Real-time Thermal Map Estimation for AMD Multi-Core CPUs using Transformer

Jincong Lu, Jinwei Zhang and Sheldon X.-D. Tan
Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA
jincong.lu@email.ucr.edu, jzhan319@ucr.edu, stan@ece.ucr.edu

2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD) | 979-8-3503-2225-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCAD57390.2023.10323817

***Abstract*—This paper presents a novel approach for real-time estimation of spatial thermal maps for the commercial AMD Ryzen 7 4800U 8-core microprocessor using a transformer-based machine learning method. The proposed method, called *ThermTransformer*, leverages real-time performance metrics of the AMD chip, provided by *uProf 4.0*, to accurately estimate transient thermal maps. These maps can be valuable for dynamic thermal, power, and reliability controls requiring higher accuracy. Unlike traditional Convolutional Neural Networks (CNN) designed for image data or Recurrent Neural Networks (RNN) suitable for transient data, *ThermTransformer* is based on a modified self-attention architecture. It takes time-series performance metrics information as input and directly generates transient thermal images. Our results demonstrate that this transformer-based method achieves the best of both worlds – surpassing CNN in prediction quality and performing well for transient data. Experimental results reveal that *ThermTransformer* achieves highly accurate predictions of power maps, with an RMSE of only $0.36°\mathrm{C}$ or 0.8% of the full-scale error. Additionally, it outperforms the recently proposed GAN-based thermal map estimation method, *ThermGAN*, by 1.66x and the LSTM-based thermal prediction method, *RealMaps*, by 6.09x in terms of accuracy on average. Furthermore, the proposed approach can be efficiently deployed on the target chip, providing real-time estimation with a speed as fast as $14\mathrm{ms}$.**

I. Introduction

With the ongoing trend of rapid integration and technology scaling, contemporary high-performance processors are facing more pronounced thermal limitations than ever before. Research has shown that higher temperatures exponentially degrade the reliability of semiconductor chips [1], making it a significant concern in the industry. To address to this trend, runtime power and thermal control schemes are being implemented in most, if not all, new generations of processors. These control schemes play a crucial role in modern processors [2], [3]. However, for these control schemes to be effective, they require accurate real-time thermal information, ideally a spatial thermal map of the entire chip area [4], [5]. On-chip temperature sensors alone cannot provide comprehensive chip-wide temperature information due to their limited number, primarily constrained by the area and power overheads [6].

Several existing methods rely on on-chip temperature sensors. However, the availability of physical sensors is typically limited, and their placement may not accurately capture the true hotspots on the chip. As a result, temperature regulation decisions based solely on these sensors can be misleading [7]. Consequently, a more popular approach is to augment the data from the few on-chip sensors with estimated temperatures of prominent hotspots using thermal models based on estimated power traces [8]. These methods provide higher spatial resolution by enabling real-time monitoring of temperatures for all hotspots on the chip [9]–[12].

However, existing thermal modeling methods still possess certain limitations. Firstly, these methods rely on accurate power traces as inputs. However, accurately estimating the power consumption of each functional unit (FU) in a real processor under varying workloads is a non-trivial and often infeasible task [13], [14]. Secondly, calibrating these models for practical use presents challenges due to simplified modeling, boundary conditions, and limited accuracy. Lastly, many models, such as HotSpot [10], still rely on computationally expensive numerical methods to determine temperature solutions, which may not be sufficiently fast for real-time applications.

Conversely, from a system-level thermal or power management perspective, readily accessible parameters include core frequency, voltage, and various utilization or performance metrics, which are natively supported by most commercial processors [15]. Software tools such as Intel's Performance Counter Monitor (PCM) [16] and AMD's uProf [17] provide the means to profile these metrics. Recently, Deep Neural Network (DNN) based approaches have been investigated for full-chip thermal map estimation of commercial multi-core processors using real-time PCM performance metrics [18]–[20]. These methods demonstrate the feasibility of fast online thermal map estimation for the first time. However, the Long Short Term Memory (LSTM) based method [18], [19] still suffers from low accuracy issues, and the Generative Adversarial Network (GAN) based method [20] is more suitable for steady-state thermal estimation due to its inability to handle transient data. Furthermore, this method has not been demonstrated in commercial AMD multi-core processors, which account for the second largest market share in the multi-core processor market.

In this work, we aim to address the aforementioned issues and propose a novel transformer-based approach for estimating the full-chip thermal map of commercial AMD multiprocessors. The key contributions of this study are as follows.

