the conventional JS-Divergence, was proposed to serve as the measurement of the difference between real and fake image distributions. With such a small change in the loss function, WGAN promises a more stable training process and less likelihood of mode collapse. The results have shown significant advantages of GAN over the conventional methods in terms of both performance and accuracy.

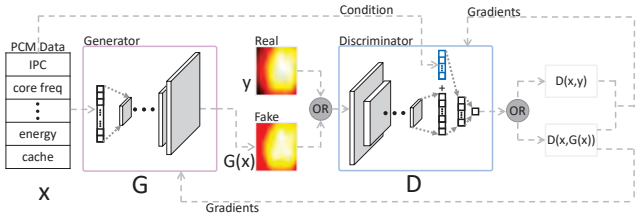


Figure 3: The proposed ThermGAN framework.

Fig. 3 illustrates our proposed structure of PCM data to thermal map WGAN. The raw PCM data $\mathbf{z}$ given to the generator $\mathbf{G}$ is a 1×170 vector with all entries standardized around zero as described in Section 3. Both PCM data and thermal maps follow a unique probability distribution separately. The generator learns the mapping method between these two distributions and transform the the input PCM data $\mathbf{z}$ to its corresponding thermal map denoted as $\mathbf{G}(\mathbf{z})$. The fake thermal map $\mathbf{G}(\mathbf{z})$ and the real ones $\mathbf{y}$ are then fed into the discriminator $\mathbf{D}$ alternatively together with its paired PCM data which serves as the condition input. For $\mathbf{G}(\mathbf{z})$, the PCM data concatenated to it is the input of $\mathbf{G}$ that was used to generate $\mathbf{G}(\mathbf{z})$. For $\mathbf{y}$, the PCM data collected in the same time instant is used as the condition input. The output of the discriminator, noted as $\mathbf{D}(\mathbf{z}, \mathbf{y})$ or $\mathbf{D}(\mathbf{z}, \mathbf{G}(\mathbf{z}))$ depending on whether real or fake thermal map was taken as input, is a real value indicating how confident the discriminator is toward the input being a correct thermal map conditioned on the given PCM data. The objective in the training of discriminator is therefore to maximize $\mathbf{D}(\mathbf{z}, \mathbf{y})$ and minimize $\mathbf{D}(\mathbf{z}, \mathbf{G}(\mathbf{z}))$ in term of expectations over the distributions of $\mathbf{y}$ and $\mathbf{z}$. Such objective function of discriminator can be mathematically expressed as following equation (2).

$$\max_{D}\{\mathbb{E}_{\mathbf{z},\mathbf{y}}[D(\mathbf{z},\mathbf{y})] - \mathbb{E}_{\mathbf{z}}[D(\mathbf{z},G(\mathbf{z}))]- \lambda_{gp}\mathbb{E}_{\hat{\mathbf{z}}}[(\|\nabla_{\hat{\mathbf{z}}}D(\hat{\mathbf{z}},\mathbf{z}))\|_2 - 1)^2]\} \tag{2}$$

$\mathbb{E}_{\mathbf{z},\mathbf{y}}$ and $\mathbb{E}_{\mathbf{z}}$ are the expectations over the distributions of $\mathbf{z}$ and $\mathbf{y}$. To maintain the 1-Lipschitz continuity of the discriminator, we adopt the gradient penalty from WGAN-GP [43]. $\hat{\mathbf{z}}$ is the interpolation between the fake and the real thermal map and λ_{gp} controls the weight of gradient penalty. The training target of the generator is to deceive the discriminator with generated thermal maps, so its objective is to maximize the expectation of $\mathbf{D}(\mathbf{z}, \mathbf{G}(\mathbf{z}))$. The objective function of the generator is defined in (3). Since the generator has no influence on the real thermal maps, the $\mathbf{D}(\mathbf{z}, \mathbf{y})$ term is omitted in the function.

