Learning Based Spatial Power Characterization and Full-Chip Power Estimation for Commercial TPUs

Jincong Lu[1], Jinwei Zhang[1], Wentian Jin[1], Sachin Sachdeva[1], Sheldon X.-D. Tan[1]
[1]Department of Electrical and Computer Engineering
University of California, Riverside
Riverside, California, United States
jincong.lu@email.ucr.edu,jzhan319@ucr.edu,wjin018@ucr.edu,ssach008@ucr.edu,stan@ece.ucr.edu

ABSTRACT

In this paper, we propose a novel approach for the real-time estimation of chip-level spatial power maps for commercial Google Coral M.2 TPU chips based on a machine-learning technique for the first time. The new method can enable the development of more robust runtime power and thermal control schemes to take advantage of spatial power information such as hot spots that are otherwise not available. Different from the existing commercial multi-core processors in which real-time performance-related utilization information is available, the TPU from Google does not have such information. To mitigate this problem, we propose to use features that are related to the workloads of running different deep neural networks (DNN) such as the hyperparameters of DNN and TPU resource information generated by the TPU compiler. The new approach involves the offline acquisition of accurate spatial and temporal temperature maps captured from an external infrared thermal imaging camera under nominal working conditions of a chip. To build the dynamic power density map model, we apply generative adversarial networks (GAN) based on the workload-related features. Our study shows that the estimated total powers match the manufacturer's total power measurements extremely well. Experimental results further show that the predictions of power maps are quite accurate, with the RMSE of only 4.98mW/mm^2, or 2.6% of the full-scale error. The speed of deploying the proposed approach on an Intel Core i7-10710U is as fast as 6.9ms, which is suitable for real-time estimation.

ACM Reference Format:
Jincong Lu[1], Jinwei Zhang[1], Wentian Jin[1], Sachin Sachdeva[1], Sheldon X.-D. Tan[1]. 2023. Learning Based Spatial Power Characterization and Full-Chip Power Estimation for Commercial TPUs. In *28th Asia and South Pacific Design Automation Conference (ASPDAC '23), January 16–19, 2023, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3566097.3568347

1 INTRODUCTION

With the continuing trend of rapid integration and technology scaling, today's high-performance processors have become more thermally constrained than ever before. An increase in temperature has been shown to exponentially degrade the reliability of the semiconductor chips [1], and hence has become one of the leading concerns in the industry today. To address this trend, runtime power and thermal control schemes are being implemented in most, if not all new generations of processors and are crucial in any modern processor [2, 3]. However, these control schemes require accurate real-time thermal information, and essentially the power information, ideally the spatial power density map of the entire chip area, in order to be effective [4, 5]. On-chip temperature sensors alone cannot provide the full-chip temperature information since the number of sensors that are typically available is very limited due to their high area and power overheads [6]. Furthermore, power characterization for commercial tensor processors (TPUs) is rarely studied and reported.

To obtain precise thermal and power control, we need to look at two important aspects of this problem: the accurate estimation of the on-chip power and the accurate calculation of temperature from the thermal model and the on-chip power inputs. Traditional power estimation methods focus on the functional unit (component-wise or core-wise) power estimation based on the measured temperature and total power [7–10]. But those methods require an understanding of the architectural details and functional units of each chip and many approaches are still ad-hoc, involving manual turning. At the same time, post-silicon (no prior layout information is needed) spatial power map estimation from thermal information has been widely studied. This problem was also coined as the *inverse thermal map to power map* problem as the temperature can be more easily measured either directly or indirectly. Many approaches have been investigated in the past [11–16]. Most of the proposed methods tried to frame the problem as a nonlinear optimization problem (deterministically or statistically) once the thermal models are known. However, those methods do not work for general off-the-shelf commercial processors where only core-level power can be obtained [16]. Many of those methods only work for specialized silicon such as FPGAs [12–14, 16]. Recently, new spatial power map estimation methods based on the measured spatial temperature, 2D spatial Laplace transformation, and processor's performance monitors were proposed for general commercial multicore processors [17]. Specifically, the machine-learning based power source hot spot estimation [6] and full chip thermal map estimation [18] have been proposed. Those methods estimate the hot spot or the full chip thermal maps based on the real-time on-chip performance information such as Intel's Performance Counting Monitor (IPCM) [19]. But these methods can hardly be applied to TPUs (like the Google Coral M.2 TPU used in this paper) as there is no real-time utilization information such as IPCM from the TPU chips. As a result, the existing full-chip power map estimation methods cannot be applied to commercial TPUs.