- We have developed a DNN-based full-chip thermal map estimation method, named *ThermTransformer*, for commercial AMD multi-core microprocessors for the first time. The case we studied here is the Ryzen 7 4800U 8-core CPU. Unlike conventional approaches that rely on traditional functional unit powers, the new DNN model utilizes real-time on-chip performance metrics of AMD chips obtained from *uProf 4.0* as input features.
- To mitigate the shortcomings of existing methods, such as the low accuracy of LSTM for image generation and the limited applicability of GAN-based methods for transient thermal prediction, we propose the use of transformer-based DNN models, which are well-suited for handling both time series input and image data outputs.
- We employ an advanced infrared thermography setup system that allows us to directly capture lucid heatmaps from commercial microprocessors during their operation under workloads. A total of 14718 pairs of performance

The work is supported in part by NSF grant under No.CCF-2007135, and in part by NSF grant under No. CCF-2113928.

metrics data and thermal maps were collected, with 80% of the data used for training purposes.

- The resulting *ThermTransfomer* can provide tool-accurate full-chip *transient* thermal maps from the given performance monitor traces of commercial off-the-shelf AMD multi-core processors.
- Experimental results demonstrates the high accuracy of the thermal map predictions, with a root-mean-square error of only $0.36°\mathrm{C}$ or 0.8% of the full-scale error. More importantly, our method outperforms the recently proposed GAN-based thermal map estimation method, *ThermGAN*, by 1.66x and the LSTM-based method, *RealMaps*, by 6.09x in terms of accuracy on average. Furthermore, the proposed approach can be deployed on the target AMD CPU with a fast inference speed of $7\mathrm{ms}$, making it suitable for real-time estimation.

This article is organized as follows. Section II provides a review of relevant work. Section III outlines the thermal modeling framework and IR thermography setup employed in this study. Section IV explains the process of collecting and preparing the training thermal data, as well as the selection of performance metrics features for the proposed method. Section V describes the architecture of the proposed transformer-based model for thermal map estimation. Section VI presents the experimental results and provides comparisons. Section VII concludes the article.

II. Related work

To estimate on-chip temperature maps, two general strategies are commonly employed. The first approach involves estimating the full-chip heatmaps using physics-based thermal models and power-related information [11], [12]. These 'bottom-up' numerical methods, such as HotSpot [10] based on simplified finite difference methods, finite element methods [21], and equivalent thermal RC networks [22], have been widely used. Recently, top-down behavioral thermal models based on the matrix pencil method [23] and subspace identification method [24], [25] have also been proposed. However, these full-chip thermal analysis methods are computationally expensive and unsuitable for online applications [26].

The second strategy involves using interpolation-based approaches to estimate the full-chip heatmaps from embedded sensor readings [8], [27]. The accuracy of this interpolation-based approach is influenced by the number and placement of sensors. To address this, smart sensor placement algorithms have been proposed to optimize the sensor placement within a given budget of embedded temperature sensors [27]–[32]. Some works have employed Fourier analysis techniques to recover the thermal map [27]. However, the accuracy is limited by the non-band-limited nature of the temperature signals and the approximations required for non-uniform sensor placement, which is common in heterogeneous multi-core processors. Other methods have attempted to minimize the number of thermal sensors and recover thermal maps by interpolating hard sensor information in the frequency and DC domains [28], [29]. The use of eigendecomposition of the interpolation matrix has further improved the sensor placement strategy, leading to near-optimal sensor numbers and placements [30].

A statistical method has been proposed by Zhang et al. [31], [33] for estimating both power and thermal maps. This method exploits the correlations of power dissipation among different modules of a chip to recover the power map from sensor readings, followed by temperature estimation. However, the estimation is based on power correlation information. Recently, Ziabari et al. [34] introduced the power blurring method for fast 2-D temperature map computation. This method utilizes the Green's function approach, where the temperature response to unit power impulses is computed first using finite element thermal analysis. However, the practicality of this method is limited as accurate thermal models are not always available in all cases.

However, the methods mentioned above either necessitate hardware modifications during the design phase, such as inserting or relocating sensors, or they rely on detailed knowledge of the chip's floorplan, power source correlations among functional units, and technology-specific constants that are typically not disclosed by the chip manufacturer. Consequently, achieving real-time estimation of the spatial temperature distribution across the entire chip area solely through a post-silicon approach, i.e., estimating the full-chip spatial heatmap $T(x,y)_t$ at time t, remains a significant challenge for existing commercial microprocessors.

Conversely, in recent years, machine learning, particularly deep learning, has gained significant attention due to its remarkable performance breakthroughs in various cognitive applications, including visual object recognition, object detection, speech recognition, and natural language understanding [35], [36]. Several popular DNN structures have emerged, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) (including LSTM), and more recently, self-attention-based Transformer [37]. Transformers have shown their ability to handle time series data more effectively than RNNs and have led to recent breakthroughs in generative AI applications, as exemplified by ChatGPT [38]. Moreover, transformers, such as the Vision Transformer, have demonstrated superior performance compared to CNNs for image classification, given sufficient data and training [39]. Some thermal analysis approaches based on power information using machine learning methods have been proposed. Examples include the generative adversairial networks (GAN) based method [20], the convolutional encoder-decoder network based method [40] and the graph convolution networks (GCN) based method [41].