$$\min_{G}\left\{\mathbb{E}_{\mathbf{z}}[-D(\mathbf{z},G(\mathbf{z}))] + \lambda_{L2}\cdot\mathbb{E}_{\mathbf{z},\mathbf{y}}[\|\mathbf{y}-G(\mathbf{z})\|_2]\right\} \tag{3}$$

In both (2) and (3), we use the Wasserstein Distance as the loss function to take its advantage of higher training stability and convergence possibility. The detailed architecture and parameters of the ThermGAN are shown in Table 2. We discard the random noise from the original GAN, as in our work, there are abundant PCM data in the training set which follow a certain distribution. This makes the PCM data itself can be seen as random noise just as the original $\mathbf{z}$ vector does. The PCM data given to the generator is first passed through a fully connected layer and reshaped to a square array. Then it is upsampled through 6 transposed convolutional layers and outputted as a 256×256 fake thermal map. All thermal maps are originally 185×154 in dimensions, however, for the convenience of being handled by the discriminator, they are expanded to 256×256 by equally padding zero values in every dimension. The discriminator is a conventional convolutional classifier with only one neuron as output and, to utilize the Wasserstein distance, no activation function is applied to it.

Table 2: ThermGAN parameters used in this work

Generator				Discriminator			
Layer	Kernel	#Output	Activation	Layer	Kernel	#Output	Activation
FC	-	8192	LReLU	Conv	5x5	128×128×64	ReLU
Reshape	-	4×4×512	-	Conv	5x5	64×64×128	ReLU
Conv_trans	5x5	8×8×512	LReLU	Conv	5x5	32×32×256	ReLU
Conv_trans	5x5	16×16×512	LReLU	Conv	5x5	16×16×512	ReLU
Conv_trans	5x5	32×32×256	LReLU	Conv	5x5	8×8×512	ReLU
Conv_trans	5x5	64×64×128	LReLU	Conv	5x5	4×4×512	ReLU
Conv_trans	5x5	128×128×64	LReLU	Conv	5x5	2×2×512	ReLU
Conv_trans	5x5	256×256×1	tanh	FC	-	512	ReLU
-	-	-	-	FC	-	1	None

4.3 Transient thermal map estimation

Traditionally, computing thermal information from power is time convolutional operation, which needs the historical data of power information. However, our thermal image generation problem from the utilization and on-chip sensor readings can be viewed as real-time inverse or fitting problem form those on-chip real-time information. Similar problem based on a limited on-chip sensor readings have been explored by many pre-silicon temperature estimation methods [21, 29, 30].

For our problem, the PCM metrics indeed consists of real-time temperature sensor information for each core and for the whole chip. Although the temperature at any time instant is determined by history thermo-information, such dependency is already decoupled by the temperature sensors which allows the thermal map be estimated As shown in the experimental section, ThermGAN can produce very accurate transient thermal map estimation, and outperforms the time-dependent LSTM model from [20] in terms of both accuracy and speed.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we present the experimental results showing both the speed and accuracy of our proposed ThermGAN model for PCM data to thermal map estimation.

We implement the whole network in Python 3.7 basing on TensorFlow(1.14.0) [44] which is a widely used open-source machine learning library. The model is trained for 10 epochs on a Linux server with 2 Xeon E5-2698v2 2.3GHz processors and Nvidia Titan X GTX GPU. The batch size is set to 8 and each data sample is a pair of synchronized PCM data and thermal map. We used 18 computationally intensive benchmarks from Phoronix benchmark suite [45] to collect the training data. As listed in Table 3, the benchmarks are split into three categories: processor, memory, and system. The variety of the benchmarks ensures the CPU is under different kinds of workloads, which further leads to the diversity of the training samples. For each workload, we keep the CPU running for 4 minutes and sampled the data at a frequency of 60Hz. In each time instant, both PCM data and the thermal map are captured simultaneously and saved in pair as one sample. We finally get 14300 samples for each benchmark and a total number of 257400 samples are collected in the training set.