In this work, we try to address the aforementioned issues and propose a novel machine-learning based approach to estimate the

Jincong Lu and Jinwei Zhang are both the first authors and have equal contributions to this work. This work is supported in part by an NSF grant under No. CCF-2007135 and in part by NSF grants under No. CCF-2113928 and No. OISE-1854276.

ASP-DAC '23, Janurary 16–19, 2023, Tokyo, Japan

ACM ISBN 978-1-4503-9783-4/23/01.
https://doi.org/10.1145/3566097.3568347

full-chip power density distribution of commercial TPU chips. The key contributions are as follows:

- We developed a generalized full-chip power map estimation method that is based on the hyperparameters of the TPU's workloads (i.e., neural networks inferencing on the TPU), without the knowledge of TPU's performance monitors or supply power.
- We treat the full-chip power density map estimation problem as an image generation problem, where the input features are given number of hyperparameters and TPU resource information (generated by the TPU compiler). We propose to use the Conditional Generative Adversarial Networks (CGAN) to generate such power map images from the given features.
- Experimental results show that the predictions of power maps are quite accurate, with the RMSE of only 4.98mW/mm^2, or 2.6% of the full-scale error. The speed of deploying the proposed approach on an Intel Core i7-8650U is as fast as 6.9ms, which is suitable for real-time estimation.

This article is organized as follows. Section 2 shows the power modeling framework and IR thermography setup used in this study. Section 3 models the spatial power from the workload features that are available in real time. Section 4 describes the architecture of the proposed CGAN-based neural net model for power map estimation. Section 5 presents the experimental results and comparisons and Section 6 concludes this article.

2 POWER MAP ESTIMATION FRAMEWORK

A brief overview of the proposed approach will be presented in this section, along with a description of the thermal setup used for collecting the necessary data from a commercial off-the-shelf TPU chip while it is under the workload.

2.1 Estimation flow overview

The proposed approach involves three engineering phases. First, we obtain full-chip power map measurements across the TPU with both high accuracy and resolution by implementing a state-of-the-art thermal-to-power technique. Second, we propose to take advantages of the hyperparameters of the NN workloads that inferences on the TPU as the model's input features, and the outputs are power maps across the TPU immediately. Last but not least, the new model employs a special Generative Adversarial Network architecture called Conditional GAN or CGAN to train the online power characterization model.

The CGAN-based power model requires two chunks of data for the training procedure, one is the off-line measured power maps when TPU is under load, which are used as targets when training the model. The other is a set of hyperparameters extracted from the NN workloads to be executed on TPU. It should be noted that those hyperparameters can be extracted either online or prior to the workload execution. Once the model is trained, we can use it for online TPU power inferencing. Fig. 1 illustrates the framework and data acquisition flow of the proposed approach. The first and the second phase, including every step shown in Fig. 1 will be described in detail in the next section. The third phase will be explained in Section 4.

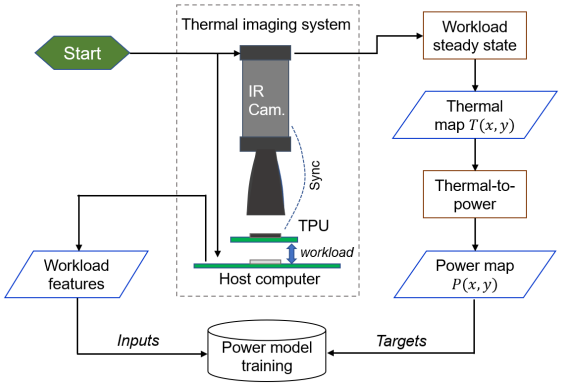
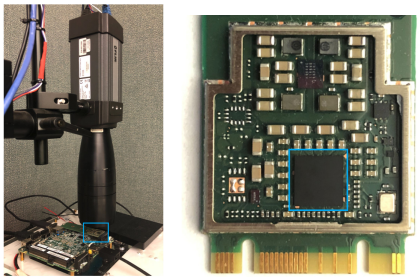


