Fast Thermal Analysis for Chiplet Design based on Graph Convolution Networks

(Invited Paper)

Liang Chen, Wentian Jin, and Sheldon X.-D. Tan
Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA
liangch@ucr.edu, wjin018@ucr.edu, stan@ece.ucr.edu

Abstract— 2.5D chiplet-based technology promises an efficient integration technique for advanced designs with more functionality and higher performance. Temperature and related thermal optimization, heat removal are of critical importance for temperature-aware physical synthesis for chiplets. This paper presents a novel graph convolutional networks (GCN) architecture to estimate the thermal map of the 2.5D chiplet-based systems with the thermal resistance networks built by the compact thermal model (CTM). First, we take the total power of all chiplets as an input feature, which is a global feature. This additional global information can overcome the limitation that the GCN can only extract local information via neighborhood aggregation. Second, inspired by convolutional neural networks (CNN), we add skip connection into the GCN to pass the global feature directly across the hidden layers with the concatenation operation. Third, to consider the edge embedding feature, we propose an edge-based attention mechanism based on the graph attention networks (GAT). Last, with the multiple aggregators and scalers of principle neighborhood aggregation (PNA) networks, we can further improve the modeling capacity of the novel GCN. The experimental results show that the proposed GCN model can achieve an average RMSE of 0.31 K and deliver a 2.6× speedup over the fast steady-state solver of open-source *HotSpot* based on SuperLU. More importantly, the GCN model demonstrates more useful generalization or transferable capability. Our results show that the trained GCN can be directly applied to predict thermal maps of six unseen datasets with acceptable mean RMSEs of less than 0.67 K without retraining via inductive learning.

I. Introduction

2.5D chiplet-based technology becomes a promising integration technique to further extend More's law due to its modular designs, multiple functionalities, high performance, low cost and few manufacturing defects [1]–[3]. The heterogeneous 2.5D systems integrate general-purpose processors and many other specialty chiplets fabricated with different technologies and processes to address the demand for high-performance computing. Compared with 3D stacking technology, the 2.5D chiplet-based design can achieve better thermal dissipation. However, the thermal issue is still the top challenge in the chiplets-based systems because of the increasing power density and reduced thermal conductivity in chiplets, and higher power semiconductors such as III-V technologies in the radio frequency integrated circuits [3]. Therefore, chiplet placement and floor planning has drawn attention recently for reducing the thermal hotspots [2], [4], [5]. Optimization for chiplet placement is a nonlinear iterative process that needs to run thermal simulation several times to obtain the sensitivity matrix. An efficient thermal computational method is highly desired for thermal-aware chiplet placement in the design cycle.

Several commercial software, such as COMSOL and ANSYS, employ finite element method [6] to accurately capture temperature distribution and require a large amount of computational time and memory. To perform a fast thermal analysis, a compact thermal model (CTM) is developed to build lumped thermal resistance network based on the well-known duality between thermal and electric fields [7]–[9]. With the desired levels of abstraction, the thermal resistance network is relatively small so that it can be solved efficiently by leveraging a fast sparse matrix solver called SuperLU [10].

Recently, deep neural networks (DNN)-based approaches have shown great potential for solving partial differential equations (PDEs) with a fast speed and high accuracy. What is more, machine learning (ML)-based models are differentiable and can directly calculate the sensitivity matrix [11], which is essential for optimization problems. Many works learn to predict on-chip thermal map using the ML-based methods [12]–[15], which have the fixed size of input and output. This weakness restricts their applications in new designs, which are not seen in the training dataset. In order to develop a transferable ML-based model, the domain decomposition method (DDM) is employed to divide the whole chip into several small regions (tiles) [16]. The convolutional neural networks (CNN) model is trained on each title and can predict thermal maps on large unseen designs. However, it is very hard to determine an appropriate tile and window size [17].

In this work, we propose a novel graph convolutional networks (GCN) architecture to solve the heat conduction equation for thermal map estimation of 2.5D chiplet-based systems. The CTM is used to transfer the chiplet-based design into a thermal resistance network, which can be viewed as a graph with an inherent structure. The GCN is a transferable model in nature via inductive learning, which means that it can predict thermal maps on large new chiplet-based systems without retraining. Our new contributions are as follows:

- We employ data-driven GCN to estimate the thermal map of 2.5D chiplet-based systems by encoding and extracting physics law in the heat conduction equation. To the best of our knowledge, this is the first work for GCN-based temperature estimation. We develop an algorithm to generate chiplet layouts randomly. Then, their thermal resistance networks, which can be viewed as graphs, are created in open-source *HotSpot* based on the CTM. At the same time, *HotSpot* calculates the ground truth and creates a large dataset to train and test our GCN model.
- We design a novel GCN architecture to perform the node-edge regression task based on the popular GraphSAGE network via inductive learning. To overcome the limitation that GCN can only extract local information by neighboring aggregation, we add, as a global input feature, the total power of all chiplets. This innovative method can extend the GCN model to be capable of modeling both local and global features. Even though we do not know the patterns of global features, we can use a lightweight neural network to learn their patterns. Inspired by CNN, we add skip connection into the GCN to pass the global feature directly across the hidden layers with concatenation operation since the global feature significantly affects the output feature. To integrate the

This work is supported in part by NSF grants under No. CCF-1527324, in part by NSF grants under No. CCF-1816361, in part by NSF grant under No. CCF-2113928 and No. OISE-1854276.

edge embedding feature into the GCN, we propose an edge-based attention mechanism that differs from the node-based attention mechanism in the graph attention networks (GAT).
- To further improve the modeling capacity and transferability of the novel GCN, we leverage the multiple aggregators and scalers of principle neighborhood aggregation (PNA) networks instead of using one single mean aggregator to perform graph convolution operation.

Our experimental results show that the proposed GCN model can achieve an average RMSE of 0.31 K and deliver a 2.6× speedup over the fast steady-state solver of open-source *HotSpot* based on SuperLU. To further demonstrate its transferable capability, the trained GCN model is directly applied to predict thermal maps of two unseen datasets, including different numbers and sizes of chiplets, with almost the same average RMSE of 0.35 K. More importantly, the trained GCN is shown to be able to estimate the thermal maps of four unseen datasets containing different chip sizes with the maximum mean RMSE of 0.67 K, which is an acceptable accuracy since the area of unseen designs is four times larger than that of the training set. PNA can reduce the mean RMSE of the GCN on large unseen designs from 1.29 K to 0.67 K and enhance its transferability. Therefore, compared with other ML-based methods, the proposed GCN thermal model indeed shows more powerful the generalization capability to predict unseen chiplet-based designs or floorplans even without using a tile-based decomposition technique.