The collected raw samples are preprocessed as described in Section 3. To better validate the performance of our trained model, we randomly pick 25% of the samples as the test set and only use the

Table 3: Benchmarks

Processor	Memory	System
AObench	PHPbench	T-test
Compress-7zip	Cyclictest	Cachebench
Encode-flac	Git	RAMspeed
Build-gcc	Mbw	Stream
Idle	Dbench	Aio-stress
-	Tinymem	Fio
-	-	Tiobench

remaining 75% for training. The learning rate and the decay parameters in the RMSProp optimizer are set to 0.0001 and 0.9. The weight of L2-norm λ_{L2} is set to 100 and λ_{gp} is set to 10. We ran the training for 10 epochs and the results reported in this section are based on the test set which was completely unseen by the model in the training process.

Fig. 4 visualizes the training process by showing the evolution of the output of the generator. We randomly picked one sample from the training set and show results in 5 epochs together with the ground truth. It can be clearly seen that the generated thermal map becomes closer to the ground truth as the training progresses.

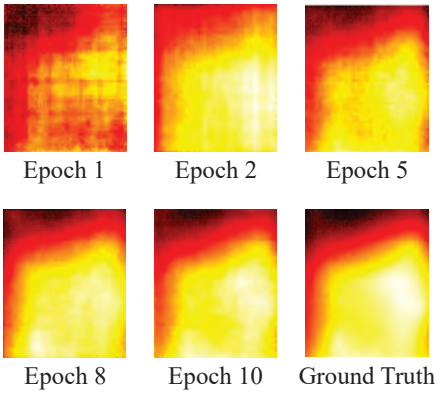


Figure 4: Evolution of one random sample as the training progresses.

5.1 Accuracy of Thermal Map Estimation

Once the ThermGAN is trained, the discriminator will be discarded and only generator is preserved. This model can take PCM data from any time instant as input and generates a real-like thermal map indicating the full-chip thermal distribution. To verify the performance of the model, we use the root-mean-square error (RMSE) given in (4) as the metric to indicate the difference between the generated and real thermal map (ground truth).

$$RMSE = \sqrt{\frac{\sum_{x=1}^{W}\sum_{y=1}^{H}(T(x,y)-T'(x,y))^2}{W \times H}} \tag{4}$$

where T and T' are the real and generated thermal map respectively. Both of them are images with only one channel which can easily suit in the equation as matrices. The vertical and horizontal dimensions of the thermal maps are $H = 185$ pixels and $W = 154$ pixels respectively. We evaluated our trained ThermGAN model on test set and the average RMSE across all 64350 samples in the test set is 0.47°C with a standard deviation of 0.56°C. In this work, the temperature in thermal maps of our test set ranges from 25 to 100°C. Comparing the absolute values of the error with this 75°C scale, the ThermGAN achieves an averaged full-scale estimation error of 0.63% and a standard deviation of 0.75%. This is a quite promising result since such resolution is accurate enough for thermal estimation applications. Fig. 8 illustrates the comparison between generated and ground truth thermal maps, which are randomly picked from the test set. The title of each thermal map indicates the benchmark it is from and the time instant in which it was collected. We show every thermal map in both 2D-image and 3D-plot with contour lines. As is shown in the figure, there are more spikes in the contour lines of the generated thermal map which indicates more noises, but the overall thermal distribution pattern is indistinguishable. The bottom row of Fig. 8 illustrates the error maps which is defined as the pixel-to-pixel difference between the real and fake thermal maps. Most of the errors are within 0.5°C except for only a few points, but still in acceptable range which is less than 1.5°C.