Figure 1: Framework and data acquisition flow



(a) (b)

Figure 2: (a) Thermal Imaging system setup (b) TPU chip under-test, Coral M.2 TPU module. TPU module is shown in the blue box.

2.2 Thermal IR imaging system

The proposed machine-learning based approach relies on proper data acquisition of the chip-level spatial power information from the TPU under workloads for the model's training and testing procedure. Directly measuring the power maps of TPU chips is not achievable. To address this issue, we indirectly measure the full-chip power map through a thermal-to-power approach proposed in [20]. This thermal-to-power approach has both high precision and resolution. It calculates the full-chip spatial power density maps from the spatial temperature maps measured when the processor is under a thermal steady state. The approach basically takes advantage of the high correlation between heat source distribution and the 2D Laplacian transformation of the temperature of the chip surface.

The thermal-to-power approach requires accurate measurements of chip surface temperature maps. Hence, we have a built-in advanced infrared (IR) thermal imaging system, as shown in Fig. 2. In order to expose the TPU chip to the IR camera, we have removed the stainless steel cover on top of the TPU. It should be noted that in our test case, the TPU module requires no external cooling kits, such as heat sinks, to ensure proper thermal conditions for the TPU to work. However, for some other TPU modules that come with heat sinks, after removing the heat sinks, the TPU module can be cooled with a widely used back-side liquid cooling technique [21] to ensure proper thermal operating conditions. The back-side liquid

cooling approach features a thermoelectric (Peltier) device mounted on the PCB directly beneath the processor module allowing it to be cooled from underneath. This leaves the front side of the processor fully exposed to the IR camera without any interference layer in between.

Product information and specs of our IR thermal imaging system are described as follows. The IR camera used in this setup is a FLIR A325sc which supports a maximum imaging resolution of 320×240 pixels (px) with 16-bits of precision per px, and a maximum capturing frequency of 60Hz. The IR sensor is factory calibrated for accuracy across the temperature range of −20°C to 120°C, and resolves the IR spectral range of 7.5μm to 13μm. A high-resolution microscope lens is used to achieve the spatial resolution of 25μm per px.

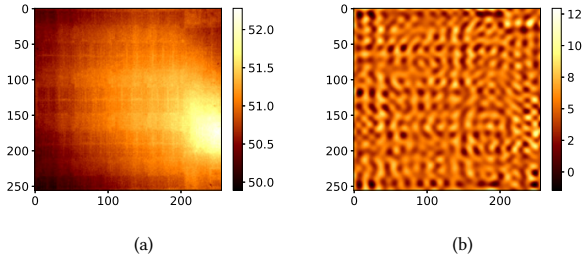


Figure 3: (a) Thermal image and (b) power map of TPU for MobileNet-V2-224-1.0 network.

3 DATA PREPARATION AND FEATURE SELECTION

In this work, we model the spatial power from the workload features that are available in real time. Like any other regression model, the machine-learning model architecture we deploy is a supervised learning model, for which the proper data set is ultimately important. As previously mentioned, the data required for training the learning-based model involves measuring the offline power maps across the TPU full-chip and collecting the hyperparameters of NN workloads running on the TPU module. It should be marked that each individual workload has a unique hyperparameters-powermap data pair, which serves as a unique training data point. Google Coral Edge TPU has 3 different frequencies, 500MHz, 250MHz, and 125MHz. Our workload mainly runs in 500MHz. In this section, the detailed process of acquiring the necessary data is presented.

3.1 Offline power map acquisition

There have been various post-silicon approaches transforming thermal distribution to power distribution [11–15, 20, 22]. Among those [6, 20] suits for our study case best, giving it calculates spatially continuous and relatively precise power maps from thermal maps with high efficiency, which is suitable for real-time inferences.