To address the aforementioned challenge of real-time full-chip thermal map estimation without power information, recent research has explored machine-learning-based approaches utilizing real-time performance metrics [18]–[20]. Sadiqbatcha et al. [18], [19] proposed a LSTM-based approach called *Realmaps*, which employs Intel PCM metrics for estimating full-chip thermal maps in commercial off-the-shelf multi-core processors. This approach has shown promising results. Building upon this work, an improvement was made by employing image-friendly CNN model based on the GAN architecture. This approach, known as *ThermGAN* [20], shows better results than the LSTM-based methods. However, it is important to note that *ThermGAN*'s predictions are based solely on the current PCM information and do not incorporate historical data. Therefore, it is more suitable for steady-state thermal map estimation.

III. Thermal map estimation framework

In this section, a brief overview of the proposed approach will be presented, along with a description of the thermal camera setup used to collect the necessary data from an AMD Ryzen 7 4800U chip while it is under workload.

A. Estimation flow overview

The proposed approach is divided of three parts. Firstly, we collect data by monitoring the AMD CPU's performance metrics, including Performance Monitoring Unit (PMU) event counters and on-chip sensor data, while executing workloads on the chip. Simultaneously, we capture full-chip thermal map measurements using a thermal infrared (IR) imaging system.

Subsequently, we use the collected data to train a transformer-based online thermal prediction model.

The training process of the transformer-based model requires two types of data. One consists of the offline measured thermal maps, which serve as the training targets for the model. The other comprises the recorded PMU event counters and sensor data, which are used as indicators to generate predictions. Once the model is trained, it can be used for online CPU thermal inference.

The framework of the proposed approach is illustrated in Fig. 1. In the following section, we will provide a detailed description of the data acquisition process encompassing each step shown in the figure. In Section V, we will further elaborate on the machine learning model.

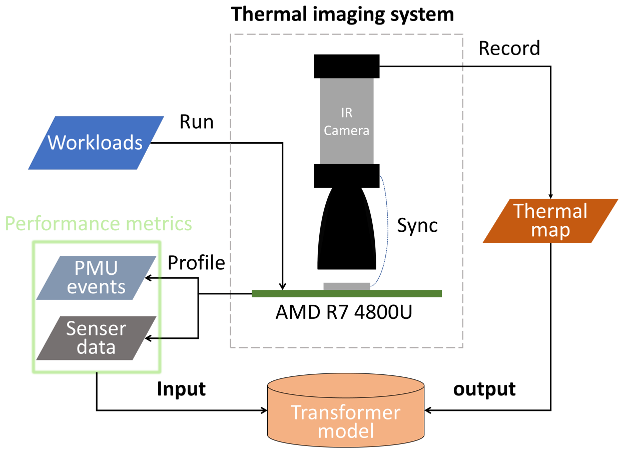


Fig. 1. Framework and data acquisition flow

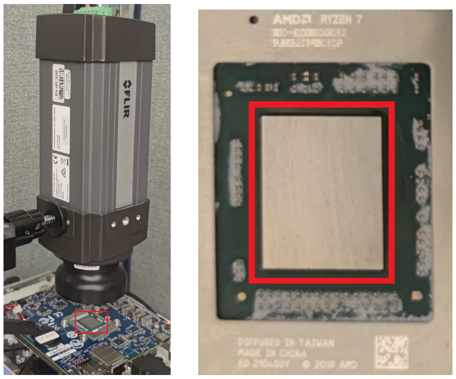
B. Thermal IR imaging system

The effectiveness of the proposed machine-learning-based approach heavily relies on precise measurements of CPU surface temperature maps. To facilitate this, we introduce an advanced IR thermal imaging system, as depicted in Fig. 2. In order to ensure proper thermal operating conditions, we have opted for a bottom-side liquid cooling system [7] instead of the conventional top-side approach. This back-side liquid cooling technique involves the use of a thermoelectric (Peltier) device directly mounted on the PCB beneath the processor module, allowing for efficient cooling from below. Consequently, the front side of the processor remains wholly exposed to the IR camera, free from any intervening layers that may cause interference.