5.2 Real case study

The proposed ThermGAN is aimed at online estimation of full-chip transient thermal distribution. To evaluate the model in real application, we run the test on another benchmark named "Gimp". It is also from the Phoronix benchmark suite and is an open-source image manipulation program which keeps the chip at intensive workload. This benchmark was kept unseen throughout the training process and has completely no overlap with the benchmarks in the training set. We run the "Gimp" work load on i7-8650U processor for 2 minutes while the PCM data are collected at the frequency of 60Hz and fed into the ThermGAN for inference. The IR camera is simultaneously capturing real thermal maps of the chip which are used as ground truth to verify the ThermGAN inference results. A total number of 7200 samples are collected and ThermGAN achieves an average RMSE of 0.83°C with a standard deviation of 0.52°C. The error increases 0.39°C comparing to the result we get on the test set, which is actually a reasonable result as the distribution of data points in real cases may vary a lot from that of the training set. Despite the degradation of accuracy, the RMSE is still within 1°C and the averaged full-scale error is only 1.1% which is far beyond enough for full-chip thermal estimation in real applications. Some of the the results are detailed in Fig. 5. We pick 3 time instants (883, 4260 and 6903) and compare the estimated thermal map with its ground truth. We also fix a point on the upper right section of the chip and plot the time series temperature prediction for this position.

5.3 Speed of inference

The training process of the ThermGAN was time-consuming and cost more than 12 hours to converge. However, once the model is trained, it only reserves the generator part which is much lighter and can be embedded into the CPU to perform the real-time thermal map estimation. In our test, the time cost for each inference (one estimation of the whole chip thermal distribution based on the PCM data acquired in real-time) has a mean of *7ms* and a maximum of *7.5ms*, which translates to an inference frequency faster

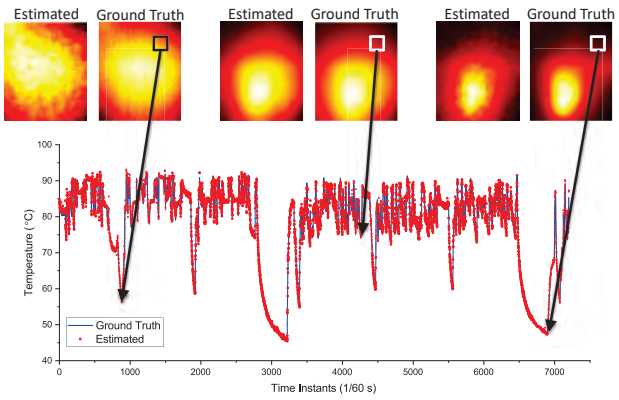


Figure 5: Comparison between estimated thermal distribution and ground truth on "Gimp" benchmark.

than 140 thermal maps per second. Such performance further verifies that our ThermGAN model is capable of real-time thermal estimation. The inference may introduce some overhead into the CPU computation, but doing temperature estimation at intervals of several seconds is fast enough to meet the need for real applications such as CPU task arrangement control.

We further study the modeling efficiency by comparing with the off-the-shelf FEM tool. Since we start with PCM parameters as inputs, we can't use traditional thermal simulator like HotSpot [10]. Instead, we first obtain the power map from the measured thermal map via 2D spatial Laplace operation and subsequent scaling based on total power [37]. We then use COMSOL to model the setup for multi-core processors and use power map obtained as the inputs. The FEM simulation was conducted on the same server as the GAN model and it takes 3 seconds on average for each thermal map generation. As a result, our study shows that the proposed ThermGAN model can achieve ~240X speedup over FEM method with similar accuracy as shown in Fig. 6. We remark that much faster numerical thermal analysis (than FEM) methods also exist. But the absolute speedup is less important than the millisecond performance we achieved in this work.

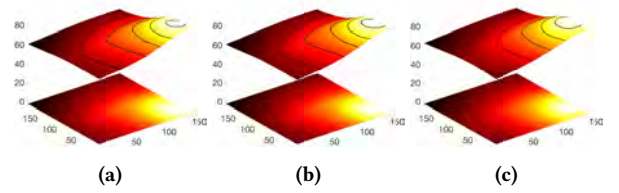


Figure 6: (a) Ground truth and estimated thermal map using (b) ThermGAN and (c) COMSOL FEM simulation.