The paper is organized as follows: Section II reviews the traditional and ML-based methods for thermal map estimation of the chip. Section III introduces the CTM and defines the input and output features, and graph construction. Inspired by the key ideas of GraphSage, GAT, PNA, and skip connection, we propose a novel GCN architecture in Section IV. Experimental results are presented in Section V. Finally, Section VI concludes this paper.

II. Relevant work

Traditional methods solve the heat conduction equation to estimate the thermal map of the chip by using numerical methods, such as the finite element method that has been integrated into commercial software COMSOL and ANSYS. These methods suffer from a large amount of computation time and memory. To trade off the accuracy and efficiency, many researchers proposed the CTM to perform fast thermal analysis with an acceptable accuracy at desired levels of abstraction [7], [8]. The CTM is to build the lumped thermal resistance and capacitance networks, which have been implemented in open-source *HotSpot* [8], [9]. To further speed up steady-state simulations, a fast sparse matrix solver called SuperLU was integrated into *HotSpot* [9], [10]. Various cooling technologies, such as microchannel liquid cooling and thermoelectric coolers (TEC), have been explored to remove the hot spots for emerging chip systems. To model microchannel, Sridhar *et al.* extended the CTM to develop a transient simulator, called 3D-ICE [18]. Long *et al.* proposed an equivalent thermal circuit model for the TEC to optimize the TEC cooling systems [19]. Choday *et al.* incorporated the TEC model into *HotSpot* thermal simulator, named *HotSpot-TE* [20]. As a result, the CTM is very powerful and can be capable of modeling any chip system as the equivalent thermal circuits. Therefore, *HotSpot* becomes popular in the thermal analysis of different kinds of chips. In the chip design flow, optimization for thermal-aware chiplet placement is a key step to check the thermal hot spots [2], [4], [5]. However, the CTM method needs to be carried out several times to calculate the sensitivity matrix for optimization, which is very time-consuming.

The thermal-aware floor planning drives us to apply DNN-based approaches for solving the heat conduction equation. ML-based methods not only present a faster speed while maintaining high accuracy, but also directly export the sensitivity matrix since the ML-based models are differentiable in nature [11]. Zhang *et al.* employed neural network and linear regression-based methods to predict the thermal response of many temperature sensors on the processors, which are not thermal maps [12]. Sheriff *et al.* applied Long-Short-Term-Memory (LSTM) network to capture dynamic temperature profiles measured by infrared thermal imaging setup [13], [21], [22]. Jin *et al.* took the performance metrics as input to generate full-chip thermal maps by using generative adversarial networks [14]. This kind of works collects the data from the real chip and is not suitable for thermal predictions in the design process. Based on the CTM, Juan *et al.* proposed a learning-based autoregressive model to estimate the thermal map of the target chip [15]. However, this model needs to be retrained when the floorplan of the target chip changes significantly. To provide a transferable ML model, Wen *et al.* divided the whole chip into several small regions (tiles) where DNN-based solvers are applied [16]. However, it is not easy to determine an appropriate tile and window size, which can impact the accuracy and speed of DNN-based models [17]. Chhabria *et al.* [17] performed thermal analysis by using convolutional encoder-decoder networks without the tile-based decomposition method.

Recently, graph neural networks (GNNs) have gained attention and popularity on graph-structured data [23]. Many works leveraged GNN to solve various problems in EDA, such as analog circuit clustering [24], layout parasitic parameters prediction [25], operation delay prediction for FPGA HLS [26], analog IC placement [27], and identifying hierarchical symmetry constraints for analog circuit layout [28] since the circuits can naturally be viewed as graph structures. Compared to previous ML-based methods, GNNs are transferable even though they do not use the tile-based decomposition [16], which means GNN models can predict new designs that are not seen in training and test sets. Hence, we employ GNN to represent thermal resistance networks built by the CTM to estimate the thermal maps of 2.5D chiplet-based systems. Kipf and Welling proposed GCN and defined the graph convolutional operation which is analogous to image convolution [29]. This GCN model can not generalize to unseen nodes since the input is a fixed adjacency matrix to represent the graph. To be transferable, GraphSAGE aggregates the information from a node's local neighborhoods and can predict unseen graphs without retraining via inductive learning [30]. GAT is based on a node-based attention mechanism to focus on the most relevant neighborhood nodes instead of treating each neighborhood node equally [31]. To improve the modeling capacity of the GCN, multiple aggregators and scalers, also called PNA [32], are proposed to extract the neighborhood messages that a single aggregator fails to distinguish.

III. Problem Formulation

This work aims to estimate the thermal map of 2.5D chiplet-based systems using the GCN. We review the well-known CTM to obtain the thermal resistance networks, which can be represented by graphs. Based on the graph representation, we employ open-source *HotSpot* to prepare the dataset.

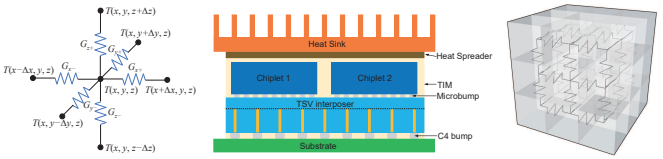
A. Compact thermal model

Based on the duality between thermal and electric fields, many researchers employ the CTM approach to build the equivalent thermal circuits to provide efficient temperature predictions with reasonable accuracy [7]–[9]. The details of the derivation for the CTM are shown below.

The governing equation to describe the steady-state of heat transfer can be written as

$$\nabla \cdot (-\kappa \nabla T) = g \tag{1}$$

where T is the temperature (K), κ is the thermal conductivity (W/(mK)), g is the heat source (W/m^3), and ∇ is the gradient vector operator.



(a) (b) (c)

Fig. 1. (a) Equivalent thermal resistance circuit for a thermal cell. (b) Cross-section view of A 2.5-D chiplet-based system. (c) Cuboid grids of the chip with 27 thermal cells and corresponding equivalent thermal resistance networks.