The product information of the IR thermal imaging system are provided as follows. The IR camera is a FLIR A325sc, offering a maximum imaging resolution of 240 × 320 pixels (px) with a precision of 16 bits per px. It is capable of capturing thermal images at a maximum frequency of 60Hz. The IR sensor comes factory calibrated to ensure accuracy across a temperature range of $-20°$C to $120°$C and resolves the IR spectral range of $7.5\mu m$ to $13\mu m$.

IV. Data Preparation and Performance Metrics Selection

In this study, our objective is to develop a model that can generate thermal maps using the real-time performance metrics of the chip. Our machine-learning architecture follows a supervised learning approach, highlighting the significance of having an appropriate dataset. As mentioned earlier, the training data for our learning-based model consists of offline thermal maps measured across the entire chip area, along with the corresponding performance metrics collected during runtime. To obtain this dataset, we begin by executing benchmark workloads on the CPU. During the execution, we continuously capture the performance metrics and temperature map at regular intervals. Each timestamp corresponds to a pair of historical metrics and the thermal map, forming a single data point. As the workload progresses, we accumulate numerous data points, resulting in a comprehensive dataset for analysis. This section provides a detailed description of the data acquisition process.



(a) (b)

Fig. 2. (a) Thermal Imaging system setup (b) CPU chip under-test, AMD Ryzen 7 4800U. Core module is shown in the red box.

A. Benchmark workloads

To simulate the operational conditions of the processor across different working environments, we will assign a diverse range of workloads to it during the data acquisition process. In this study, we utilized the SPLASH-2x suite available in PARSEC 3.0 [42], an open-source benchmark software for Linux, which provides a wide range of workloads. Specifically, the following SPLASH-2x workloads were executed on the processor: barnes, cholesky, fft, fmm, ocean_cp, ocean_ncp, radiosity, radix, raytrace, volrend, water_nsquared, and water_spatial. These workloads were performed with varying sizes of input to cover different situations.

B. Thermal map acquisition

During the actual measurement, the chip does not occupy the entire 240×320 px field of view of the camera. Therefore, we need to crop the specific chip area from the captured photo. As a result, the final size of the chip thermal map is reduced to 223×280 px. In keeping with the profiling speed of performance metrics, the camera records at 10Hz. Fig. 3 shows the thermal map example of the commercial AMD Ryzen 7 4800U 8-core microprocessor, in which one of the cores is running a workload with elevated temperature.

C. Performance metrics acquisition

In this work, we study an AMD Ryzen 7 4800U chip (8 cores, 16 threads, released in 2020). To monitor the performance of this chip, we utilize AMD *uProf 4.0* [17], which is the performance monitoring software supported by AMD. This software allows us to gather system-level utilization metrics as well as core-wise power characteristics from the chip. By utilizing these metrics, we can gain insights into the chip's performance and thermal behavior.

AMD *uProf 4.0* offers two types of performance metrics that are relevant to our study. The first type comprises CPU power metrics, including package and core energy usage, core frequency, and temperature readings from embedded

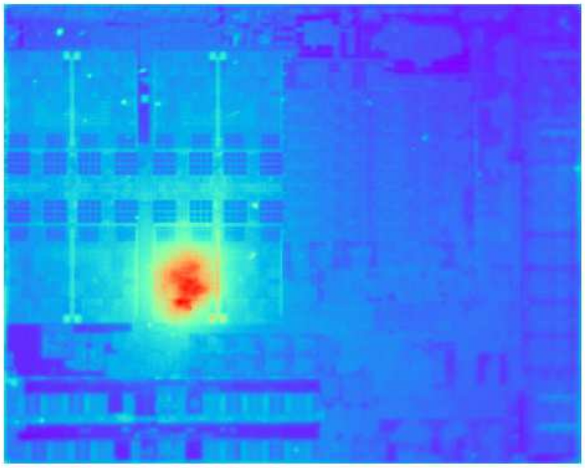


Fig. 3. The thermal image example of AMD Ryzen 7 4800U 8-core microprocessor in which one core is actively running

TABLE I
SELECTED PERFORMANCE METRICS (AMD R7 4800U)