5.4 Metrics in PCM that really matters

As is detailed in Table 1, we utilized all 170 PCM metrics as the input of the ThermGAN, which is actually an overkill since not all metrics are necessarily relevant to thermal estimation. For all 170 metrics, only 9 of them, i.e. temperature sensor readings of 8 cores and 1 socket, are directly related to the thermal information. For the rest 161 metrics, it is hard to tell which of them are more correlated with CPU thermal performance and which are of less importance. In this work, we leave this question to the model itself as the training process will automatically assign heavier weights to thermo-relevant metrics. When doing inference, the irrelevant metrics will contribute less influence on the accuracy of estimated thermal map.

To verify this and identify the thermo-relevant metrics, The following PCM masking test is conducted using the trained ThermGAN model. For each PCM vector, we mask only one entree at a time which is corresponding to the metric of interest. Thus, the input dimension remains unchanged and the trained ThermGAN model can still be applied to it while the masked metric will not participate in the feed forward calculation. In this way, we can observe how much the output accuracy is influenced by the masked metric with all the rest 169 metrics remain unchanged. The RMSE of the generated thermal map against the ground truth is calculated in the same way introduced in Section 5.1. For each input, the mask slides through all 170 entrees and results in 170 thermal maps with each corresponding to a masked PCM metric. We ran the masking test on the test set and the average RMSEs for all masked metrics are shown in Fig. 7.

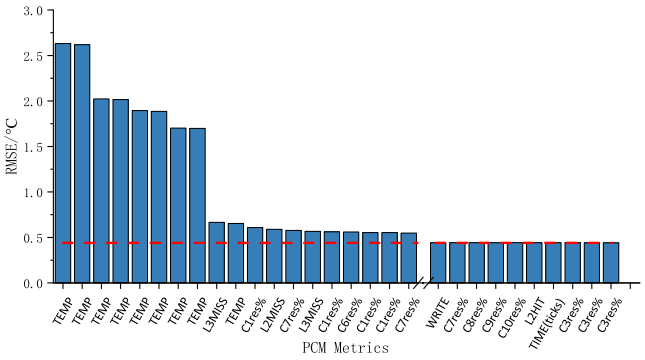


Figure 7: RMSE distribution across 170 masked PCM metrics.

The red line represents the average RMSE of thermal maps generated using unmasked PCM data. Masking different metrics results in vastly various accuracy degradation. The importance of each metric is proportional to the increment it introduces to the output RMSE. The top 8 errors are all caused by masking core temperature sensor readings. Each of them leads to an accuracy loss of more than 1.3°C. Masking the socket temperature metric caused an error of 0.65°C, which is not as much as the core temperatures but still among the top 10 metrics. Such observation is within our expectation but they are obviously not the only factors causing the accuracy degradation. The L3MISS is influencing the accuracy even more than the socket temperature. For the rest 160 metrics, 70 of them caused more than 5% degradation in accuracy compared to the baseline 0.44°C, and among which 33 metrics led to >10% accuracy loss. We refer to these top 80 metrics as thermo-relevant metrics, and the rest 90 metrics are playing a relatively small(<5%) or even negligible role in the estimation which implies that they are not thermo-relevant metrics. Apart from the temperatures, most of the thermo-relevant metrics are related with C-state