By applying finite difference approximation, the partial differential equation (1) is written in discretization form:

$$\begin{aligned} &G_{x+}(T(x+\Delta x,y,z)-T(x,y,z))\\ &+G_{x-}(T(x-\Delta x,y,z)-T(x,y,z))\\ &+G_{y+}(T(x,y+\Delta y,z)-T(x,y,z))\\ &+G_{y-}(T(x,y-\Delta y,z)-T(x,y,z))\\ &+G_{z+}(T(x,y,z+\Delta z)-T(x,y,z))\\ &+G_{z-}(T(x,y,z-\Delta z)-T(x,y,z))=g\Delta x\Delta y\Delta z \end{aligned} \tag{2}$$

where Δx, Δy, and Δz are the discretization lengths in x-, y-, and z-directions, respectively. G is the thermal conductance, which is calculated by

$$G_{x\pm}=\kappa_{x\pm}\frac{\Delta y\Delta z}{\Delta x}, G_{y\pm}=\kappa_{y\pm}\frac{\Delta x\Delta z}{\Delta y}, \\ G_{z\pm}=\kappa_{z\pm}\frac{\Delta x\Delta y}{\Delta z} \tag{3}$$

The discretization (2) describes that the total heat entering a junction is equal to the total heat leaving the same junction, which is similar to Kirchhoff's current law. Therefore, the equation (2) can be represented by a thermal resistance circuit for a thermal cell, as shown in Fig. 1(a). $T(x,y,z)$ is the temperature on the center node. $T(x\pm\Delta x,y,z)$, $T(x,y\pm\Delta y,z)$ and $T(x,y,z\pm\Delta z)$ are the temperatures on the six neighborhood nodes of the center node in x-, y-, and z-directions, respectively. Fig. 1(b) shows a cross-section view of a 2.5D chiplet-based system, which can be divided into many layers. To model the 2.5D chiplet-based system, we first mesh each layer with cuboid grids, as shown in Fig. 1(c). Then, based on the meshed thermal cells, we build the equivalent thermal resistance networks by using the formula (2). Each cell is modeled as a node and each node is connected to the nodes of its neighboring cells. Finally, with these nodes, we form the linear matrix equations

$$\mathbf{GT}=\mathbf{P} \tag{4}$$

where $\mathbf{T}$ is the vector to denote all node temperatures (K), $\mathbf{G}$ is the conductance matrix (W/K), and $\mathbf{P}$ is the vector to represent all node heat sources (W). This CTM has been fully implemented in the open-source *HotSpot*. Many researchers have extended the compact thermal model to consider microchannel and thermoelectric cooler, which are the popular and efficient cooling methods [18]–[20]. In summary, thermal resistance networks can model any electronic device with reasonable accuracy.

B. Graph construction with node and edge embedding features

The thermal resistance network can be naturally viewed as a graph, as shown in Fig. 1(c). The node and edge embedding features are illustrated in Table I. When chiplets are operating, power is generated on the node. To consider the global feature, we add the total power of the whole chip into the embedding feature on each node. The conductance on the edges determines the impact of the node on its neighboring nodes. Based on the input information, we predict the temperature for each node. Therefore, we can obtain an undirected graph $\mathcal{G}=(\mathbf{V},\mathbf{E})$ where $\mathbf{V}$ and $\mathbf{E}$ are the sets of the nodes and edges, respectively. The node embedding features of input are the power and total power ($\mathbf{x}_{v1}=P,\mathbf{x}_{v2}=tP,v\in\mathbf{V}$). The edge embedding feature of input is the conductance ($\mathbf{x}_{v,u}=G,(v,u)\in\mathbf{E}$). The node embedding feature of output is the temperature ($\mathbf{z}_v=T,v\in\mathbf{V}$).

TABLE I
INPUT AND OUTPUT FOR GCN MODEL

	Features	Type	Definition
input	P	node	power (W)
	tP	node (global feature)	total power (W)
	G	edge	conductance (W/K)
output	T	node	Temperature (K)

C. Dataset generation

To generate a training and test dataset, we employ *HotSpot* to transfer the 2.5D chiplet-based system into thermal resistance networks. The 2.5D chiplet-based system consists of three layers: heat sink, heat spreader and chiplets. We ignore the TIM, microbump, interposer, C4, and substrate layers because we observe that their temperature profiles are similar to that of the chiplets layer. Chiplets layer contains 4 chiplets with the dimensions of 3×3 mm^2. The area of the whole chip is 12×12 mm^2. We develop an algorithm to randomly place 4 chiplets in the chiplets layer. First, the whole region can be randomly divided into four subregions. Then, one chiplet is placed randomly in each subregion. Each chiplet is assigned with a random power ranging from 1 to 9 W. With the *HotSpot*, we can calculate the ground truth temperature with the mesh of 64×64×3 grids. Each thermal resistance network has 12288 nodes and 32384 edges. The dataset contains 8000 samples (400 chiplets floorplans × 20 power assignments). To validate the knowledge transfer of the proposed GCN, we create six new datasets which are not seen in the training and test sets. The new datasets have different numbers and sizes of chiplets, and different chip sizes. The maximum mesh size among them is up to 128×128×3 grids, which consists of 49152 nodes and 130304 edges.

IV. GRAPH CONVOLUTIONAL NETWORKS

We propose a novel GCN architecture to estimate the thermal map of the chiplets-based system, which can be modeled by thermal resistance networks, as illustrated in Section III. We take the power and total power of the nodes and conductance on the edges as inputs and predict the temperature of the nodes. The new GCN architecture is based on the key ideas of GraphSAGE, GAT, PNA, and skip connection.

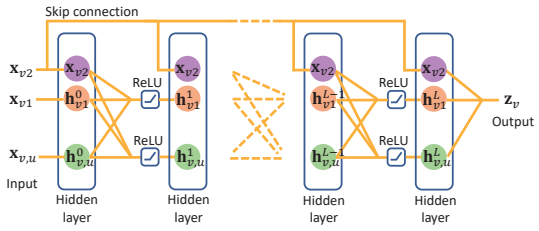


Fig. 2. An architecture of the proposed multi-layer GCN for thermal map estimation.

Fig. 2 shows the overall architecture of the proposed multi-layer GCN, which consists of inputs, output and several hidden

layers to provide deep learning. The inputs are the power $\mathbf{x}_{v1}$ of the node v, the total power $\mathbf{x}_{v2}$ of the chip, and the thermal conductance $\mathbf{x}_{v,u}$ on the edge (v,u). We directly pass the total power $\mathbf{x}_{v2}$ into each hidden layer by concatenating $\mathbf{x}_{v2}$ with node hidden feature $\mathbf{h}_{v1}^{l}$, which is called skip connection via the concatenation. The output is the temperature $\mathbf{z}_v$ of the node v. For each hidden graph convolutional layer, there are two parallel operations, including node features aggregation and edge features update. The node features aggregation is to aggregate neighborhood feature information $\mathbf{h}_{v1}$, edge features $\mathbf{h}_{v,u}$ connected with the neighborhood nodes and the global feature $\mathbf{x}_{v2}$. The edge features update is to build communication from two adjacent nodes v and u to the edge (v,u) based on node features $\mathbf{h}_{v1}$, edge features $\mathbf{h}_{v,u}$ and global feature $\mathbf{x}_{v2}$. To model non-linearity, we add ReLU activations for the hidden layers. This GCN model is applied for the regression task. Therefore, the last layer has no activation function. The proposed graph convolutional layer can be expressed as