CPU power metrics		
ThreadEffectiveFrequency 0-15	ThreadPerformanceState 0-15	Socket0Power
CorePower 0-7	Socket0Temperature	
PMU events		
FpRetSseAvxOps	FpRetiredSerOps	FpDispFaults
LsBadStatus2	LsLocks	LsRetClClush
LsRetCpuid	LsDispatch	LsSmiRx
LsIntTaken	LsSTLF	LsStCommitCancel2
LsMabAlloc	LsRefillsFromSys	LsL1DTlbMiss
LsMisalLoads	LsPrefInstrDisp	LsInefSwPref
LsSwPfDcFills	LsHwPfDcFills	LsAllocMabCount
LsNotHaltedCyc	IcCacheFillL2	IcCacheFillSys
BpL1TlbMissL2TlbHit	BpL1TlbMissL2TlbMiss	BpL1BTBCorrect
BpL2BTBCorrect	BpDynIndPred	BpDeReDirect
BpL1TlbFetchHit	DeDisUopQueueEmpty	DeDisUopsFromDecoder
DeDisDispatchTokenStalls1	DeDisDispatchTokenStalls0	ExRetInstr
ExRetCops	ExRetBrn	ExRetBrnMisp
ExRetBrnTkn	ExRetBrnTknMisp	ExRetBrnFar
ExRetNearRet	ExRetNearRetMispred	ExRetBrnIndMisp
ExRetMmxFpInstr	ExRetCond	ExDivBusy
ExDivCount	ExRetMsprdBrnchInstrDirMsmtch	ExTaggedIbsOps
ExRetFusBrnchInst	L2RequestG1	L2RequestG2
L2CacheReqStat	L2PfHitL2	L2PfMissL2HitL2
L2PfMissL2L3		

sensors, among others. The second type consists of PMU event counters, which track various events such as instruction counts, cache hits, and branch predictions. In Table I, we present the list of AMD *uProf 4.0* performance metrics from these two domains. The CPU Power Metrics domain encompasses 42 metrics, while the PMU Events domain includes 58 different types of events. It is important to note that some PMU events have specific unit masks to distinguish different conditions, resulting in further categorization. As a result, we have selected a total of 116 metrics for the PMU Events domain. Combining both domains, we have a total of **158** metrics selected for the AMD Ryzen 7 4800U chip.

For thermal prediction, we use the metrics vectors from the previous 100 frames and consider them as a time series that can be processed similarly to sentences in natural language tasks. Having a longer history of data helps enhance the accuracy of state estimation but also increases the resource requirements. The length of the historical data can be adjusted according to the specific needs of the application.

V. THERMTHANSFOMER: THE PROPOSED TRANSFORMER-BASED FULL-CHIP THERMAL ESTIMATION MODEL

A. Review of the transformer DNN architecture

As mentioned earlier, our problem can be viewed as on-chip thermal image generation task from a given chip performance metrics vector. The input vectors can be viewed as a time series data over a period of time. To process such type of data, the self-attention based transformer architecture can be a good fit [37]. Transformer mitigates the difficulty to model long-history relationship in a time series data in the existing RNN based structure using the self-attention mechanism, which allow the model to access all previous states and process the entire input all at once, which can be highly parallelized.

In the sequel, we briefly review the basic attention layer structure and formula used in the transformer DNNs. Given a sequence of input vector $\{x_i\}$, the attention unit first embeds each vector x_i into three vectors: query vector q_i, key vector k_i, and value vector v_i, which are computed as

$$q_i = x_i W^Q, k_i = x_i W^K, v_i = x_i W^V$$

Here q and k share the same dimension d_{qk}. Then a weighted average will be performed among $\{v_i\}$. The ith weight comes from the dot production between $\{q_i\}$ and $\{k_i\}$, divided by $\sqrt{d_{qk}}$ and then passed through a softmax for the normalization. The weighted average is the attention vector $Attention$ $(\{q_i\}, \{k_i\}, \{v_i\})$. If we write the sequence of vectors in a way of matrices (i.e. Q, K, V for the matrices where the i-th rows are q_i, k_i, v_i), then the attention can be represented in a more concise form. This operation is shown in the left side of Fig. 4.

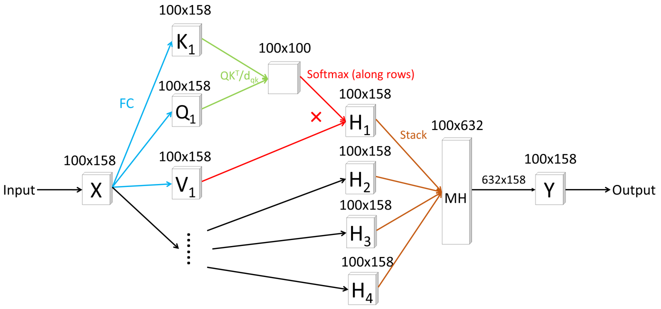


Fig. 4. Structure of the Multi-Head Attention layer in this work

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{qk}}}\right) V \qquad (1)$$

where softmax is along the rows. Now the input vectors sequence $\{x_i\}$ becomes a single vector output. In general, we may want to process the data with multiple attention interests. Then we can have multiple $\left(W^Q, W^K, W^V\right)$ (which is called the head), concatenate multiple attention vectors together into a matrix, and then project it into a final output matrix as shown below:

$$\begin{aligned} &\text{MultiheadedAttention}\left(Q, K, V\right) = \\ &\quad \text{Concat}\left(\text{Attention}\left(\mathrm{QW_i}^{\mathrm{Q}}, \mathrm{KW_i}^{\mathrm{K}}, \mathrm{VW_i}^{\mathrm{V}}\right)\right) \mathrm{W}^{\mathrm{O}} \end{aligned} \qquad (2)$$

where $\left(W_i{}^Q, W_i{}^K, W_i{}^V\right)$ are the mulitheads, and W^O is the final projection matrix. Fig. 4 shows an example of the multiheaded attention layer used in our work. By repeating multiple attention layers, we have the typical encoder or decoder structure in the original transformer model [37].

