Fast Thermal Analysis for Chiplet Design based on Graph Convolution Networks

Liang Chen, Wentian Jin, and Sheldon X.-D. Tan
Department of Electrical and Computer Engineering
University of California, Riverside

27th Asia and South Pacific Design Automation Conference
ASP-DAC 2022

Contents

1 Motivation

2 Proposed GCN Networks

3 Data Preparation

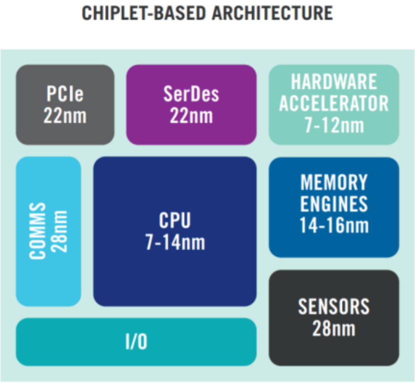
4 Experimental Results

5 Conclusions

Emerging chiplet design face many challenges

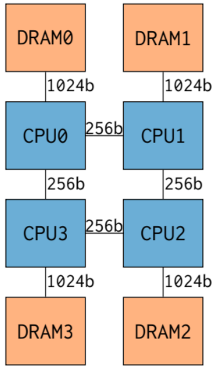
2.5-D Chiplet-based Systems

A 6 chiplets design[1]

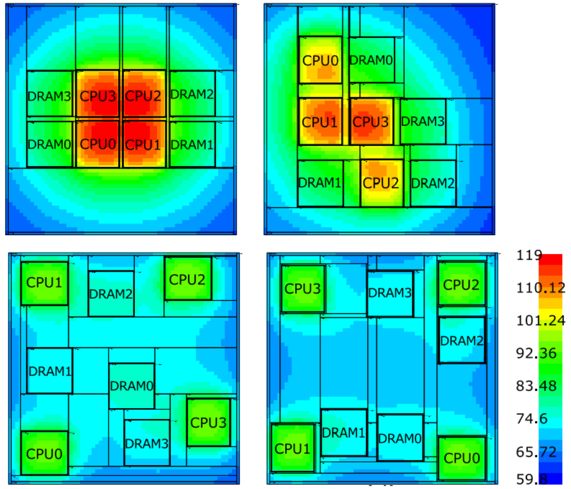


Top view

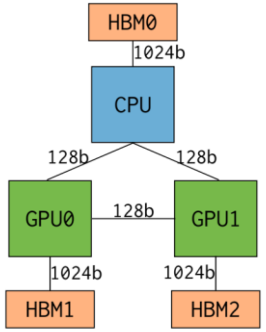
Optimization for Chiplet Placement in 2.5-D Systems



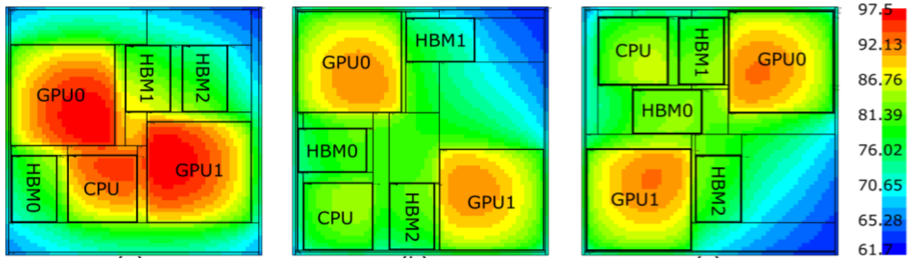
Network topologies of CPU-DRAM System [2]



Thermal maps for different chiplet placements [2]



Network topologies of Multi-GPU System [2]



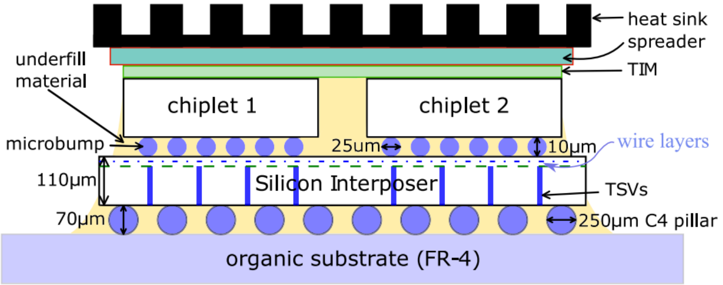
Thermal maps for different chiplet placements [2]

[1] P. Vivet *et al.*, "IntAct: A 96-Core Processor With Six Chiplets 3D-Stacked on an Active Interposer With Distributed Interconnects and Integrated Power Management," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 79-97, Jan. 2021.

