

Sensor Placement and Transformer-Based Thermal Map Generation for Reusable Interposers

Aristotelis Tsekouras, Theodoros Papavasileiou, Panagiotis C. Petrantonakis,
Georgios Keramidas, and Vasilis F. Pavlidis
Aristotle University of Thessaloniki, Greece
{aristotet, ppetrant, vpavlid}@ece.auth.gr, {tpapava, gkeramidas}@csd.auth.gr

Abstract—2.5D integration has been a promising packaging approach intrinsically underpinning heterogeneous integration. The physical proximity of diverse components (*e.g.*, chiplets) on interposers entails multi-physics, including thermal coupling, which affects the performance and reliability of the entire system. Consequently, interposer-level thermal monitoring is required to avoid overheating during run-time. Furthermore, reusable interposers have also recently been proposed in the literature, implying that a specific interposer is used for multiple systems. Therefore, conventional thermal sensor placement methods, developed for a specific system, are incompatible with this emerging design concept. A new flow focusing on thermal sensor allocation and thermal map reconstruction for reusable interposers is proposed. The flow utilizes a transformer neural network to reconstruct the thermal map of the interposer and hyperparameter tuning to select the appropriate thermal sensor locations that minimize the reconstruction error across the entire set of available floorplans for a specific transformer architecture. The benchmarks used to train the transformer are produced through gem5, McPat, HotSpot and TAP-2.5D for ten different floorplans, showcasing the effectiveness and generality of the approach compared with prior art and achieving an average maximum error of less than 1K.

Index Terms—Interposer, 2.5D integration, thermal sensors, deep learning, hyperparameter tuning, transformers

I. INTRODUCTION

The use of interposers and 2.5D integration technologies has increased recently. Interposers, which act as an interconnect fabric for multiple components, have become an increasingly attractive solution for addressing the challenges arising from the nanometer scaling of semiconductor devices [1]. Furthermore, interposers enable high routing density, lower energy dissipation, and heterogeneous integration (HI), which is the integration of components from different processes and technologies [2]. As a result, interposers have become a key enabler among several applications, such as high-performance computing [3], automotive [4], and telecommunications [5].

Interposers are categorized as either active or passive. In active interposers, transistors can be fabricated on the interposer, whereas passive interposers comprise only interconnects and passive components. Although passive interposers exhibit low fabrication cost, active interposers have also been explored as a potent packaging approach [6]. In addition, the concept of reusable interposers can mitigate the high cost and shorten the development cycle as discussed in [7], where a reusable general interposer architecture (GIA) is proposed, applicable to both active and passive interposers.

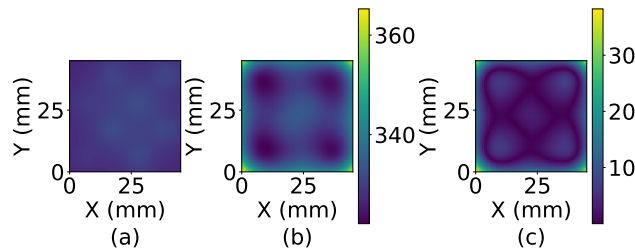


Fig. 1. (a) Thermal map of the interposer with 16 chiplets, (b) estimation of the thermal map using solely the chiplet thermal sensors and RBF Multiquadric interpolation and (c) the temperature estimation error in Kelvin.

With the advances of HI and the strides in both active and passive interposers, thermal monitoring at the system level, where the term “system” means, herein, both the integrated components (*e.g.*, chiplets) and the interposer, has become crucial. HI implies diverse components sourced from different technologies, some of which exhibit high sensitivity to temperature fluctuations or require controlled heating to function, such as phase-change materials for silicon photonics [8]. In addition, thermal coupling, if not considered, can affect the operation and reliability of the system due to overheating [9], [10]. For this purpose, several thermal-aware chiplet placement algorithms have been proposed to alleviate thermal issues, stressing the importance of system-level thermal monitoring and management [11], [12], [13].

Furthermore, thermal monitoring across the entire system cannot be achieved using solely the information provided by the on-chiplet sensors and interpolation-based estimation techniques. As a motivational example, consider the temperature distribution of a system of 16 chiplets on an interposer, as depicted in Fig. 1. Using Radial Basis Function (RBF) interpolation, a temperature estimation is produced in Fig. 1b. The estimation error in the areas between the chiplets is significant, exceeding 5 K, while around the periphery and corners of the interposer can reach over 30 K, as shown in Fig. 1c. This level of error is unacceptable, and, therefore, the availability of sensors in an interposer and the use of more accurate reconstruction techniques is vital to capture and manage the thermal effect of a component to the entire system in order to avoid excessive heating across the system. In addition, mechanical stresses from thermal gradients on different components of the interposer (*e.g.*, μ bumps, TSVs) and thermal aging across the

large area of the interposer cannot be accurately evaluated by the local on-chip sensors that provide the junction temperature of each chiplet thermally far from the interposer substrate.