TABLE II
ACCURACY COMPARISON AMONG DIFFERENT METHODS

Accuracy	ThermTransformer	ThermGAN [20]	LSTM [19]	Accuracy imp over [20]	Accuracy imp over [19]
Average RMSE	0.360	0.596	2.190	1.656x	6.085x
Maximum RMSE	2.776	10.880	19.762	3.919x	7.118x
RMSE deviation	0.234	0.958	2.684	4.095x	11.470x

B. *Proposed transformer-based thermal estimation framework*

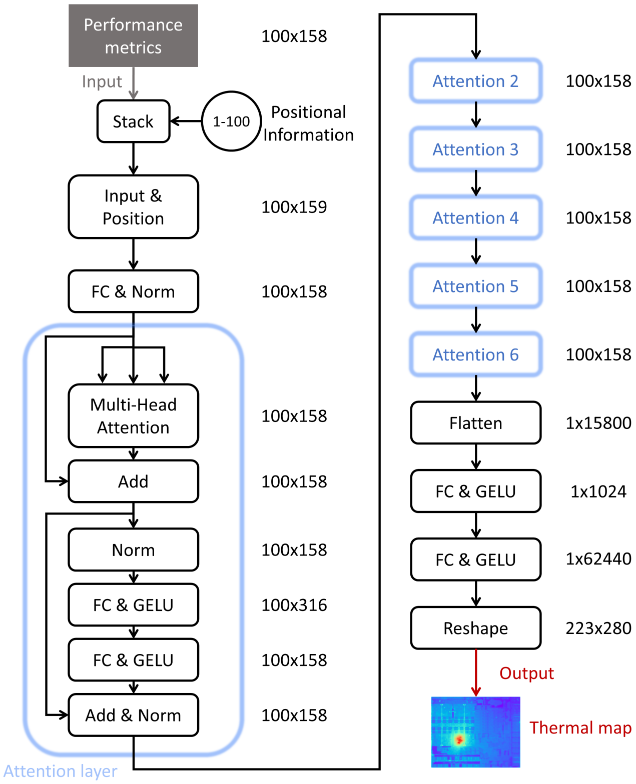


Fig. 5. Architecture of proposed Transformer model

The framework of our metrics-to-thermal-map model is shown in Fig. 5. In contrast to the encoder-decoder structure originally proposed for language translation tasks, we solely utilize the encoder structure of the original transformer [37], since our objective is to predict new thermal images based on historical performance metrics data. Instead of generating internal codes or features for the given performance metrics data, we add multi-layer perceptron (MLP) heads at the end of the attention layers and shape the final outcome into a 223×280 image. This strategy is similar to the approach in GPT-1 [43], which only use the decoder structure to predict the next word.

The input sequence $\{x_i\}$ is composed of 100 1x158 vectors. Then we extend the length of the vectors by one to incorporate the position number, which can be viewed as integrating timing information. A straightforward positional encoding scheme is adopted, wherein the extended vectors are directly mapped back to 1x158 using a fully-connected (FC) layer.

Then the data moves to the attention layer, which is indicated by the large blue box. Inside the attention layer, it has multi-head attention structure shown in Fig. 4. After this multi-head attention computation, the data moves through the normalization layer or "Norm" layer. We use the Gaussian Error Linear Unit (GELU) [44] as the activation function. All FC layers operate on columns. For instance, in the attention block we have an "FC&GELU" layer after the "Norm" layer, transforming a 100x158 input to a 100x316 output. Instead of having a weight matrix of size 15800x31600, here we only employ a matrix of 158x316, which is applied separately to 100 vectors. We have six such attention layers shown in blue in our design. Following these attention layers, the extracted feature data undergoes flattening before passing through a MLP, which has two "FC&GELU" layers. Eventually, the output is reshaped into the final 223x280 thermal image.

Last not least, for the estimation, we minimize the L2 loss between $F(x)$ and ground truth y. The loss function is:

$$loss_G = \mathbb{E}_{(x,y)}[||y - F(x)||^2] \tag{3}$$

VI. Experimental results and discussions

In this section, we present the experimental results of the proposed approach.