[2] Y. Ma. Cross-layer design of thermally-aware 2.5 D systems. Diss. Boston University, 2020.

Existing thermal analysis for chiplet design

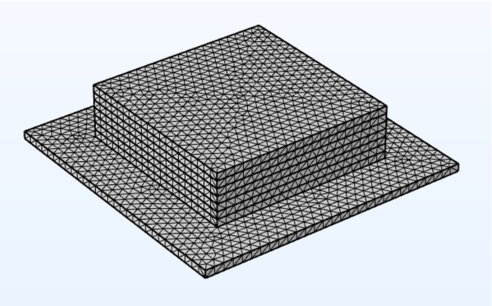
Thermal Map estimation for Chiplet based Systems



Cross-section view of chiplet systems[3]

Thermal simulation methodology

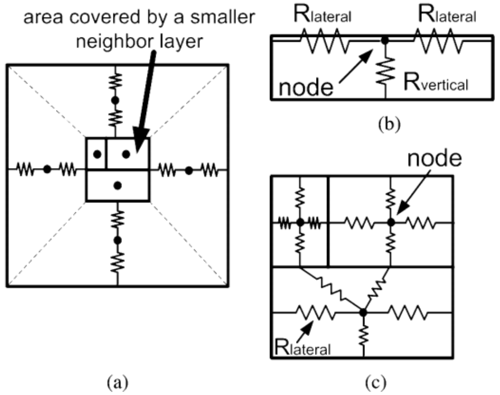
- Finite element method (COMSOL)



Tetrahedral Mesh

$$\sum_i T_i \int_\Omega k\nabla\psi_i \cdot \nabla\psi_j dV + \sum_i \int_{\partial\Omega} (-kT_i\nabla\psi_i) \cdot \mathbf{n}\psi_j dS = \int_\Omega g \left(\sum_i T_i\psi_i\right) \psi_j dV$$

- Compact Thermal Model (Hotspot)



Thermal resistance networks[4]

$$\nabla \cdot (-\kappa \nabla T) = g$$

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

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

[3] A. Coskun *et al.*, “Cross-Layer Co-Optimization of Network Design and Chiplet Placement in 2.5-D Systems,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 12, pp. 5183-5196, Dec. 2020.

[4] 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. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

Several ML-based analysis proposed recently

Machine learning for thermal estimation

Advantage

- Differentiable and directly provide the sensitivity matrix
- Fast speed with high accuracy

Machine learning methods for thermal simulation

- Neural Networks[5][6]
- Long-Short-Term-Memory Networks(LSTM) [7]
- Generative adversarial networks (GAN) [8]
- Convolutional Neural networks (CNN) [9]

[5] K. Zhang *et al.*, “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.

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

[7] S. Sadiqbatcha *et al.*, “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.

[8] 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.

[9] V. A. Chhabria *et al.*, “Thermal and ir drop analysis using convolutional encoderdecoder networks,” *in Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, 2021, pp. 690–696.

Challenges facing existing solutions

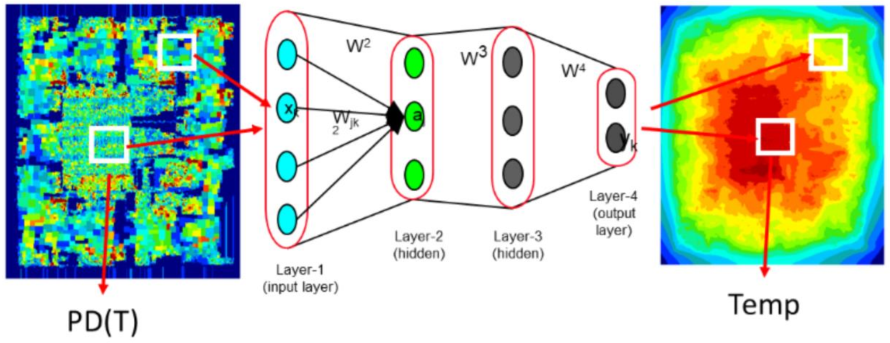
Machine learning for thermal estimation

Disadvantage

- Not transferable (The machine learning models fail to predict the unseen design)