None of the State-of-the-art (SotA) approaches can address these issues, readily providing holistic thermal monitoring applicable to any ensemble of components and any placement of these components on an interposer substrate. Therefore, this paper innovates through a thermal sensor placement approach and a thermal map reconstruction method, both leveraged by machine learning (ML), targeting reusable interposers and capturing accurately the thermal intricacies of 2.5D systems. The new method is validated across diverse floorplans. The floorplans differ from each other in the number and/or placement of the components. In the new flow, the sensor locations are considered as hyperparameters and are tuned during the hyperparameter tuning process, while ML helps with real-time thermal map reconstruction. Hyperparameter tuning considers the influence of the thermal sensor locations on the thermal map generation accuracy and is suitable for large datasets, making the approach useful for reusable interposers that should accommodate multiple floorplans. This approach differs from other techniques that fail to generalize on reusable interposers or rely on methods that cannot satisfy the required accuracy.

Consequently, the contributions of this paper are:

- A dataset of thermal maps is created utilizing open-source tools, including gem5 [14], McPat [15] and HotSpot [16].
- The flow enables sensor placement through hyperparameter tuning and utilizes a transformer neural network for thermal map reconstruction offering superior results compared to prior art.
- The results are produced for a different number of sensors, ranging from eight to 64 sensors, and are compared with other SotA methods in known and unseen floorplans. Our framework, where the term “framework” refers to the transformer-sensor configuration, and the thermal sensor allocation flow, achieves a low average RMSE and a total average maximum error that does not exceed 1K.
- The utilized code and results presented in Section V are publicly available ¹.

The remainder of this paper is organized as follows. Prior sensor placement and thermal reconstruction methods are described in Section II. The thermal modeling process for the evaluated benchmarks of the dataset utilized by the transformers is presented in Section III. The proposed framework and hyperparameter tuning flow are presented in Section IV and the experimental results are discussed in Section V, respectively. Finally, Section VI summarizes our work.

II. RELATED BACKGROUND

Prior works have focused on developing allocation strategies for on-chip thermal sensors. Memik *et al.* [17] proposed a technique to place thermal sensors in microprocessor systems. The technique relies on an interpolation method that calculates the temperature between the thermal sensors located in a grid.

Nowroz *et al.* [18] propose two methods for sensor allocation, an energy-center-based and an energy-cluster-based method, respectively. The energy-center method places a sensor at the geometric center of a cutting region after bisecting the die, while the energy-cluster method places the sensor at the centroids computed by k -means clustering.

More recent works emphasize solving the thermal map reconstruction problem at chip-level along with the sensor allocation problem. Ranieri *et al.* [19] propose a framework for thermal map reconstruction of many-core SoCs using a small number of sensors. The authors used the results from thermal simulations during design time to generate a linear model that represents the thermal behavior of the die with the use of principal component analysis. A greedy sensor allocation algorithm is also utilized to minimize the reconstruction error. However, capturing features in data with non-linear relationships is a challenge for this approach.

Chen *et al.* propose a temperature reconstruction technique for NoCs [20], [21]. Initially, the sensor allocation is performed randomly, while the temperature estimation of the nodes without sensors is performed by using compressed sensing [22] with Orthogonal Matching Pursuit. An entropy-based allocation technique [23] has shown to significantly reduce the thermal reconstruction error compared to [19] and [21]. Finally, with the LASSO method, the sensor locations are initialized and are, subsequently, adjusted with Q-learning (Q-LASSO) to reduce the prediction error [24]. LASSO and compressed sensing, however, make sparsity assumptions that cannot efficiently represent a dataset consisting of multiple different floorplans.

Several neural network models have also been proposed to solve the problem of thermal map reconstruction. Xin *et al.* utilize convolutional neural networks to recreate the thermal map using only the temperatures of the available sensors [25]. Wentian *et al.* present *ThermGAN*, a thermal map estimation method that uses a Generative Adversarial Net [26]. Other methods have also been proposed that include the use of graph convolutional networks [27] and long short-term memory (LSTM) [28]. Recently, Jincong *et al.* have presented *ThermTransformer*, which estimates the thermal maps in real time utilizing a transformer and specific input metrics [29]. This method exhibits the smallest average RMSE compared to other neural network techniques [25]-[28], showing the advantages of transformers in thermal map reconstruction problems and are, therefore, used as the starting point in this work. However, *ThermTransformer* considers only the metrics of a specific system, and the authors do not demonstrate whether the method can be generalized to reusable interposers. Therefore, this work expands on the *ThermTransformer* by demonstrating that, through hyperparameter tuning and transformer-based reconstruction, systems with different floorplans can be monitored with a small margin of error overcoming the limitations of prior art in the field.