Considering the steady state 2D spatial thermal distribution of the processor as $T(x, y)$, where (x, y) is the coordinates of the thermal map. Power map can be approximated as [20]:

$$p(x,y) \approx \begin{cases} k[-\nabla^2 T(x,y)], & -\nabla^2 T(x,y) > 0 \\ 0, & -\nabla^2 T(x,y) \le 0 \end{cases} \quad (1)$$

with

$$k = \kappa \Delta z \quad (2)$$

where $p(x, y)$ stands for the spatial power map (density, Watt/area), κ and Δz for thermal conductivity and chip thickness, which are constants. And $\nabla^2 T(x, y)$ is the 2D Laplacian of temperature. The coefficient k is expressed by:

$$k = \kappa \Delta z \approx \frac{P}{-\int_{S_P} \nabla^2 T(x,y) dx dy} \quad (3)$$

where S_P indicates the area where the negative-Laplacian term of temperature $[-\nabla^2 T(x, y)]$ is positive. The negative-Laplacian term reflects the pattern of spatial power distribution. In this work, we call k the thermal-to-power coefficient. It can be calculated by the thermal measurement of idle status $T_{idle}(x, y)$ combined with standby total power consumption P_{idle} provided by the official specification. Once we have $T_{idle}(x, y)$ and P_{idle}, we can substitute them for equation (3) to obtain k. After k is obtained, power maps under any workloads can be acquired straightforwardly through equation (1) from thermal measurements.

As we see, one special requirement of this method is that it needs the processor to be under thermal steady-states when calculating its power maps from thermal measurements. Hence, to satisfy this requirement, we have the TPU module run each workload for sufficiently long (e.g. 2 minutes per workload) to stabilize its temperature. Multiple thermal images are captured after the TPU reaches a steady state corresponding to that workload. Once the steady-state thermal images are obtained, they will be processed to calculate an averaged power map for that workload as the estimation target of that data point. It should be noted that the proposed learning-based power model does not require any waiting during deployment since the proposed model needs no thermal measurements for power estimations. As an example, Fig. 3(a) shows an averaged steady-state thermal image for MobileNet-V2-224-1.0 network, which is a widely used image-classification model available on TensorFlow [23]. Fig. 3(b) further illustrates the resulting TPU power map. In order to automate the measuring procedure, we arranged a sequence of workloads for the TPU module, and in the meantime, the IR camera is synchronized with the TPU for each workload's running period.

The size of the TPU module is about 5.06 × 4.94mm. And it only occupies partial camera's 320 × 240px field of view. We crop the chip area out of the photo and then calculate the power map. Thermal noise is a big problem when we need to calculate the Laplacian. Although the noise is small relative to the temperature, its Laplacian can be locally larger than the Laplacian of temperature, overshadowing useful information. An effective method to extract information is the discrete cosine transform (DCT) [24]. The majority of the information is contained in low-frequency coefficients of DCT. Therefore, we transform the heatmaps into the spatial frequency domain by 2D DCT, keeping the low-frequency coefficients, and then transform them back. This reduces some resolution but allows us to analyze the spatial distribution of power.

3.2 Feature selection considering TPU workloads

For neural networks such as TPU's workloads, power distribution is an immediate reflection of hardware resource utility invoked by the neural networks executing on the TPU. TPU hardware resources that the network demands are tightly related to the network model architecture, size, operations, etc. Hence, we are able to characterize TPU's power from the workloads' hyperparameters such as operation type, count and workload size, etc.