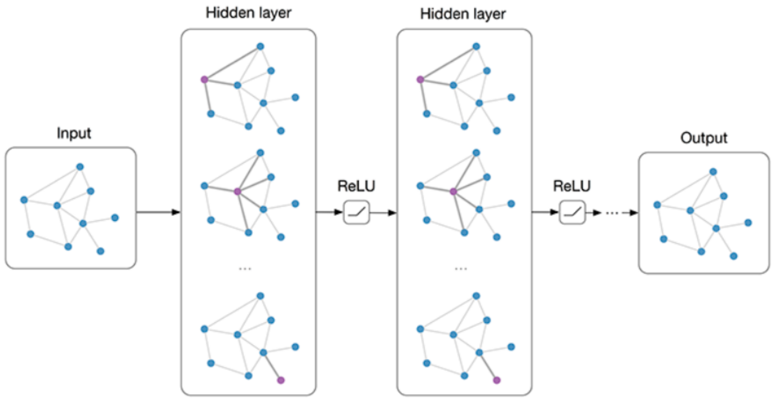
Make machine learning methods transferable

- Tile-based methods[10]



- It is very difficult to determine the tile size

- Graph convolutional networks (GNN)[11]



Multi-layer Graph Convolutional Network (GCN) with first-order filters.

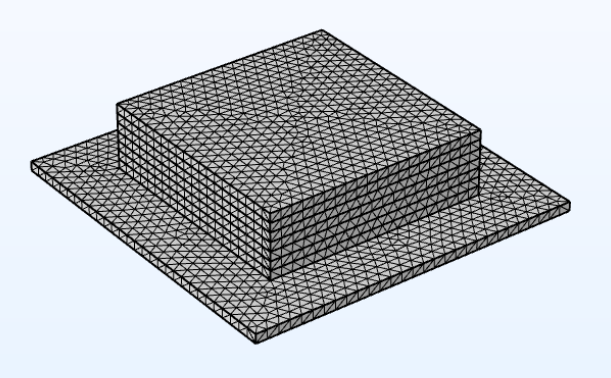
[10] J. Wen *et al.*, "DNN-based fast static onchip thermal solver," in *Proc. Semiconductor Thermal Meas., Modeling Manage. Symp. (SEMI-THERM)*, 2020, pp. 65–75.

[11] W. Hamilton *et al.*, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 1024–1034.

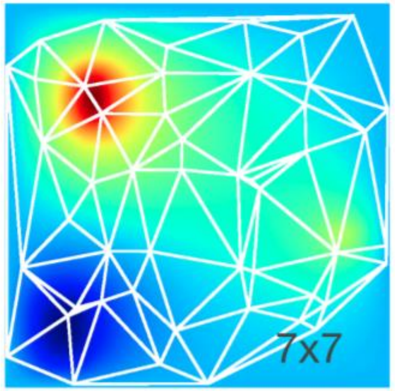
Numerical methods starts with mesh/graph representation of the problem

We can leverage existing numerical analysis framework for new ML-based approach

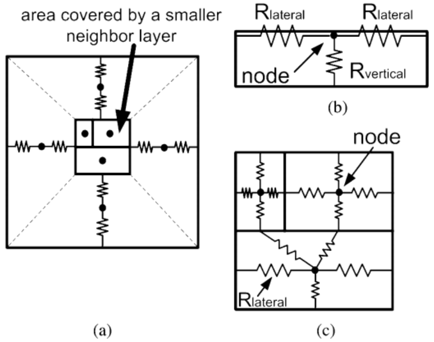
Graph in numerical method



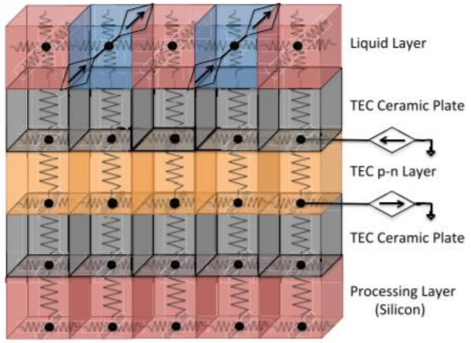
Tetrahedral Mesh



Graph to represent geometry[12]



Thermal resistance networks[4]



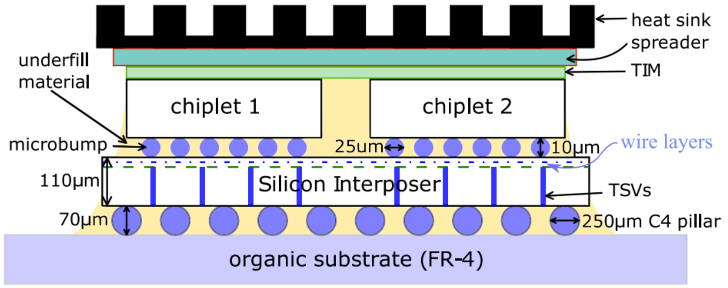
Networks for TEC and microchannel[13]

[12] F. Alet *et al*. "Graph element networks: adaptive, structured computation and memory." *International Conference on Machine Learning*, 2019.