III. BENCHMARKS AND THERMAL MODELING

The dataset required to train the transformer framework is produced via modeling and simulation analysis [30] of the aimed systems. The simulations include gem5 [14], McPat [15],

¹<https://github.com/AristoTsek/Sensor-Placement-and-Transformer-Based-Thermal-Map-Generation-for-Reusable-Interposers>

TABLE I
CONFIGURATION PARAMETERS OF ANALYZED SYSTEMS

| Parameters | Value |
|--------------------------|--------------|
| Core Clock Rate (GHz) | 2.2 |
| Core Area (mm^2) | 73.9944 |
| L3 Area (mm^2) | 117.627 |
| Workload Duration (s) | 4 |
| Power Sample Period (ms) | 1 |
| Initial Temperature (K) | 318.15 |
| L3 cache | 16MB/16-ways |

TABLE II
DEFINITION OF UTILIZED VARIABLES

| Variable | Definition |
|--------------------------|--|
| n_S | Number of available sensors |
| $d_{out} \times d_{out}$ | Resolution of the reconstructed thermal maps |
| seq_len_{TR} | Length of the input sequence at Transformer TR |
| T_{TR} | Period of operation of Transformer TR |
| T_S | Sampling period of the thermal sensors |
| τ_{TR} | Inference time for Transformer TR |
| d_{model} | Dimensionality of the data used to represent the inputs in the transformer |
| d_{ff} | Hidden Feed-Forward layer dimension |
| num_heads | Number of attention heads |
| num_encod | Number of encoders |

and HotSpot [16]. Several of the configuration parameters are listed in Table I. Without loss of generality, the investigated system comprises a CPU core communicating with an L3 cache, which are assumed to be fabricated at a 45 nm technology process. Each simulated core is equipped with private, first level instruction and data caches (32KB/8-ways), and a unified level-two cache (512KB/16-ways). The default x86 gem5 out-of-order CPU model is used. Thirteen different benchmarks from the SPEC2017 suite [31] are simulated using the reference inputs with gem5 to create accurate performance data. The selected workloads include both compute-bound and memory-bound benchmarks. In all cases, the most frequent simpoint sample is simulated. This generated data is fed into McPat to produce the power trace for each workload, which includes 4,000 power samples per benchmark for a duration of four seconds. The power traces are created for each benchmark based on these samples, including both leakage and dynamic power.

Before the temperature data is produced by HotSpot, TAP-2.5D [11] is used to place the chiplets such that the total temperature of the system is minimized. TAP-2.5D utilizes the stochastic method of simulated annealing, hence the results differ between runs for the same set of chiplets. The average transient power of the core and L3 chiplets is used as input to TAP-2.5D, calculated based on the available benchmarks, with different number of core-L3 pairs, ranging between two and seven. Furthermore, the core-L3 pairs are considered independent of each other, each running its own benchmark at a time. Therefore, interconnections only between cores and their respective L3 are assumed. Consequently, a set of benchmarks is chosen for each floorplan, where each set can combine several workload benchmarks multiple times. Ten different

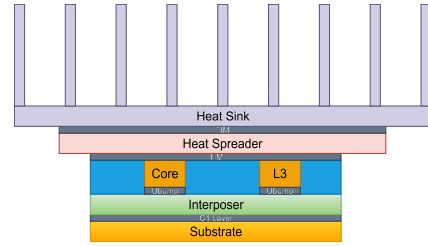


Fig. 2. The complete interposer-based system modeled in HotSpot simulator. The model consists of nine different layers stacked vertically.

floorplans are created with a different number of core-L3 pairs. The final resulting floorplan, determined by TAP-2.5D after a specific number of steps, is used as input to HotSpot along with the power traces to produce the temperature data. For a specific floorplan and power traces resulting from a set of benchmarks, HotSpot generates temperature maps at 1 ms sampling interval.

The dimensions of the core and L3 chiplets are chosen to have aspect ratio of one and 0.15 mm thickness. The interposer dimensions are chosen as 45 mm \times 45 mm \times 0.11 mm, and TAP-2.5D chooses the dimensions of the heat spreader and heat sink at 90 mm \times 90 mm and 180 mm \times 180 mm, respectively. The modeled system is illustrated in Fig. 2, starting from the substrate layer and ending at the aluminum heat sink. TAP-2.5D also generates files that simulate the C4, μ bump and thermal interface material (TIM) layers and a configuration file, used as input in simulations with HotSpot. The fifth layer includes the chiplets mounted on the interposer surrounded by TIM [32]. Our approach assumes passive interposers as no heating in the interposer layer has been considered. Furthermore, in this system, heat is primarily transferred through the heat sink and, hence, the considerably weaker thermal path through the package substrate is not modeled. However, this assumption does not change the applicability of the method, only slightly affects the absolute temperature values.

A resolution of 64 \times 64 is chosen for the thermal maps generated through HotSpot. This resolution appropriately balances the complexity of the transformer model (*i.e.*, number of parameters) and the required resources, while also providing adequately accurate thermal maps that capture the temperature distribution across the interposer system.