Table 1: Selected Workload Features (Coral M.2 TPU, Google Edge)

Overall			
image_shape	pooling_mode	onchip_mem_rem	num_op_tpu
width_multiplier	model_size	offchip_mem_used	num_op_cpu
depth_multiplier	onchip_mem_used	total_op_cnt	infer_time
Operational Statistics			
add	full_connect	pad	reduce_max
avg_pool_2d	l2_norm	quant	relu
concat	max_pool_2d	reshape	strslc
conv2d	mean	sft_max	hard_swish
depconv2d	mul	sub	

In this work, we divide the network's hyperparameters as features into two groups, called the *overall features* and *operational statistics*. Neural network models that are coded to run on CPU need to be compiled to a TPU readable version. In our study, EdgeTPU Compiler [25] is employed to transform a CPU version network model to a TPU version. On the one hand, the overall features such as model size and memory usage are recorded through the process. On the other hand, we collect statistical information for the type and count of operations indicated by the network in the meantime. We mark that for different TPUs different tools may be involved, however, those network information should always be reachable. Today's world has a vast number of neural networks and hundreds of kinds of operations. To find the most popular operational features of network models, we explored a number of the most popular and widely used open-source deep neural network models from TensorFlow. The selection of models will be explained in more detail in Section 5. Table 1 shows the 31 features selected for the network workloads.

4 CGAN-BASED ESTIMATION MODEL

4.1 Review of CGAN

As a machine-learning problem, our purpose is to generate the on-chip power image from the workload features. Generative Adversarial Network (GAN) can be a competitive choice [26].

GAN has two contrary networks, generator G and discriminator D. G is trying to map an input vector to an output image, while D attempts to tell if an image comes from the real data or G. They will be trained simultaneously and keep trying to optimize themselves to fool/expose others. At last, when the generated image is close enough to the ground truth, the generator should have become a mastered projector, and the discriminator can never tell the difference between a fake and a real image.

Original GAN is used to produce new images within the range of existing image distribution and the generator is fed by noise. As a variant, Conditional GAN (CGAN) also give some labels to the generator, so it can map features to corresponding images [27]. Based on CGAN, we no longer use random noise because we expect to give a unique power distribution with one certain feature vector.

Sometimes it can be tough to train the GAN model because of the gradient vanishing. We can introduce Wasserstein Distance instead of the conventional JS-Divergence to measure the similarity between the distributions of real and fake images [28]. This modification can stabilize the training process and reduce the frequency of collapses.

4.2 Proposed CGAN-based power estimation framework

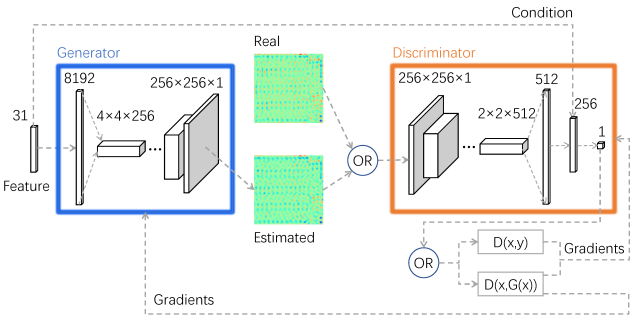


Figure 4: Architecture of CGAN model

The framework of our hyper-parameter to power map model is shown in Fig. 4. The input condition x is a 1x31 vector, which will be given to both the generator G and the discriminator D. The generator learns how to map it to the correct power map image y, and produce $G(x)$. Then D accepts y or $G(x)$ alternatively with the condition x, score a degree $D(x, y)$ or $D(x, G(x))$ how confident the given power map y or $G(x)$ is true. Our goal is to maximize $D(x, y)$ and minimize $D(x, G(x))$ over all (x, y) pairs in the training set. We can write down the objective function to minimize as:

$$\begin{aligned} loss_D = &\mathbb{E}_{(x,y)}[D(x, G(x)) - D(x, y)] + \\ &\lambda_{gp}\mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}} D(\hat{x}, x)||_2 - 1)^2] \end{aligned} \tag{4}$$

Here $\mathbb{E}_{(x,y)}$ is the expectations over the (x,y) pairs in the training set. Also, we introduce an extra gradient penalty term, so that the discriminator has the 1-Lipschitz continuity [28]. $\hat{x}$ is the interpolation between $G(x)$ and y, and λ_{gp} is the weight.