$$\begin{aligned}\mathbf{h}_{v}^{l+1} =&\text{ReLU}(\mathbf{W}_4^l(\mathbf{x}_{v2}||\mathbf{h}_v^l||\mathbf{W}_3^l \underset{u\in N(v)}{\oplus}(\boldsymbol{\alpha}_{v,u}\mathbf{W}_1^l\mathbf{h}_u^l)) \\ &+\mathbf{b}_1^l) \end{aligned} \tag{5}$$

$$\mathbf{h}_{v,u}^{l+1} =\text{ReLU}\left(\mathbf{W}_5^l\left(\mathbf{x}_{v2}||\mathbf{h}_v^l||\mathbf{h}_{v,u}^l||\mathbf{h}_u^l\right)+\mathbf{b}_2^l\right) \tag{6}$$

where ReLU$(\cdot)$ is an activation function, node u is the neighborhood of node v, $N(v)$ is the set of neighborhood nodes of the node v, l represents the lth hidden layer, $||$ denotes concatenation [30], [31]. $\mathbf{h}_v$ and $\mathbf{h}_{v,u}$ are node and edge embedding features in the hidden layer, respectively. $\mathbf{W}$ and $\mathbf{b}$ are the learnable weights and biases. Inspired by GAT [31], the coefficient with the attention mechanism is expressed as

$$\begin{aligned}\boldsymbol{\alpha}_{v,u} &= \text{softmax}_u(\text{LeakyReLU}(\mathbf{W}_2^l\mathbf{h}_{v,u}^l)) \\ &= \frac{\exp(\text{LeakyReLU}(\mathbf{W}_2^l\mathbf{h}_{v,u}^l))}{\sum_{u\in N(v)}\exp(\text{LeakyReLU}(\mathbf{W}_2^l\mathbf{h}_{v,u}^l))}\end{aligned} \tag{7}$$

where $\text{softmax}_u(\cdot)$ and LeakyReLU$(\cdot)$ are activation functions. Based on PNA [32], combined aggregation with multiple aggregators and scalers is defined as

$$\underset{u\in N(v)}{\oplus} = \begin{bmatrix} I \\ S(D,\alpha=1) \\ S(D,\alpha=-1)\end{bmatrix} \otimes \begin{bmatrix}\mu \\ \sigma \\ \max \\ \min\end{bmatrix}_{u\in N(v)} \tag{8}$$

where μ, σ, $\max$, and $\min$ are the mean, standard deviation, maximum, and minimum aggregations, respectively. $\otimes$ represents tensor product. $S(D,\alpha)$ is a logarithmic scaler defined as

$$S(D,\alpha) = \left(\frac{\log(D+1)}{\delta}\right)^{\alpha} \tag{9}$$

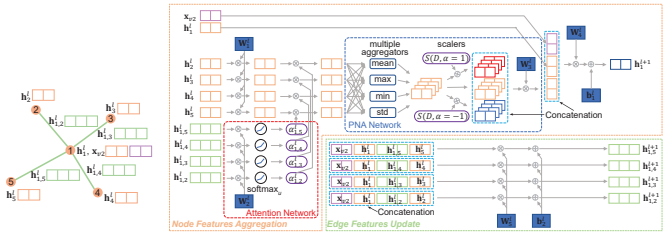
where D is the number of its neigborhood nodes, α is a variable parameter that is -1 for attenuation, 1 for amplification or zero for no scaling,and δ is an average degree of the training set, which is computed by

$$\delta = \frac{1}{N_{\text{train}}}\sum_{i\in\text{train}}\log(D_i+1) \tag{10}$$

where N_{train} is the total number of the nodes in training set. The last layer only has node ouput, which is represented by

$$\mathbf{z}_v = \mathbf{W}_4^L\left(\mathbf{x}_{v2}||\mathbf{h}_v^L||\mathbf{W}_3^L\underset{u\in N(v)}{\oplus}(\boldsymbol{\alpha}_{v,u}\mathbf{W}_1^L\mathbf{h}_u^L)\right)+\mathbf{b}_1^L \tag{11}$$

To understand the hidden graph convolutional layer, we use an example to describe node features aggregation and edge features update, as shown in Fig. 3. The GCN can only aggregate the neighboring information so that it can not model the global features of the whole graph. However, the average temperature of the chip is closely related to the total power of the chiplets, which is a global feature. Due to the lack of considering the pattern of the overall input features for all nodes, GCN fails to predict temperature based on only the power. To mitigate the problem, we take the total power for each graph as the input features for each node. The total power is the summation of the power of all nodes on a graph, which is expressed as



(a) (b)

Fig. 3. (a) One node with the node embedding feature $\mathbf{h}_1^l$ and the global feature $\mathbf{x}_{v2}$ has four neighborhood nodes with the node embedding features $\mathbf{h}_u^l$(u=2, 3, 4, and 5). Four edges are embedded with edge features $\mathbf{h}_{1,u}^l$(u=2, 3, 4, and 5). (b) One hidden graph convolutional layer consists of node features aggregation and edge features update. Node features aggregation is based on concatenating global feature, attention mechanism, and combined aggregation. Edge features are updated by concatenating global feature, edge embedding feature, and its endnode embedding features.

$$\mathbf{x}_{v2} = \sum_{v\in\mathbf{V}}\mathbf{x}_{v1} \tag{12}$$

If we do not know the pattern of the overall input features, we can use a lightweight neural network which is given by

$$\mathbf{x}_{v2} = \text{ReLU}(\mathbf{W}\underset{v\in\mathbf{V}}{||}\mathbf{x}_{v1}+\mathbf{b}) \tag{13}$$

With this global feature, the GCN is capable of modeling the impact of the total power on the average temperature. As GCN becomes deep, the input features pass across many layers and may vanish at the output. Therefore, we concatenate the global feature into each layer to create short paths from the first layer to the last layer, which is similar to the skip connection in the CNN-based DenseNet [33]. GraphSAGE only has node features as input so that it cannot be directly applied for node-edge regression/classification tasks. Hence, to integrate the edge feature into aggregation, we use edge-based attention to compute the coefficient $\boldsymbol{\alpha}_{v,u}$ instead of two adjacent nodes-based attention in the GAT [31]. The attention mechanism is to represent the connection strength between two adjacent nodes. Based on the observation that a single aggregator fails to differentiate between received messages, we employ multiple aggregators and scalers to further increase the modeling capacity and transferability of the GCN, which is the key idea of the PNA model [32]. Mean Square Error (MSE) is used as a loss function for the thermal map regression task.