IV. PROPOSED SENSOR PLACEMENT AND THERMAL RECONSTRUCTION FLOW

Transformers have emerged as a useful approach for processing sequential data in a variety of applications, such as large language models [33], speech [34], and time series [35], since dependencies in a data sequence can be captured through attention mechanisms [36], [37]. In this section, the proposed transformer framework and sensor placement approach are described in detail. Several variables used in this and the following sections are defined in Table II.

A. Transformer Description

The basic transformer model is illustrated in Fig. 3. The proposed framework consists of a single thermal reconstruction

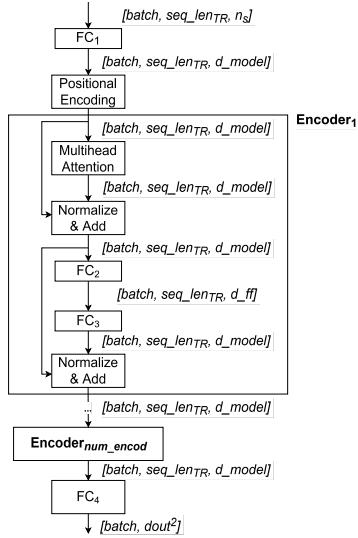


Fig. 3. Schematic of the TR transformer used in the proposed framework. (FC = Fully Connected, batch = size of the batches).

(TR) transformer. The TR transformer has an input size that is equal to $n_S \times seq_len_{TR}$, and starts with positional encoding (PE), since the network cannot process data sequences without crucial information related to the transient order of the data. Thus, this step adds temporal information about each thermal map in the sequence. The method followed is similar to the approach in [36], where sinusoidal vectors are determined and added to the scaled dataset.

The transformer also consists of a stack of encoders, which processes the data after positional encoding. Similarly to [29], the decoders are omitted from the transformer structure. Each encoder includes a multi-head attention mechanism, two normalization-addition modules, and one feed-forward block. The first part of the encoder is the multi-head attention mechanism, which highlights different aspects of the input sequence. The implemented attention mechanism is based on [36]. The attention mechanism aims to calculate different attention scores between the thermal maps and capture multiple dependencies among them. In this way, the transformer exhibits a better understanding of the characteristics of the dataset.

The feed-forward block is responsible for further processing the output of the multi-head attention module. The architecture includes a pair of two fully connected (FC) layers that double the dimension of the data, and then contract them back to the dimension used by the transformer. The normalization-addition block optimizes learning and facilitates the passing of the gradient flow during backpropagation. The TR transformer also includes a FC layer after the encoders, which resizes the data back to the initial dimensions.

B. Framework Analysis

The purpose of this subsection is to present the holistic thermal reconstruction framework that utilizes sensors allocated to the reusable platform to minimize the reconstruction error. This framework is shown in Fig. 4a. The interposer is assumed to comprise n_S thermal sensors, where all thermal sensors are

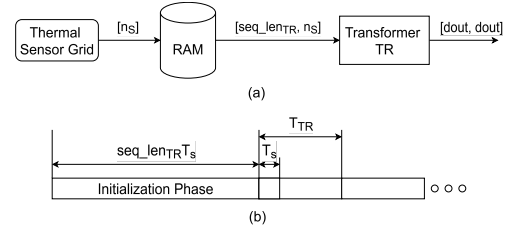


Fig. 4. (a) Schematic of the proposed framework utilizing n_S thermal sensors and (b) timing diagram of its operation.

constantly active capturing temperatures at intervals of T_S . The captured temperatures are stored and are input to the TR Transformer. The required memory is $seq_len_{TR} \times n_S \times Temp_{bits}$ bits, where $Temp_{bits}$ is the number of bits used to represent the temperature. In the experiments of Section V, 32-bit floating point representation is used, but other representations can also be used without loss of generality (*e.g.*, 8-bit integer representation [38]) to reduce the required memory. Once memory is full and, hence, all inputs are available, the TR Transformer provides $dout \times dout$ thermal maps at intervals of T_{TR} .

The operation of this framework is shown in Fig. 4b. Reasonably, initialization is required where some time is allowed for the thermal sensors to capture temperatures and fill the memory. The duration of this interval depends on two factors, the sampling period of the sensors (T_S) and the length of the sequence needed by the transformer (seq_len_{TR}). After this initialization phase, the TR Transformer has all the necessary input data available to produce thermal maps at intervals of T_{TR} . The period T_{TR} must be selected larger than the inference time of the TR Transformer (τ_{TR}), so that the transformer has ample time to produce its output before starting the next iteration. Furthermore, if the product $seq_len_{TR} \times T_S$ is greater than T_{TR} , the subsequent TR Transformer operation can reuse a portion of the thermal map data sequence used in the previous periods. The size of the reused data sequence can be found from the following equation:

$$DS_{reused} = seq_len_{TR} - \frac{T_{TR}}{T_S}. \quad (1)$$

C. Hyperparameter Tuning Flow

Hyperparameter tuning (HT) analysis has been widely used in ML applications to find an optimal selection of hyperparameters that improve the accuracy of a neural network model. The most common tuning techniques include Grid Search, Random Search, and Bayesian Optimization. For our experiments, the Optuna Python library [39] is used with the Tree-structured Parzen Estimator (TPE) sampler [40]. TPE is efficient in handling complex high-dimensional spaces, which satisfies the needs of the sensor allocation problem, as the size of the search space is directly proportional to the number of thermal sensors. Pruning of trials can also be adopted in the methodology to further reduce runtime and increase scalability.