For the generator, we want to maximize $D(x, G(x))$ and minimize the L2 loss $||y - G(x)||^2$. The generator has nothing to do with the real power map y, so there is no $D(x, y)$ term. The loss function is:

$$loss_G = \mathbb{E}_{(x,y)}[-D(x, G(x)) + \lambda_{L2} \cdot ||y - G(x)||^2] \tag{5}$$

The architecture and parameters of the generator and discriminator networks are shown in Table 2.

First, the generator transforms the input condition vector into an image by a fully connected layer and a reshape operation. After that, there are 6 transposed convolutional layers to finally produce a 256×256px power map. The discriminator is a conventional convolutional classifier that has a similar but reversed structure with the generator, and goes from a 256×256px image to only one real number as the output.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we demonstrate the experimental results of the proposed approach in two folds. On the one hand, we convince that the power maps obtained through the thermal-to-power way are sufficiently reliable. On the other hand, we show that the online inferencing of power maps by the proposed CGAN-based model is computationally efficient and technically sound.

Table 2: Architecture and Parameters

Generator			
Layer	Kernel	#Output	Activation
FC	-	8192	Leaky ReLU
Reshape	-	4x4x512	-
Conv_trans	5x5	8x8x512	Leaky ReLU
Conv_trans	5x5	16x16x512	Leaky ReLU
Conv_trans	5x5	32x32x256	Leaky ReLU
Conv_trans	5x5	64x64x128	Leaky ReLU
Conv_trans	5x5	128x128x64	Leaky ReLU
Conv_trans	5x5	256x256x1	-
Discriminator			
Layer	Kernel	#Output	Activation
Conv	5x5	128x128x64	Leaky ReLU
Conv	5x5	64x64x128	Leaky ReLU
Conv	5x5	32x32x256	Leaky ReLU
Conv	5x5	16x16x512	Leaky ReLU
Conv	5x5	8x8x512	Leaky ReLU
Conv	5x5	4x4x512	Leaky ReLU
Conv	5x5	2x2x512	Leaky ReLU
FC	-	512	Leaky ReLU
(+Cond) FC	-	256	Leaky ReLU
FC	-	1	-

5.1 Validation of the total power consumption

As discussed in Section 3.1, directly measuring the spatial power distribution of the TPU is not realistic, we have implemented one of the state-of-the-art methods that compute spatial power maps from the thermal measurements. The question is that whether those inter-mediately obtained power maps are sufficiently reliable in our test case. Fortunately, Coral has open-sourced a few but limited data for total power measurements. Hence, we are able to compare our estimated total power with the manufacturer's provided total power measurements, to see how well they match. The more they match, the more convincing they are. To our knowledge, this indirect way is the best way for validating the power maps in our case.

The estimated total power is simply an integration from the estimated spatial power maps. Coral M.2 TPU module's official specification has released 7 power measurement data points, pertaining to two different workloads under three different operation frequencies, respectively, plus an idle power measurement. Power under idle status is officially indicated by a range between 0.375 and 0.400W. In our work, we take the central value 0.3875W as its golden idle power. Then combine that with the thermal maps captured under idle status to calculate the thermal-to-power coefficient k. Then we use it to calculate the power maps for the same two workloads at those three different frequencies, and further their total power.

By combining two models and three operating frequencies, Table 3 lists the six golden total power data points, which are provided from the official specification, with our estimated total power consumption. As we can see, all of the estimated total power data points mirror the real power measurements remarkably well. Given that one decimal point precision is available in official power data, the root-mean-squared-error of total power estimation is only 0.0147W, and the percentage error is within 2%.

5.2 Power map estimation accuracy

The dataset consists of a number of well-known neural network models for image recognition, such as EfficientNet, InceptionResNet, MobileNet, etc. By varying their architectural hyperparameters, many of their variants were generated and added to the dataset. The final dataset has 7066 data points (networks) in total, where 6359 points are randomly selected for training and 707 points for testing. All networks are executed with the TPU at the nominal frequency 500MHz.