V. Experimental Results

In this section, the experiments are performed to evaluate the proposed GCN on the dataset which contains 8000 samples. We split the generated dataset into a training set with 6800 samples and a test set with 1200 samples. In addition, we make a great effort to demonstrate the knowledge transfer of the proposed GCN on several datasets with unseen designs, which have different numbers and sizes of the chiplets, and different chip sizes.

All programs, including open source *HotSpot* and the proposed GCN model, are run on a Linux server with Xeon

E5 2.2 GHz CPU and NVIDIA Titan RTX GPU with 24GB memory. The proposed GCN framework is implemented with Deep Graph Library (DGL) on top of the PyTorch platform. The depth of the proposed GCN model is 15 layers where the vector dimensions of node and edge embedding features are set to [1 16 32 64 128 256 512 512 512 256 128 64 32 16 1] and [1 16 32 64 128 256 512 512 512 256 128 64 32 16], respectively. The learning rate of the Adam optimizer is 10^{-4}. To study the modeling capacity and transferability of the PNA, we develop two GCN models, which are GCN+PNA and GCN. The notation "GCN+PNA" represents the proposed GCN model which is illustrated in Fig. 2 and Fig. 3. The notation "GCN" denotes the proposed GCN model which leverages a single mean aggregation instead of PNA aggregation. GCN and GCN+PNA models are trained for 95 and 58 epochs, respectively.

A. Accuracy and speedup of the thermal map prediction

TABLE II
ACCURACY AND SPEED COMPARISON ON TEST SET

Metrics	GCN+PNA	GCN	HotSpot
Max RMSE	**0.80** K	0.93 K	Ground truth
Min RMSE	0.09 K	0.09 K	
Mean RMSE	**0.31** K	0.35 K	
Max AE	**2.07** K	2.54 K	
Mean RMSPE	**0.80**%	0.91%	
Inference Speed	0.1322 s (2.6×)	**0.0719** s (4.8×)	0.3486 s

To demonstrate the accuracy of the proposed GCN model, we calculate the maximum, minimum and mean root-mean-square error (RMSE), max absolute error (AE), and mean root-mean-square percentage error (RMSPE) between predictions and ground truths on the test set with 1200 samples, which are illustrated in Table II. Mean RMSPE is a ratio of mean RMSE to full temperature difference of 38.51 K (=360.61 K−322.10 K). The RMSE of GCN+PNA ranges from 0.09 K to 0.80 K with the mean value of 0.31 K and mean RMSPE of 0.8%. The maximum AE is 2.07 K. As we can see, the GCN+PNA model has slightly better accuracy than the GCN model since PNA can further improve the modeling capacity.

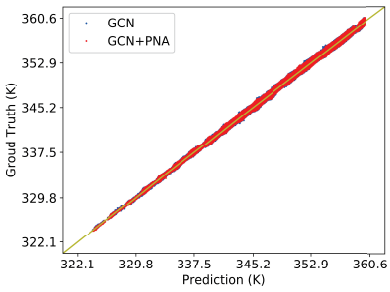


Fig. 4. Temperature predictions vs ground truths on all test cases for GCN and GCN+PNA.

Fig. 4 shows the comparison of the temperature predictions and ground truths on the test set with around 15 million nodes. The node temperatures predicted by both GCN+PNA and GCN models are located close to the yellow line (ground truth). The red dots are plotted on top of the blue dots. It can be observed from Fig. 4 that the red area is slightly smaller than the blue area, which means that the GCN+PNA model is more accurate than the GCN model, but the accuracy is improved a bit by using PNA.

Fig. 5 and Fig. 6 show the comparison of thermal maps predicted by GCN+PNA, GCN and *HotSpot*. There are three

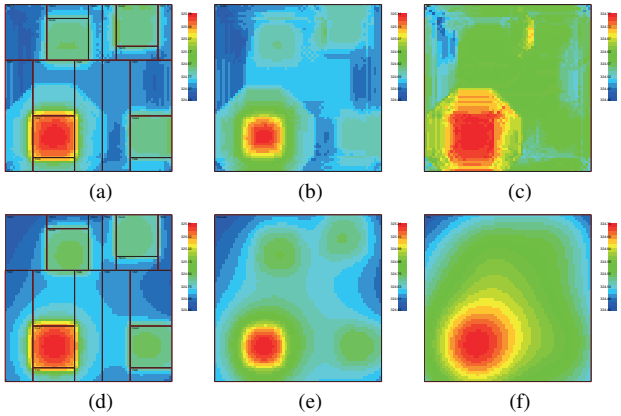


Fig. 5. Thermal maps of (a)(d) chiplets layer, (b)(e) heat spreader layer, and (c)(f) heat sink layer estimated by (a)-(c) GCN+PNA and (d)-(f) *HotSpot*.

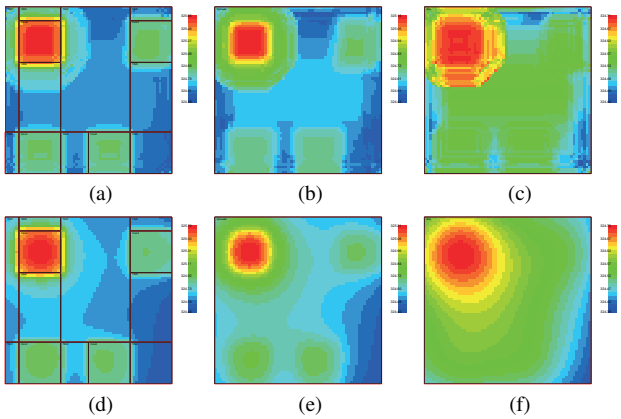


Fig. 6. Thermal maps of (a)(d) chiplets layer, (b)(e) heat spreader layer, and (c)(f) heat sink layer estimated by (a)-(c) GCN and (d)-(f) *HotSpot*.

layers, including chiplets layer, heat spreader layer and heat sink layer. Compared with CNN, GCN can model 3D data easily because CNN needs to perform 3D convolution that is more complicated than 2D convolution. Both GCN+PNA and GCN models can estimate temperature accurately on the hot spots for three layers. The regions we concern about are the hot spots that can lead to thermal reliability problems. Thermal-aware chiplet placement optimization is to reduce the maximum temperature on the hot spots. Therefore, the accuracy of thermal maps is acceptable for optimization problem.