We evaluate the proposed method on a commercial AMD Ryzen 7 4800U 8-core microprocessor, which features the Zen 2 (Renoir) architecture and is commonly used in thin and light laptop computers. The implementation of the proposed *ThermTransformer* is based on Python 3.8 and utilizes TensorFlow (2.11.0) [45], a widely adopted open-source machine learning library. The model is trained on a Linux server equipped with a Xeon E5-2699v4 2.20GHz processor and an NVIDIA Titan RTX GPU. During training, a batch size of 32 is employed, and each data sample consists of historical performance metrics and a corresponding thermal map. The final dataset comprises a total of 14718 data points, with 11774 of them for training and 2944 of them reserved for testing.

To assess the prediction accuracy, we compare the proposed *ThermTransformer* with two recently proposed methods for full-chip thermal map estimation: the LSTM-based approach, called *RealMaps* [18], [19] and the GAN-based approach, called *ThermGAN* [20]. Both of these methods are also implemented in Python using TensorFlow [45]. All the DNN models are trained on the same dataset, and the training process terminates when there is no significant improvement in performance.

A. *Thermal map estimation accuracy*

We begin by examining the accuracy of predicting thermal maps using the proposed method on the given dataset. To evaluate the accuracy, we calculate the Root-Mean-Square Error (RMSE) between the generated thermal map and the measured thermal map across all pixels.

In our dataset, the temperature ranges from 44.11 to 90.08°C. The average RMSE of the thermal map estimation on the test set is 0.360°C, with a standard deviation of only 0.234°C. These results are remarkably accurate considering the range of the data. Fig. 6 illustrates the estimated and measured thermal maps, showcasing examples from the test set. Each column in the figure represents the comparison results at a specific time step. We display the results for three time steps. It is evident that the transformer-based model has successfully learned the contour of the real thermal map.

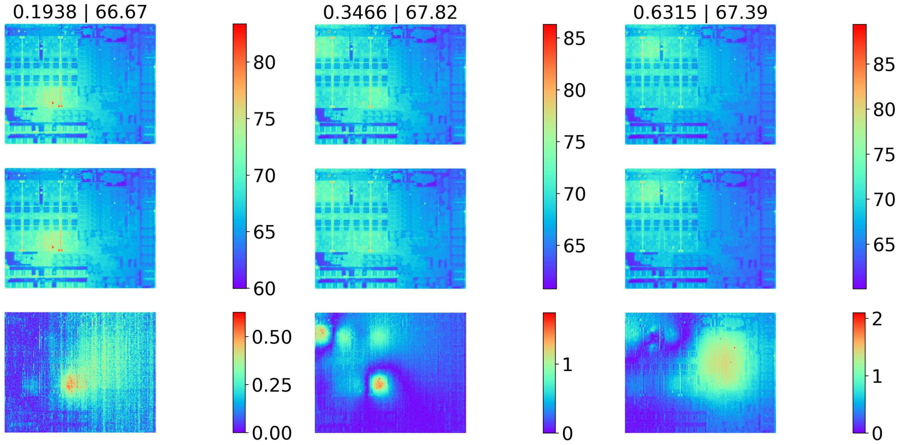


Fig. 6. Measured thermal maps (row #1), estimated thermal maps (row #2), and error maps (row #3). The numbers above the maps indicate the *Temperature RMSE* | *Average Temperature* (unit: °C). Each column indicate result at one particular time step.

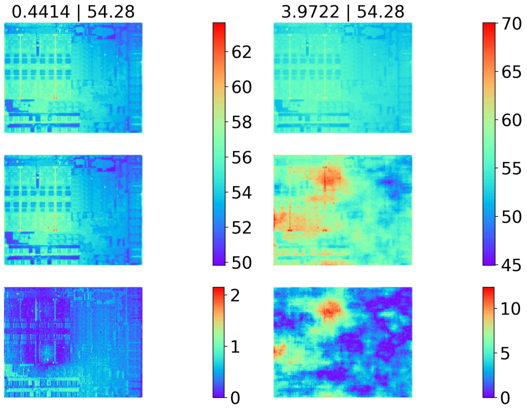


Fig. 7. Accuracy comparison with ThermGAN at one time step. The left column is the result of our model, and the right column is the result of ThermGAN.