Table 3: Total Power Comparison

Workloads	Total Power	500 MHz	250 MHz	125 MHz
MobileNet V2	Real	1.4 W	0.9 W	0.6 W
	Est.	1.42 W	0.92 W	0.60 W
Inception V3	Real	0.7 W	0.6 W	0.5 W
	Est.	0.69 W	0.58 W	0.50 W

After the training process, the generator of the CGAN model is able to estimate the power map with the input hyperparameters. To characterize its accuracy, we calculate the root-mean-squared error (RMSE) over each pixel between the generated power map and the measured power map.

In our dataset, the power density ranges from 0 to 189.34mW/mm^2. The averaged RMSE of the power map estimation on the test set is 4.98mW/mm^2 with a standard deviation of only 2.53mW/mm^2. The results are quite accurate considering the data range. Fig. 5 compares the estimated and the measured power maps with some examples from the test set. It should be noted that the right-most column shows the worst estimation on the test set, which is about 10% percentage error on the total power. It can be seen that the CGAN-based model has learned the contour of real power remarkably well.

The power map can also be used to calculate the total power by simply integrating the power density over the power map. The mean error of the total power is 0.0968W. Considering that the average total power is 1.375W on the test set, these estimations of total power are sufficiently accurate as well.

5.3 Computational efficiency

Training procedure normally takes a few to a dozen hours to complete. Once the generator is well-trained, it can be deployed for real-time power prediction. The average inference time we measured in our experiments is 6.9ms, with Intel Core i7-10710U as the host board and the Coral Edge TPU. This low latency ensures the effectiveness of real-time power estimation. On the one hand, most of the models on the TPU have a single inference time of well over 6.9ms (they take dozens or even hundreds of milliseconds). On the other hand, TPUs do not switch deployed neural networks frequently, and those applications themselves that switch the neural networks generally take more time. As a result, the proposed model is sufficiently rapid to keep up with the TPU.

6 CONCLUSION

In this article, we have proposed a machine-learning-based approach for real-time estimation of full-chip power maps for commercial Google Coral M.2 TPU chips for the first time. The new method focuses on the DNN inference applications on the TPU and apply workload-related features such as the hyperparameters of the DNN networks and TPU resource information generated by TPU compilers as the input of the deep neural network models. To build the dynamic power density map model, we applied generative adversarial networks (GAN) to model the power density

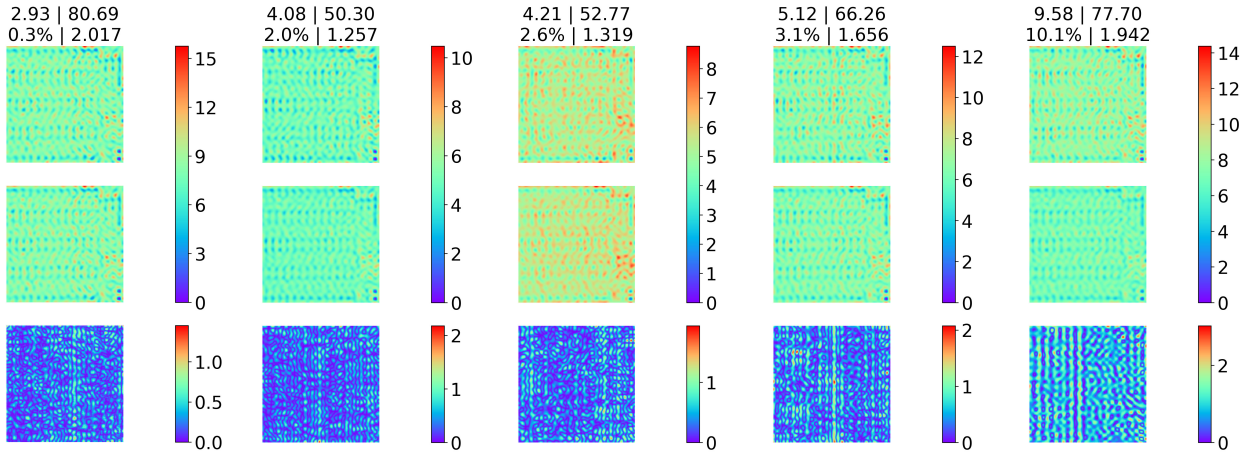


Figure 5: Measured power maps (row #1), estimated power maps (row #2), and error maps (row #3). The numbers in the first row indicate the *Power Density RMSE* | *Average Power Density* (unit: $\mathrm{mW/mm^2}$). And numbers in the second row indicate the *Total Power Percentage Error* | *Total Power* (unit: W).