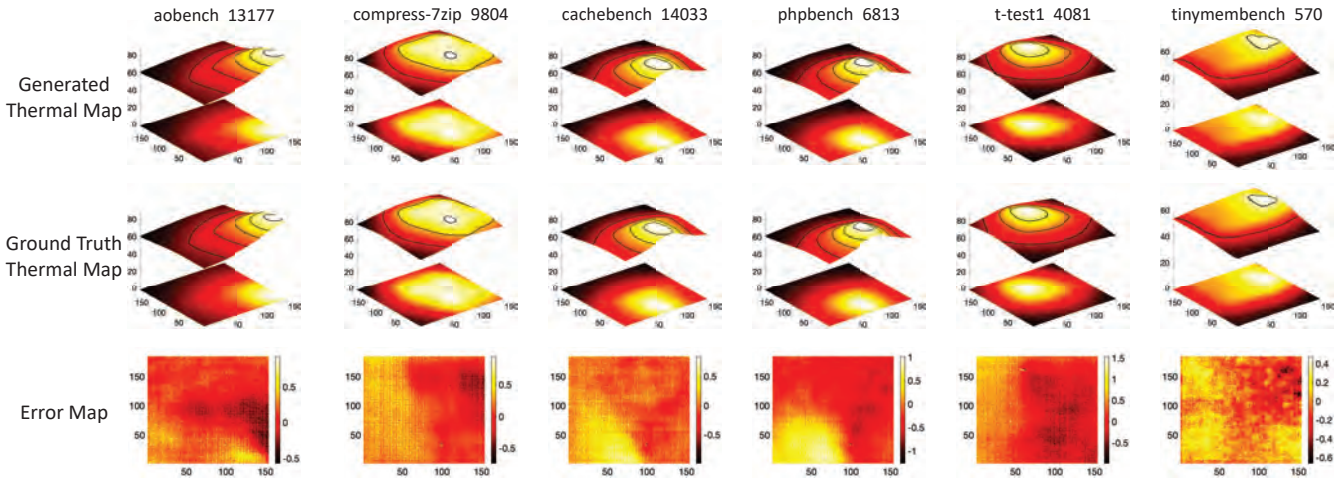


Figure 8: Comparison between generated and ground truth thermal maps.

which reflects the idle power saving information per core. The other thermo-relevant metrics consists of frequencies, L3 caches, instructions per cycle and so on.

5.5 Comparisons with state of the arts

In this subsection, we compare ThermGAN with a recently proposed post-silicon full-chip thermal estimation methods [20] and the pre-silicon estimation method [21].

Work in [20] is a machine-learning based model aimed at full-chip thermal estimation using PCM data. It employed LongShort-Term-Memory (LSTM) as its backbone and is implemented on the dual-core i5-3337U which has only 80 PCM metrics as input. To conduct a fair comparison, we increased the number of units in both its input and first layers to 170 to accommodate to the 170 PCM metrics of i7-8650U. The same dataset introduced in Sec 3 was used for both training and testing.

The average RMSE across all testing workloads is 1.84°C and the standard deviation is 1.11°C. In contrast, the proposed ThermGan which yields an average RMSE of 0.47°C and standard deviation of 0.56°C respectively as previously mentioned in Sec 5.1. Further more, the computational cost for each inference is ~17ms which is also slower than the ~7ms inference time yields by ThermGan.

Since there is no other research on post-silicon thermal estimation other than [20], we further compare our method with the state-of-the-art pre-silicon method known as "Eigenmaps" proposed in [21]. We note that this is not an apples-to-apples comparison as the "Eigenmaps" method requires optimized sensor locations in the chip design process. For commercial off-the-shelf microprocessors, both number and locations of the temperature sensors are fixed and may not meet the requirements of "Eigenmaps" method. However, in this comparison research, we assume such optimizations are done and allows the "Eigenmaps" method get the temperatures from the measured thermal maps instead of the physical sensors. The locations where the temperatures are sampled can be seen as virtual sensors which are optimized according to the algorithms in [21]. To make a fair comparison, the number of virtual sensors is set to , i.e. one for each of the 4 physical cores and one for socket. We ran "Eigenmaps" method on the test set and the average RMSE of estimated thermal maps is 0.94°C with a standard deviation of 0.45°C. It is slightly better than [20] but still worse than the proposed ThermGAN method. In therms of the overhead in real-time thermal estimation, "Eigenmaps" method requires to pre-calculate and save a dense matrix with 811680100 single-precision floating point entries which translates to 3.25GB in memory. This is quite expensive and is therefore not suitable for real-time application.