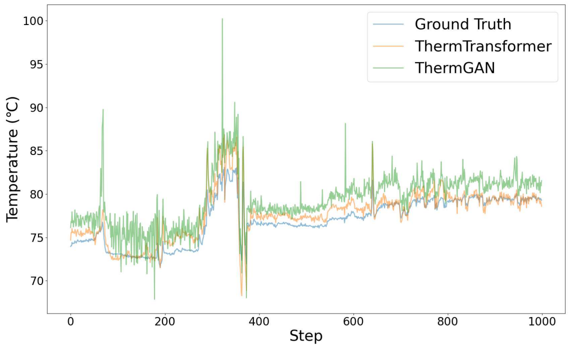


Fig. 8. Comparison in transient temperature predicted by the two models and the ground truth at a fixed location in the AMD chip over time

Furthermore, we conducted a comparison between our method and two recently proposed full-chip thermal map estimation methods: the LSTM-based approach, *RealMaps* [18], [19] and the GAN-based approach, *ThermGAN* [20], in terms of prediction accuracy. The results are presented in Table II. Our proposed *ThermTransformer* demonstrates approximately 1.66x higher accuracy than *ThermGAN* on average and 6.09x higher accuracy than *RealMaps*. In terms of the maximum RMSE, the proposed method is about 3.92x and 7.12x more accurate than *ThermGAN* and *RealMaps*, respectively. This trend also applies to the RMSE deviation. Overall, the proposed *ThermTranformer* exhibits significant accuracy improvement over the other two methods across all accuracy measurements for the given dataset.

Fig. 7 presents the predictions of *ThermTransformer* and *ThermGAN* using the same input. Clearly, the proposed method yields significantly lower errors in both contour and temperature compared to the results given by *ThermGAN*. Fig. 8 illustrates the predictions of two models at a fixed pixel over time. We observed a substantial higher error exhibited by *ThermGAN*. Additionally, *ThermGAN* demonstrated high instability, often leading to incorrect predictions.

B. Computational efficiency

The training procedure typically takes a few to several hours to complete. Once the model is properly trained, it can be deployed for real-time thermal prediction. In our experiments, we measured an average inference time of 14ms on the target board. This inference time is comparable to that of ThermGAN (16ms) and LSTM (19ms). The low latency ensures the effectiveness of real-time thermal estimation. Compared to the performance metrics collection interval of 100ms, the proposed model is sufficiently fast to keep up with the process.

VII. Conclusion

In this article, we propose a machine-learning-based approach for real-time estimation of full-chip thermal maps for the commercial AMD Ryzen 7 4800U CPU. The new method leverages performance monitor traces of multi-core processors and sensor information as input for the machine-learning models. To construct the dynamic thermal map model, we have developed a modified self-attention architecture to model thermal maps. Experimental results further show that *ThermTransformer* outperforms recently proposed GAN-based thermal map estimation methods, *ThermGAN*, by 1.66x and LSTM-based thermal prediction methods, *RealMaps*, by 6.09x in terms of average accuracy. The predictions of thermal maps exhibit a high level of accuracy, with an average RMSE of only 0.36°C degrees or 0.8% of the full-scale error. Furthermore, the proposed approach can be deployed on the target CPU with a speed of just 14ms, making it 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, no. 3, 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, no. 5, pp. 8–19, October 2013.

[4] 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.

[5] 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

[6] S. Sadiqbatcha, J. Zhang, H. Zhao, H. Amrouch, J. Henkel, 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] 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 software profiling tool," https://developer.amd.com/amd-uprof/.

[18] S. Sadiqbatcha, Y. Zhao, J. Zhang, H. Amrouch, J. Henkel, and S. X. D. Tan, "Machine learning based online full-chip heatmap estimation," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 229–234.

[19] 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.

[20] W. Jin, S. Sadiqbatcha, J. Zhang, and S. X.-D. Tan, "Full-chip thermal map estimation for commercial multi-core cpus with generative adversarial learning," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*. New York, NY, USA: ACM, Nov. 2020, pp. 1–9.

[21] 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.

[22] Y. C. Gerstenmaier and G. Wachutka, "Rigorous model and network for transient thermal problems," *Microelectronics Journal*, vol. 33, pp. 719–725, September 2002.

[23] 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.

[24] 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.

[25] 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.

[26] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang, *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.

[27] 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.

[28] 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.

[29] 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.

[30] 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

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

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, http://www.deeplearningbook.org.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[38] "Chatgpt from openai." [Online]. Available: https://chat.openai.com

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[40] V. A. Chhabria, V. Ahuja, A. Prabhu, N. Patil, P. Jain, and S. S. Sapatnekar, "Thermal and ir drop analysis using convolutional encoder-decoder networks," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, 2021, pp. 690–696.

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

[42] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2008.

[43] A. Radford and I. Sutskever, "Improving language understanding by generative pre-training," in *arxiv*, 2018.

[44] K. G. Dan Hendrycks, "Gaussian Error Linear Units (GELUs)," *arXiv e-prints*, p. arXiv:1606.08415v4, June 2016.

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/