To validate the inference speed of the proposed GCN, we apply the GCN+PNA, GCN and *HotSpot* to capture thermal maps of the test set with 1200 samples. The *HotSpot*-6.0 [9] is sped up by using SuperLU, which is a fast sparse matrix solver [10]. Table II shows the average run time of GCN+PNA and GCN for each design are 0.1322 s and 0.0719 s, respectively, which are 2.6× and 4.8× faster than the execution time of 0.3486 s cost by *HotSpot*. GCN is 1.8× faster than GCN+PNA while they are almost the same accuracy. Therefore, considering accuracy and speed, GCN is superior to GCN+PNA on the test set.

B. Knowledge transfer on several unseen datasets

Compared with CNN, the GCN can be naturally transferable to unseen designs even though GCN does not use the tile-based decomposition method. To demonstrate the transferability of the proposed GCN model, we create three types of unseen datasets with different numbers and sizes of chiplets, and different chip sizes. Each dataset has 1200 samples. We do not need to compare the inference speed for the first two

kinds of new datasets since their chip sizes of 12×12 mm^2 are the same as that of the test set. The last kind of unseen datasets with different chip sizes can also validate the scalability of the proposed GCN model on large unseen graphs.

TABLE III
ACCURACY COMPARISON ON 1200 UNSEEN DESIGNS WITH SIX CHIPLETS

Metrics	GCN+PNA	GCN
Max RMSE	**0.54** K	0.55 K
Min RMSE	0.13 K	**0.12** K
Mean RMSE	**0.27** K	0.29 K
Max AE	**1.73** K	1.88 K
Mean RMSPE	**0.70**%	0.75%

TABLE IV
ACCURACY COMPARISON ON 1200 UNSEEN DESIGNS WITH DIFFERENT SIZES OF CHIPLETS

Metrics	GCN+PNA	GCN
Max RMSE	**0.66** K	0.73 K
Min RMSE	0.10 K	**0.09** K
Mean RMSE	**0.35** K	0.38 K
Max AE	**3.44** K	4.53 K
Mean RMSPE	**0.91**%	0.99%

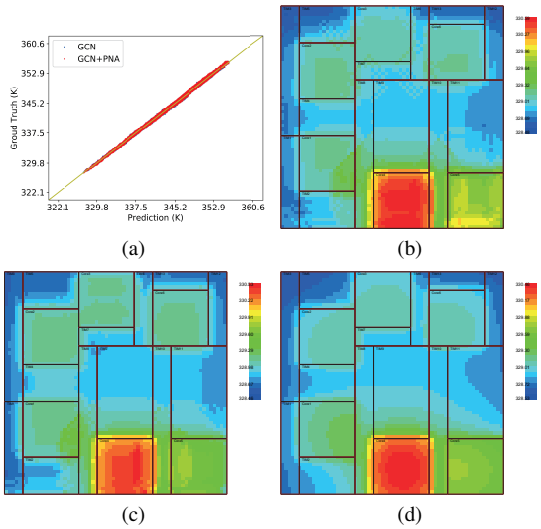


Fig. 7. (a)Temperature predictions vs ground truths on 1200 unseen designs with six chiplets for GCN and GCN+PNA. Thermal maps of chiplets layer on an unseen design with six chiplets estimated by (b) GCN+PNA, (c) GCN and (d) *HotSpot*.

The first kind of new dataset is 1200 unseen designs with six chiplets, as shown in Fig. 7. The sizes of chiplets and chips are 3×3 mm^2 and 12×12 mm^2, respectively, which are the same as that of the test set. It should be noted that the accuracy of GCN+PNA and GCN on this unseen dataset is higher than that on the test set, as shown in Table III. Similar to the test set, the accuracy of GCN+PNA is slightly better than that of GCN on this new dataset. Fig. 7(a) shows that temperatures of all nodes have a very good match between predictions and ground truths. Fig. 7(b)-7(d) show that the thermal maps predicted by GCN+PNA and GCN are almost the same as that of *HotSpot*. Hence, the proposed GCN model shows excellent performance on this unseen dataset.

The second kind of new dataset contains 1200 unseen designs where four chiplets have different sizes, such as 1×1 mm^2, 1×3 mm^2, 3×2 mm^2, and 3×3 mm^2, as shown in Fig. 9. Their chip sizes are still 12×12 mm^2. Table IV illustrates that the average RMSEs of GCN+PNA and GCN on this new dataset are slightly worse than those on the test set. As shown in Fig. 8, considering these designs are more complicated than that on the test set, the accuracy of GCN+PNA and GCN is acceptable. It can be observed from Fig. 9 that GCN+PNA has more accurate temperature predictions on four chiplets than GCN. To sum up, the proposed GCN has relatively good accuracy on this unseen dataset.

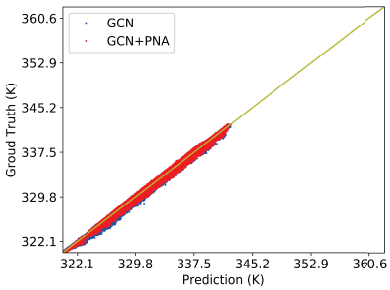


Fig. 8. Temperature predictions vs ground truths on 1200 unseen designs with the different sizes of the chiplets for GCN and GCN+PNA.

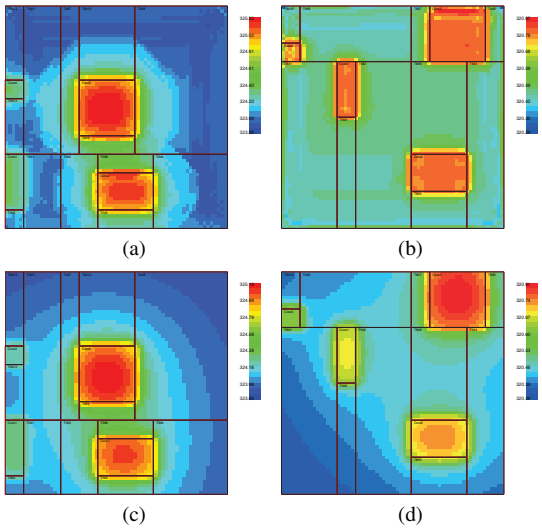


Fig. 9. Thermal maps of chiplets layer on two unseen designs with the different sizes of the chiplets estimated by (a) GCN+PNA, (b) GCN and (c)(d) *HotSpot*.