6 CONCLUSION

In this paper, we have proposed a new data-driven full-chip transient thermal map estimation method for commercial multi-core microprocessors based on the generative adversarial learning method. The proposed method, named *ThermGAN*, only uses the existing embedded temperature sensors and system level utilization information, which are available in real-time. Consequently, the methods presented in this work can be implemented by either the original chip manufacturer or a third party alike. In our approach, we treat this traditional thermal modeling problem as the image generation based on the customized conditional generative adversarial networks. The resulting *ThermGAN* can provide tool-accurate full-chip *transient* thermal maps from the given performance monitor traces of commercial off-the-shelf multi-core processors. Experimental results have showed the trained model is very accurate in thermal estimation with an average RMSE of 0.47°C, namely, 0.63% of the full-scale error. Our data further show that the speed of the model is faster than *7.5ms* per inference, which is two orders of magnitude faster than the traditional finite element based thermal analysis. Furthermore, the new method is ~4x more accurate than recently proposed LSTM-based thermal map estimation method and has faster inference speed. It also achieves ~2x accuracy with much less computational cost than a state-of-the-art pre-silicon based estimation method.

REFERENCES

[1] M. Taylor, "A landscape of the new dark silicon design regime," *IEEE/ACM International Symposium on Microarchitecture*, vol. 33, no. 5, pp. 8–19, October 2013.
[2] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling," *IEEE Trans. on Computers*,