The proposed flow is illustrated in Fig. 5 and is influenced by [41], from which this work adopts the hyperparameter terminology. At first, the hyperparameters are split into three

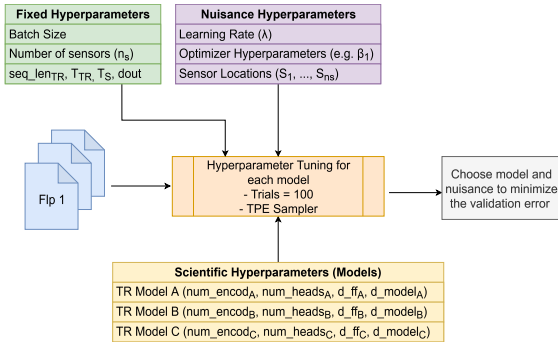


Fig. 5. Proposed flow for solving the combined sensor allocation and thermal map generation problems, influenced by [41]. The figure shows how the hyperparameters are split into the three different groups for our experiments.

TABLE III
EXPERIMENTAL SET OF SCIENTIFIC HYPERPARAMETERS

| Models | d_model | d_ff | num_heads | num_encod |
|---------------|---------|------|-----------|-----------|
| Default Model | 32 | 64 | 2 | 1 |
| Model A | 64 | 128 | 2 | 1 |
| Model B | 32 | 64 | 4 | 1 |
| Model C | 32 | 64 | 2 | 2 |

distinct categories: scientific, nuisance, and fixed. The experiments presented in subsection V-A measure the effect of the scientific hyperparameters on the accuracy of predictions. On the other hand, nuisance hyperparameters also influence the accuracy and, hence, must be tuned for specific values of the scientific hyperparameters each time. The target is to find an optimal set of nuisance hyperparameters for a specific combination of scientific hyperparameters and number of trials. Finally, fixed hyperparameters are considered constant during the experiments to reduce the complexity of the flow.

All experiments are performed according to the following procedure: each unique set of scientific hyperparameters defines a distinct experimental study for which hyperparameter optimization is used to tune the nuisance hyperparameters [41]. Every hyperparameter that influences the architecture of the model and does not affect the dataset (*i.e.*, d_{model} , d_{ff} , num_heads , num_encod) is chosen as a scientific hyperparameter. The learning rate, the Adam optimizer hyperparameter β_1 , which considers the initial decay rate of the gradient [42], and the sensor locations, represented as indexes of the 64×64 grid (4096 locations), are chosen as nuisance. A sensor location can be restricted by the user within a specific area, through a set of indexes as input. The learning rate is adjusted between 10^{-5} and 10^{-3} and β_1 between 0.8 and 0.999. For the fixed hyperparameters, the batch size is equal to 256, based on the size of the dataset. The size of the input sequence (seq_len_{TR}) has been chosen equal to 25 to balance the required memory of Fig. 4a and the model performance, while the reconstruction period T_{TR} has been chosen equal to 10 ms.

V. EXPERIMENTAL RESULTS

The proposed flow has been executed on a Linux server with an AMD EPYC processor and a NVIDIA A100 graphics card,

TABLE IV
HYPERPARAMETER TUNING FLOW RESULTS

| Scenario | Best Model for 100 trials | Minimum MSE in Validation set (K) | Average Inference Time (ms) ^a |
|-----------------------|---------------------------|-----------------------------------|--|
| 8 Sensors Scenario 1 | Model B | 1.87e-3 | 1.15 |
| 8 Sensors Scenario 2 | Model C | 2.13e-3 | 1.90 ^b |
| 16 Sensors Scenario 1 | Model B | 1.08e-3 | 1.18 |
| 16 Sensors Scenario 2 | Default | 1.88e-3 | 1.17 |
| 32 Sensors Scenario 1 | Model C | 1.01e-3 | 1.98 ^b |
| 32 Sensors Scenario 2 | Model A | 1.73e-3 | 1.19 |
| 64 Sensors Scenario 1 | Model B | 8.45e-4 | 1.21 |
| 64 Sensors Scenario 2 | Model B | 1.51e-3 | 1.21 |

^a The training duration for each model was around 6 to 7 hours.