The third kind of new datasets consists of four sets of 1200 unseen designs. Each set has different chip sizes, such as 15×15 mm^2, 18×18 mm^2, 21×21 mm^2, and 24×24 mm^2. They have four chiplets with the same size (3×3 mm^2). Compared with the previous two kinds of new datasets, these new datasets increase the scale of the chip. The meshes of the four sets are 80×80×3, 96×96×3, 112×112×3, and 128×128×3 grids. The number of nodes increases from 12288 to 49152 and the number of edges increases from 32384 and 130304. The results estimated by GCN+PNA, GCN and *HotSpot* are described in Table V and Fig. 10. The maximum mean RMSE of GCN+PNA on these unseen datasets is 0.67 K, which is twice as high as that of GCN+PNA on the test set. The predictions of GCN+PNA on these new datasets are reasonably accurate since their maximum area is four times larger than that of the test set. However, GCN has relatively poor accuracy for the thermal estimation where the maximum mean RMSE is 1.29 K and the maximum max AE is 7.37 K, which are twice as high as those of GCN+PNA. It can be observed from Table V that the GCN+PNA has better accuracy

TABLE V
ACCURACY AND SPEED COMPARISON ON FIVE SETS OF 1200 UNSEEN DESIGNS WITH DIFFERENT CHIP SIZES

Chip Size (mm^2)	Max RMSE (K)		Min RMSE (K)		Mean RMSE (K)		Max AE (K)		Mean RMSPE (%)		Inference Speed (s)		
	GCN +PNA	GCN	GCN +PNA	GCN	GCN +PNA	GCN	GCN +PNA	GCN	GCN +PNA	GCN	GCN +PNA	GCN	HotSpot
12×12	**0.80**	0.93	0.09	0.09	**0.31**	0.35	**2.07**	2.54	**0.80**	0.91	0.1322 (2.6×)	**0.0719** (4.8×)	0.3486
15×15	**1.16**	1.18	0.10	0.10	**0.42**	0.44	**2.73**	4.70	**1.10**	1.14	0.2070 (2.7×)	**0.1104** (5.1×)	0.5583
18×18	**1.37**	1.47	**0.16**	0.25	**0.56**	0.69	**3.13**	4.49	**1.45**	1.79	0.2984 (2.9×)	**0.1612** (5.4×)	0.8681
21×21	**1.52**	1.78	**0.21**	1.10	**0.67**	1.29	**3.39**	7.37	**1.74**	3.35	0.4058 (2.9×)	**0.2172** (5.4×)	1.1679
24×24	**1.29**	1.4	**0.12**	0.71	**0.59**	0.93	**3.97**	6.88	**1.53**	2.41	0.5285 (2.8×)	**0.2821** (5.3×)	1.4850

than the GCN, especially for large chip size. Fig. 10(a)-10(d) shows that the temperatures of all nodes predicted by GCN+PNA are distributed much closer to the yellow line compared with GCN. Therefore, PNA can further improve the knowledge transfer of GCN on large unseen designs compared to the single mean aggregator. In comparison, *such 3D predictions on large unseen designs will be very difficult, if not impossible, for CNN and other image-based deep neural network*.

VI. CONCLUSION

In this paper, we have proposed a novel GCN architecture to estimate the thermal map of 2.5D chiplet-based systems with the thermal resistance networks, which were built in open-source *HotSpot* based on the CTM. We took the total power of all chiplets as a global input feature, which mitigates the problem that the GCN can only extract local information by the neighborhood aggregation. The proposed GCN framework was based on the key ideas of GraphSAGE, GAT, PNA, and skip connection. The experimental results showed that the proposed GCN model can achieve an average RMSE of 0.31 K and 2.6× speedup over the fast steady-state solver of *HotSpot* based on SuperLU. Furthermore, the trained GCN model can predict two sets of unseen designs, including different numbers and sizes of chiplets with almost the same average RMSE of 0.35 K, which validates the transferable capability of the proposed method. Furthermore, the trained GCN with PNA shows the generalization capability of estimating the thermal maps of the large unseen chip designs containing different chip sizes with the maximum mean RMSE of 0.67 K, a task difficult for CNN and other image-based deep neural network. Therefore, compared with other ML-based methods, the GCN model can be transferable to predict unseen chiplet-based designs even though it does not use the tile-based decomposition technique.

REFERENCES

[1] S. Naffziger, K. Lepak, M. Paraschou, and M. Subramony, "2.2 amd chiplet architecture for high-performance server and desktop products," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2020, pp. 44–45.

[2] S. K. Moore, "Intel's View of the Chiplet Revolution," *IEEE Spectrum*, April 2019, https://spectrum.ieee.org/tech-talk/semiconductors/processors/intels-view-of-the-chiplet-revolution.

[3] F. Herrault, J. C. Wong, Y. Tang, H. Y. Tai, and I. Ramos, "Heterogeneously integrated rf circuits using highly scaled off-the-shelf gan hemt chiplets," *IEEE Microw. Wireless Compon. Lett*, vol. 30, no. 11, pp. 1061–1064, 2020.

[4] A. Coskun, F. Eris, A. Joshi, A. B. Kahng, Y. Ma, A. Narayan, and V. Srinivas, "Cross-layer co-optimization of network design and chiplet placement in 2.5-d systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 5183–5196, 2020.

[5] Y. Ma, L. Delshadtehrani, C. Demirkiran, J. L. Abellán, and A. Joshi, "Tap-2.5d: A thermally-aware chiplet placement methodology for 2.5d systems," in *Proc. Design, Automation and Test In Europe. (DATE)*, 2021, pp. 1–6.

[6] R. W. Lewis, P. Nithiarasu, and K. N. Seetharamu, *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. John Wiley & Son, 2004.

[7] M.-N. Sabry, "Compact thermal models for electronic systems," *IEEE Trans. Compon. Packaging Technol.*, vol. 26, no. 1, pp. 179–185, 2003.

[8] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[9] R. Zhang, M. R. Stan, and K. Skadron, "Hotspot 6.0: Validation, acceleration and extension," University of Virginia, Tech. Rep., CS-2015-04.

[10] X. S. Li, "An overview of superlu: Algorithms, implementation, and user interface," *ACM Trans. Math. Softw.*, vol. 31, no. 3, p. 302–325, Sep. 2005.

[11] H. Zhou, W. Jin, and S. X.-D. Tan, "GridNet: Fast Data-Driven EM-Induced IR Drop Prediction and Localized Fixing for On-Chip Power Grid Networks," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20, Nov. 2020, pp. 1–9.

[12] 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.

[13] 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.

[14] 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.

[15] D.-C. Juan, H. Zhou, D. Marculescu, and X. Li, "A learning-based autoregressive model for fast transient thermal analysis of chip-multiprocessors," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, 2012, pp. 597–602.