vol. 63, no. 2, pp. 349–360, February 2014.
[3] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, "Task migrations for distributed thermal management considering transient effects," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp. 397–401, 2015.
[4] H. Wang, J. Ma, S. X.-D. Tan, C. Zhang, H. Tang, K. Huang, and Z. Zhang, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. on Design Automation of Electronics Systems*, vol. 22, no. 1, pp. 1:1–1:21, July 2016.
[5] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, 2003, pp. 2–13.
[6] J. Kong, S. W. Chung, and K. Skadron, "Recent thermal management techniques for microprocessors," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 13:1–13:42, jun 2012. [Online]. Available: http://doi.acm.org/10.1145/2187671.2187675
[7] H. Amrouch and J. Henkel, "Lucid infrared thermography of thermally-constrained processors," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, July 2015, pp. 347–352.
[8] F. Beneventi, A. Bartolini, P. Vivet, and L. Benini, "Thermal analysis and interpolation techniques for a logic + wideio stacked dram test chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 4, pp. 623–636, April 2016.
[9] M. Pedram and S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: Principles and methods," *Proc. of the IEEE*, vol. 94, no. 8, pp. 1487–1501, Aug. 2006.
[10] 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. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.
[11] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated space and time adaptive chip-package thermal analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 86–99, 2007.
[12] H. Wang, S. X.-D. Tan, G. Liao, R. Quintanilla, and A. Gupta, "Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2011.
[13] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "Efficient power modeling and software thermal sensing for runtime temperature monitoring," *ACM Trans. on Design Automation of Electronics Systems*, vol. 12, no. 3, pp. 1–29, 2007.
[14] K. Dev, A. N. Nowroz, and S. Reda, "Power mapping and modeling of multi-core processors," in *International Symposium on Low Power Electronics and Design (ISLPED)*, Sept 2013, pp. 39–44.
[15] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman, "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 2, pp. 405–419, Feb 2018.
[16] Intel, "Intel Performance Counter Monitor (PCM)," https://software.intel.com/en-us/articles/intel-performance-counter-monitor.
[17] AMD, "AMD uProf," https://developer.amd.com/amd-uprof/.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[19] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.
[20] 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. Asia South Pacific Design Automation Conf. (ASPDAC)*, Jan. 2020.
[21] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli, "Eigenmaps: Algorithms for optimal thermal maps extraction and sensor placement on multicore processors," in *Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 636–641. [Online]. Available: http://doi.acm.org/10.1145/2228360.2228475
[22] S. P. Gurrum, Y. K. Joshi, W. P. King, K. Ramakrishna, and M. Gall, "A compact approach to on-chip interconnect heat conduction modeling using the finite element method," *Journal of Electronic Packaging*, vol. 130, pp. 031 001.1–031 001.8, September 2008.
[23] Y. C. Gerstenmaier and G. Wachutka, "Rigorous model and network for transient thermal problems," *Microelectronics Journal*, vol. 33, pp. 719–725, September 2002.
[24] D. Li, S. X.-D. Tan, E. H. Pacheco, and M. Tirumala, "Parameterized architecture-level dynamic thermal models for multicore microprocessors," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 15, no. 2, pp. 1–22, 2010.
[25] T. Eguia, S. X.-D. Tan, R. Shen, D. Li, E. H. Pacheco, M. Tirumala, and L. Wang, "General parameterized thermal modeling for high-performance microprocessor design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2011.
[26] Z. Liu, S. X.-D. Tan, H. Wang, Y. Hua, and A. Gupta, "Compact thermal modeling for packaged microprocessor design with practical power maps," *Integration, the VLSI Journal*, vol. 47, no. 1, January 2014, in press, online access: http://www.sciencedirect.com/science/article/pii/S0167926013000412.
[27] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang, *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.
[28] R. Cochran and S. Reda, "Spectral techniques for high-resolution thermal characterization with limited sensor data," in *Proc. Design Automation Conf. (DAC)*, 2009, pp. 478–483.
[29] A. Nowroz, R. Cochran, and S. Reda, "Thermal monitoring of real processors: Techniques for sensor allocation and full characterization," in *Proc. Design Automation Conf. (DAC)*, 2010.
[30] S. Reda, R. Cochran, and A. N. Nowroz, "Improved thermal tracking for processors using hard and soft sensor allocation techniques," *IEEE Transactions on Computers*, vol. 60, no. 6, pp. 841–851, June 2011.
[31] Y. Zhang, B. Shi, and A. Srivastava, "Statistical framework for designing on-chip thermal sensing infrastructure in nanoscale systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 270–279, 2014.
[32] X. Li, X. Li, W. Jiang, and W. Zhou, "Optimising thermal sensor placement and thermal maps reconstruction for microprocessors using simulated annealing algorithm based on pca," *IET Circuits, Devices Systems*, vol. 10, no. 6, pp. 463–472, 2016.
[33] Yufu Zhang, A. Srivastava, and M. Zahran, "Chip level thermal profile estimation using on-chip temperature sensors," in *2008 IEEE International Conference on Computer Design*, 2008, pp. 432–437.
[34] A. Ziabari, J. Park, E. K. Ardestani, J. Renau, S. Kang, and A. Shakouri, "Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 11, pp. 2366–2379, 2014.
[35] J. L. Greathouse and G. H. Loh, "Machine learning for performance and power modeling of heterogeneous systems," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–6.
[36] R. G. Kim, J. R. Doppa, and P. P. Pande, "Machine learning for design space exploration and optimization of manycore systems," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–6.
[37] S. Sadiqbatcha, H. Zhao, H. Amrouch, J. Henkel, and S. X.-D. Tan, "Hot spot identification and system parameterized thermal modeling for multi-core processors through infrared thermal imaging," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2019.
[38] J. Liu, Y. Ding, J. Yang, U. Schlichtmann, and Y. Shi, "Generative adversarial network based scalable on-chip noise sensor placement," in *2017 30th IEEE International System-on-Chip Conference (SOCC)*. IEEE, 2017, pp. 239–242.
[39] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "Lithogan: End-to-end lithography modeling with generative adversarial networks," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 107:1–107:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317852
[40] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, and D. Z. Pan, "Gan-sraf: Sub-resolution assist feature generation using conditional generative adversarial networks," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 149:1–149:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317832
[41] B. Xu, Y. Lin, X. Tang, S. Li, L. Shen, N. Sun, and D. Z. Pan, "Wellgan: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 66:1–66:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317930
[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
[43] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv e-prints*, p. arXiv:1701.07875, Jan 2017.
[44] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi
[45] Phoronix, "Open-Source, Automated Benchmarking," https://www.phoronix-test-suite.com/.