^b The additional encoder increases the inference time.

using Python 3 and Pytorch [43]. The flow is compared with other SotA sensor allocation and reconstruction methods in literature for 8, 16, 32, and 64 sensors. The hyperparameter tuning results are described in subsection V-A. Two scenarios are explored in subsections V-B and V-C, where in scenario 1, the dataset is split for training, validation, and testing such that there is always a portion of the workloads of a floorplan in each of the three sets, while in scenario 2, the dataset has been split such that two floorplans are not included in the training set and are considered unseen during training. During the training process, the model is trained for 500 epochs, where for each epoch the validation loss is calculated to identify potential underfitting or overfitting. The model with the minimum validation loss resulting from the 500-epoch period is used in the testing phase. For each of the ten floorplans, 20 sets of benchmarks are chosen, for a total of 79,000 data points. Two key metrics are used to evaluate the efficacy of predictions in the testing dataset: the root mean square error (RMSE) [29] and the maximum error (MAX) [25], while the mean squared error (MSE) is used for HT and training.

A. Hyperparameter Tuning Results

The models and related scientific hyperparameters used in our experiments are reported in Table III. In the first hyperparameter tuning run for each number of sensors, the default model is used to find the nuisance hyperparameters that minimize the MSE in the validation set. Afterward, hyperparameter tuning is run for three different cases, where each case corresponds to an increase in one of the scientific hyperparameters which control the architectural size of the TR model, to record improvements in the prediction error. After tuning, the model and nuisance hyperparameters with the minimum prediction error across 100 trials are selected.

The results after executing all 32 hyperparameter tunings are listed in Table IV. For most cases, the least MSE in the validation set resulted after choosing Model B, which doubles

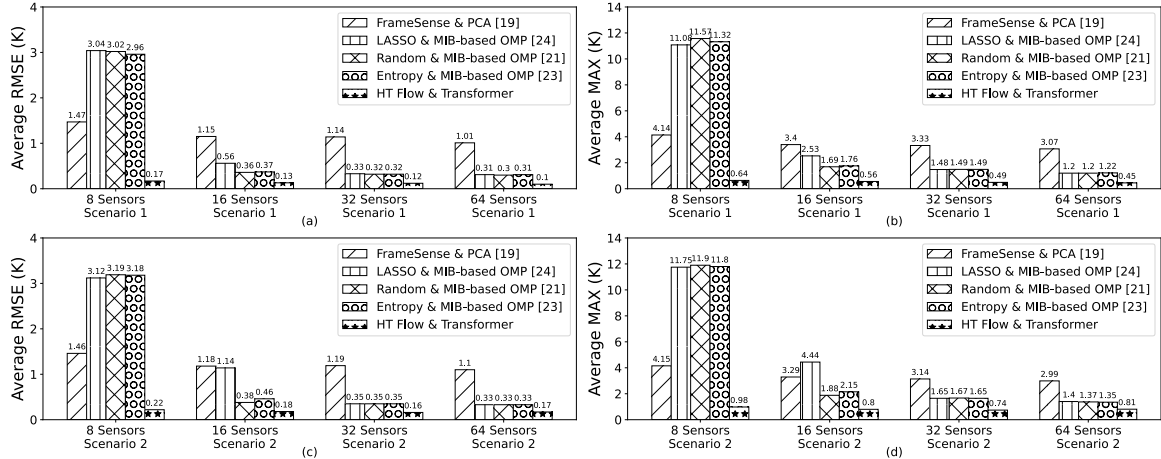


Fig. 6. (a) and (c) show the average RMSE, while (b) and (d) show average MAX errors for the proposed method and SotA, for scenario 1 and 2, respectively.

the number of heads of the transformer, hence improving the attention diversity and the ability of the network to capture the complex temporal relationships. The minimum MSE also decreases as the number of sensors increases, and the prediction error of scenario 2 is much higher, since the validation set consists of floorplans with which the model has not been trained. Furthermore, for the four explored models in Table III, the number of parameters is less than 7 million, which in turn translates into a memory requirement of less than 28 MB for the model parameters. On the other hand, the RAM size in Fig. 4 is equal to a maximum of 6.4 KB for 64 available thermal sensors. This low memory requirement combined with the small architecture makes the model easily portable in hardware. Finally, the model performs inference in less than 2 ms using the Nvidia A100 GPU, which is lower than the operating period of the TR transformer (T_{TR}).

B. Scenario 1: Results on Known Floorplans

For each floorplan the workloads are split as 16:2:2, where 16 are used for training and two for validation and testing. The same training dataset has also been used in the techniques which are compared with our method (e.g., LASSO coefficient calculation). The results of the proposed HT Flow and Transformer approach are compared with FrameSense and principal component analysis (PCA) reconstruction technique [19], entropy-based allocation [23] combined with compressed sensing (CS) that includes the matrix inversion bypass (MIB) orthogonal matching pursuit technique proposed in [22], and random placement [21] with CS. A comparison is also made with the LASSO technique from [24], and although Q-LASSO is shown to have an improved performance over the standard LASSO method, the use of Q-Learning significantly limits the scalability of Q-LASSO for problems with a large number of sensor locations. The average RMSE and MAX for the same testing dataset are illustrated in Fig. 6a and Fig. 6b. Our method achieves up to 85.34% average MAX reduction, compared to FrameSense-PCA, and 94.34% reduction compared to Entropy Placement and CS. The results show that our technique suits reusable interposers used across multiple floorplans, which are

known beforehand, always achieving an average MAX error of less than 1K, even with as few as 8 sensors.