map based on the selected workload-dependent features. Our study showed that the estimated total powers match the manufacturer's total power measurements extremely well. Experimental results further showed that the predictions of power maps are quite accurate, with the RMSE of only $4.98\mathrm{mW/mm^2}$, or 2.6% of the full-scale error. The speed of deploying the proposed approach on an Intel Core i7-10710U is as fast as 6.9ms, which is suitable for real-time estimation.

REFERENCES

[1] "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)," 2003. In International Sematech Technology Transfer Document 03024377A-TR, 2003.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Micro, IEEE*, vol. 32, pp. 122–134, May 2012.

[3] M. Taylor, "A landscape of the new dark silicon design regime," *IEEE/ACM International Symposium on Microarchitecture*, vol. 33, pp. 8–19, October 2013.

[4] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Intl. Symp. on Computer Architecture*, 2006.

[5] J. Kong, S. W. Chung, and K. Skadron, "Recent thermal management techniques for microprocessors," *ACM Comput. Surv.*, vol. 44, pp. 13:1–13:42, jun 2012.

[6] S. Sadiqbatcha, J. Zhang, H. Zhao, H. Amrouch, J. Hankel, and S. X.-D. Tan, "Post-silicon heat-source identification and machine-learning-based thermal modeling using infrared thermal imaging," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[7] R. Joseph and M. Martonosi, "Run-time power estimation in high-performance microprocessors," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pp. 135–140, 2001.

[8] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proceedings of MICRO*, 2003.

[9] 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.

[10] 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)*, pp. 39–44, Sept 2013.

[11] X. Wang, S. Farsiu, P. Milanfar, and A. Shakouri, "Power trace: An efficient method for extracting the power dissipation profile in an ic chip from its temperature map," *IEEE Transactions on Components and Packaging Technologies*, vol. 32, no. 2, pp. 309–316, 2009.

[12] R. Cochran, A. N. Nowroz, and S. Reda, "Post-silicon power characterization using thermal infrared emissions," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, (New York, NY, USA), pp. 331–336, ACM, 2010.

[13] S. Paek, W. Shin, J. Sim, and L. Kim, "Powerfield: A probabilistic approach for temperature-to-power conversion based on markov random field theory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1509–1519, 2013.

[14] A. Nowroz, G. Woods, and S. Reda, "Power mapping of integrated circuits using ac-based thermography," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, pp. 1398–1409, aug 2013.

[15] 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, 2016.

[16] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors Journal*, vol. 18, pp. 680–691, Jan 2018.

[17] J. Zhang, S. Sadiqbatcha, M. O'Dea, H. Amrouch, and S. X.-D. Tan, "Full-chip power density and thermal map characterization for commercial microprocessors under heat sink cooling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2021.

[18] S. Sadiqbatcha, J. Zhang, H. Amrouch, and S. X.-D. Tan, "Real-time full-chip thermal tracking: A post-silicon, machine learning perspective," *IEEE Transactions on Computers*, 2021.

[19] Intel, "Intel Performance Counter Monitor (PCM)." https://software.intel.com/en-us/articles/intel-performance-counter-monitor.

[20] J. Zhang, S. Sadiqbatcha, W. Jin, and S. X. . Tan, "Accurate power density map estimation for commercial multi-core microprocessors," in *2020 Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1085–1090, 2020.

[21] 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)*, pp. 347–352, July 2015.

[22] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 680–691, 2018.

[23] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[24] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, Jan 1974.

[25] "Edge TPU Compiler." Available from coral.ai/docs/edgetpu/compiler.

[26] 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.), pp. 2672–2680, Curran Associates, Inc., 2014.

[27] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv e-prints*, p. arXiv:1411.1784, Nov. 2014.

[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv e-prints*, p. arXiv:1701.07875, Dec. 2017.