[16] J. Wen, S. Pan, N. Chang, W.-T. Chuang, W. Xia, D. Zhu, A. Kumar, E.-C. Yang, K. Srinivasan, and Y.-S. Li, "Dnn-based fast static on-chip thermal solver," in *Proc. Semiconductor Thermal Meas., Modeling Manage. Symp. (SEMI-THERM)*, 2020, pp. 65–75.

[17] 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.

[18] A. M. Sridhar, A. Vincenzi *et al.*, "3D-ICE: Fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*. IEEE Press, 2010, pp. 463–470.

[19] J. Long and S. O. Memik, "A framework for optimizing thermoelectric active cooling systems," in *Proc. Design Automation Conf.*, 2010, pp. 591–596.

[20] S. H. Choday, K.-W. Kwon, and K. Roy, "Workload dependent evaluation of thin-film thermoelectric devices for on-chip cooling and energy harvesting," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2014, pp. 535–541.

[21] 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.

[22] 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.

[23] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[24] K. Settaluri and E. Fallon, "Fully automated analog sub-circuit clustering with graph convolutional neural networks," in *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2020, pp. 1714–1715.

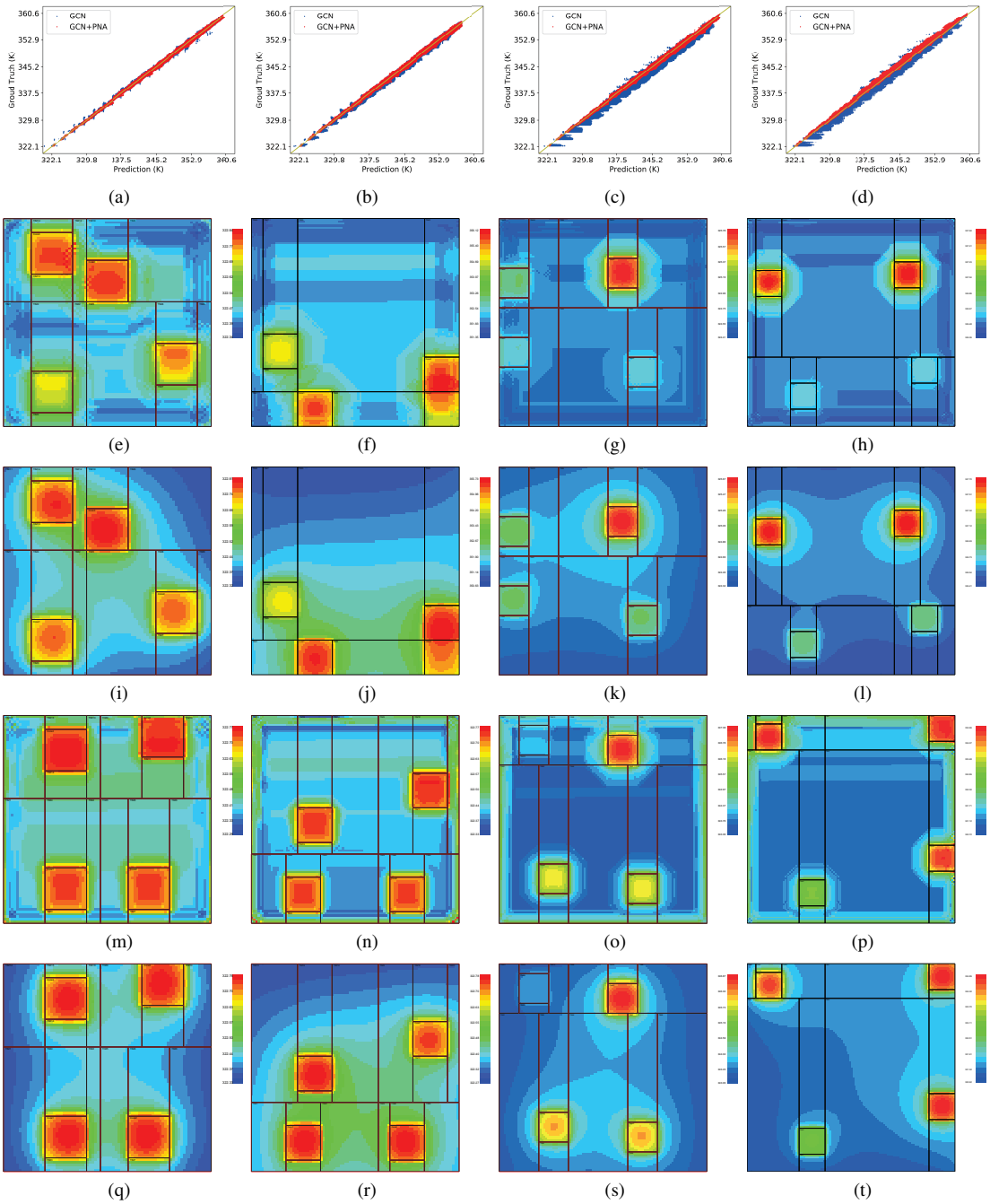


Fig. 10. Temperature predictions vs ground truths on four sets of 1200 unseen designs with chip sizes of (a) 15×15 mm^2, (b) 18×18 mm^2, (c) 21×21 mm^2, (d) 24×24 mm^2 for GCN and GCN+PNA. Thermal maps of chiplets layer on unseen designs with chip sizes of (e)(i)(m)(q) 15×15 mm^2, (f)(j)(n)(r) 18×18 mm^2, (g)(k)(o)(s) 21×21 mm^2, (h)(l)(p)(t) 24×24 mm^2 estimated by (e)-(h) GCN+PNA, (m)-(p) GCN and (i)-(l) (q)-(t) *HotSpot*.

[25] H. Ren, G. F. Kokai, W. J. Turner, and T. S. Ku, "Paragraph: Layout parasitics and device parameter prediction using graph neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[26] E. Ustun, C. Deng, D. Pal, Z. Li, and Z. Zhang, "Accurate operation delay prediction for fpga hls using graph neural networks," in *2020 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[27] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, and W. Xu, "A customized graph neural network model for guiding analog ic placement," in *2020 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[28] K. Kunal, J. Poojary, T. Dhar, M. Madhusudan, R. Harjani, and S. S. Sapatnekar, "A general approach for identifying hierarchical symmetry constraints for analog circuit layout," in *2020 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2020, pp. 1–9.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representation*, 2017.

[30] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 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, pp. 1024–1034.

[31] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2017, pp. 1–12.

[32] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 13 260–13 271.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.