C. Scenario 2: Results on Unseen Floorplans

All workloads from the eight known floorplans are added to the training set, while the testing and validation sets consist only of ten workloads from each of the two unknown floorplans. This scenario investigates the capability of the reconstruction methods (i.e., PCA, CS, transformer) to predict accurate thermal maps with insufficient knowledge from the thermal sensors. The average RMSE and MAX for each explored flow are plotted in Fig. 6c and Fig. 6d, respectively, where most techniques exhibit inferior accuracy compared to the previous scenario. Our method demonstrates a MAX error reduction of 91.76% compared to SotA.

VI. CONCLUSION

The proposed methodology demonstrates several benefits compared to existing sensor placement and thermal map generation methods. This novel thermal monitoring approach for reusable interposers addresses the challenges posed by the intricacies of 2.5D packaging, such as thermal coupling. By leveraging a transformer, the method achieves a twofold objective: enabling efficient thermal monitoring of different chiplet configurations with a small number of sensors and providing sufficiently accurate thermal maps of the interposer during runtime. This new approach is compared with other SotA techniques that focus on a specific chip architecture, demonstrating that both diverse and unseen chiplet floorplans and workloads can be thermally monitored with sufficient accuracy, yielding an average MAX error below 1 K with a moderate number of active sensors and, at the same time, ensuring reuse.

ACKNOWLEDGEMENTS

This work was supported in part by the European Commission through the Horizon Europe Framework Programme for Research and Innovation within the SkyANN Project under Grant 101135729. Results presented in this work have been produced using the Aristotle University of Thessaloniki (AUTH) High Performance Computing Infrastructure and Resources.

REFERENCES

- [1] A. Kannan, N. E. Jerger, and G. H. Loh. 2015. Enabling interposer-based disintegration of multi-core processors. *Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 546–558.
- [2] F. Sheikh, R. Nagisetty, T. Karnik, and D. Kehlet. 2021. 2.5d and 3d heterogeneous integration: emerging applications. *IEEE Solid-State Circuits Magazine*, vol. 13, pp. 77–87.
- [3] S. Chéramy *et al.*, 2021. Active silicon chiplet-based interposer for exascale high performance computing. *International Symposium on VLSI Technology, Systems and Applications*, 1–2.
- [4] K. Meier *et al.*, 2022. Board level reliability study on large antenna-in-package solutions for automotive radar applications. *IEEE Electronics Packaging Technology Conference*, pp. 734–743.
- [5] A. Rao *et al.*, 2021. Towards integrated photonic interposers for processing octave-spanning microresonator frequency combs. *Light, Science & Applications*, vol. 10.
- [6] D. C. Stow, Y. Xie, T. Siddiqua, and G. H. Loh. 2017. Cost-effective design of scalable high-performance systems using active and passive interposers. *IEEE/ACM International Conference on Computer-Aided Design*, pp. 728–735.
- [7] F. Li *et al.*, 2022. Gia: a reusable general interposer architecture for agile chiplet integration. *IEEE/ACM International Conference On Computer Aided Design*, 1–9.
- [8] E. Taheri, S. Pasricha, and M. Nikdast. 2022. Resipi: a reconfigurable silicon-photonic 2.5d chiplet network with pcms for energy-efficient interposer communication. *IEEE/ACM International Conference On Computer Aided Design*, 1–9.
- [9] H. Oprins, Y. Ban, V. Cherman, and J. Van Campenhout. 2020. Thermal aspects of silicon photonic interposer packages. *International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems*, 1–6.
- [10] K. Dhananjay, V. F. Pavlidis, A. K. Coskun, and E. Salman. 2022. High bandwidth thermal covert channel in 3-d-integrated multicore processors. *IEEE Transactions on Very Large Scale Integration Systems*, vol. 30, pp. 1654–1667.
- [11] Y. Ma *et al.*, 2021. Tap-2.5d: a thermally-aware chiplet placement methodology for 2.5d systems. *Design, Automation & Test in Europe Conference & Exhibition*, pp. 1246–1251.
- [12] H. Sun *et al.*, 2023. Chiplet multi-objective optimization algorithm based on communication consumption and temperature. *Electronics*.
- [13] Z. Li *et al.*, 2024. Comprehensive review and future prospects on chip-scale thermal management: core of data center’s thermal management. *Applied Thermal Engineering*.
- [14] N. L. Binkert *et al.*, 2011. The gem5 simulator. *SIGARCH Computer Architecture News*, vol. 39, 1–7.
- [15] S. Li *et al.*, 2009. McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures. In *Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 469–480.
- [16] W. Huang *et al.*, 2006. Hotspot: a compact thermal modeling methodology for early-stage vlsi design. *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, pp. 501–513.
- [17] S. Ogrenç Memik, R. Mukherjee, M. Ni, and J. Long. 2008. Optimizing thermal sensor allocation for microprocessors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 516–527.
- [18] A. N. Nowroz, R. Cochran, and S. Reda. 2010. Thermal monitoring of real processors: techniques for sensor allocation and full characterization. *Design Automation Conference*, pp. 56–61.
- [19] J. Ranieri *et al.*, 2015. Near-optimal thermal monitoring framework for many-core systems-on-chip. *IEEE Transactions on Computers*, vol. 64, pp. 3197–3209.
- [20] K.-C. J. Chen, Y.-H. Chen, and Y.-P. Lin. 2017. Thermal sensor allocation and full-system temperature characterization for thermal-aware mesh-based noc system by using compressive sensing technique. *International Symposium on VLSI Design, Automation and Test*, 1–4.
- [21] K.-C. Chen, H.-W. Tang, C.-H. Wu, and C.-H. Chen. 2022. Thermal sensor placement for multicore systems based on low-complex compressive sensing theory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, pp. 5100–5111.
- [22] E. J. Candès and M. B. Wakin. 2008. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pp. 21–30.
- [23] K.-C. J. Chen and C.-H. Chen. 2022. Entropy-based thermal sensor allocation for temperature-aware multi-core platforms. *IEEE International Symposium on Circuits and Systems*, pp. 2534–2537.
- [24] K.-C. J. Chen and L.-Q. Wang. 2024. Q-learning assisted lasso-based thermal sensor placement for thermal-aware multi-core systems. *IEEE International Symposium on Circuits and Systems*, 1–5.
- [25] X. Li, Z. Li, W. Zhou, and Z. Duan. 2020. Accurate on-chip temperature sensing for multicore processors using embedded thermal sensors. *IEEE Transactions on Very Large Scale Integration Systems*, vol. 28, pp. 2328–2341.
- [26] W. Jin, S. Sadiqbacha, J. Zhang, and S. X.-D. Tan. 2020. Full-chip thermal map estimation for commercial multi-core cpus with generative adversarial learning. *IEEE/ACM International Conference On Computer Aided Design*, 1–9.
- [27] L. Chen, W. Jin, and S. X.-D. Tan. 2022. Fast thermal analysis for chiplet design based on graph convolution networks. *Asia and South Pacific Design Automation Conference*, pp. 485–492.
- [28] S. Sadiqbacha, J. Zhang, H. Amrouch, and S. X.-D. Tan. 2022. Real-time full-chip thermal tracking: a post-silicon, machine learning perspective. *IEEE Transactions on Computers*, vol. 71, pp. 1411–1424.
- [29] J. Lu, J. Zhang, and S. X.-D. Tan. 2023. Real-time thermal map estimation for amd multi-core cpus using transformer. *IEEE/ACM International Conference on Computer Aided Design*, 1–7.
- [30] K. Feng, J. Wang, and M. Tang. 2023. An accurate deep learning-based thermal reconstruction technique for microprocessors using embedded sensors. *IEEE International Conference on Integrated Circuits, Technologies and Applications*, pp. 174–175.
- [31] A. Limaye and T. Adegbiya. 2018. A workload characterization of the spec cpu2017 benchmark suite. *IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 149–158.
- [32] Y. Zhang, T. E. Sarvey, and M. S. Bakir. 2017. Thermal evaluation of 2.5-d integration using bridge-chip technology: challenges and opportunities. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, pp. 1101–1110.
- [33] K. S. Kalyan. 2023. A survey of gpt-3 family large language models including chatgpt and gpt-4. *Natural Language Processing Journal*, vol. 6, 100048.
- [34] S. Latif *et al.*, 2023. Transformers in speech processing: a survey. *ArXiv*, abs/2303.11607.
- [35] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. 2022. A time series is worth 64 words: long-term forecasting with transformers. *ArXiv*, abs/2211.14730.
- [36] A. Vaswani *et al.*, 2017. Attention is all you need. In *Neural Information Processing Systems*.
- [37] D. Soydaner. 2022. Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, vol. 34, pp. 13371–13385.
- [38] B. Nouné *et al.*, 2022. 8-bit numerical formats for deep neural networks. *ArXiv*, abs/2206.02915.
- [39] T. Akiba *et al.*, 2019. Optuna: a next-generation hyperparameter optimization framework. *Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining*.
- [40] S. Watanabe. 2023. Tree-structured parzen estimator: understanding its algorithm components and their roles for better empirical performance. *ArXiv*, abs/2304.11127.
- [41] V. Godbole *et al.*, 2023. Deep learning tuning playbook. Version 1.0. http://github.com/google-research/tuning_playbook.
- [42] D. P. Kingma and J. Ba. 2014. Adam: a method for stochastic optimization. *CoRR*, abs/1412.6980.
- [43] A. Paszke *et al.*, 2019. Pytorch: an imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703.