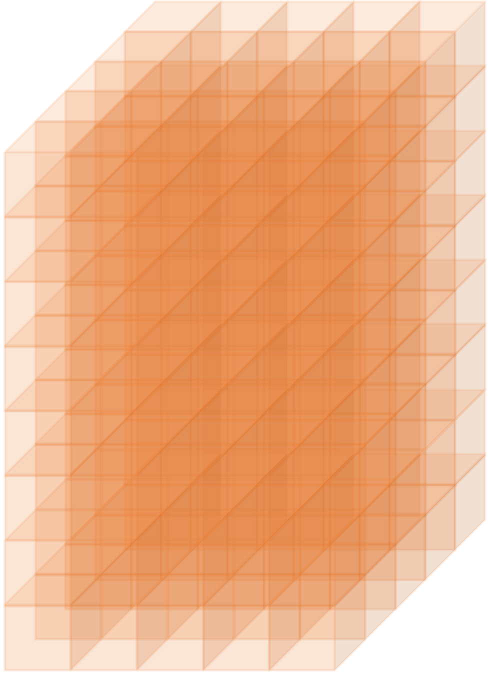
[13] F. Kaplan, M. Said, S. Reda and A. K. Coskun, "LoCool: Fighting Hot Spots Locally for Improving System Energy Efficiency," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 4, pp. 895-908, April 2020.

Compat thermal model (HotSpot) for chiplet thermal analysis

We use thermal resistance networks from HotSpot to represent chiplet



Cross-section view of chiplet systems



Chip

Heat Sink

Heat Spreader

TIM

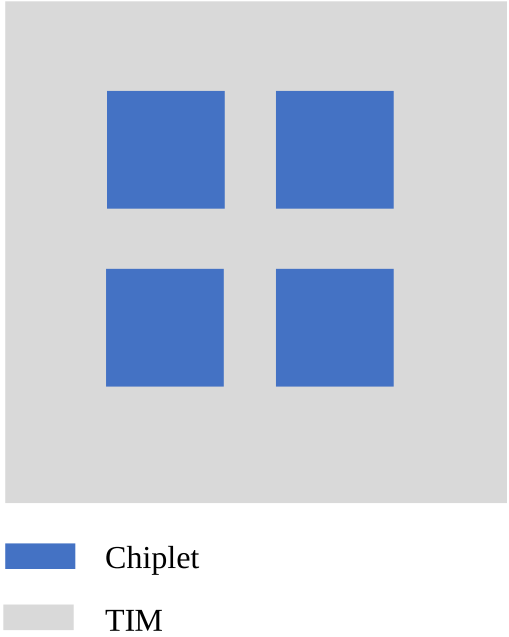
Chiplets →

Microbump

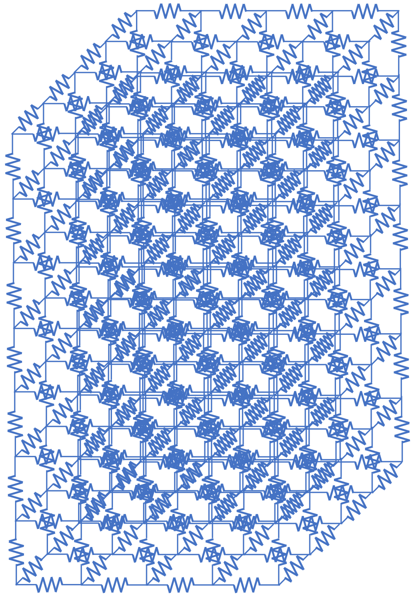
TSV Interposer

C4 bump

FR4



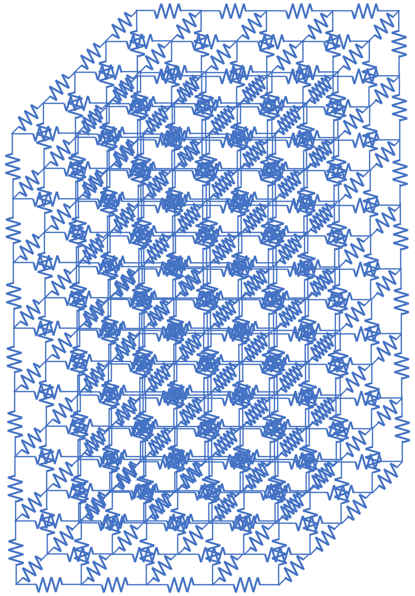
Chiplets Layers



Thermal resistance networks in Hotspot

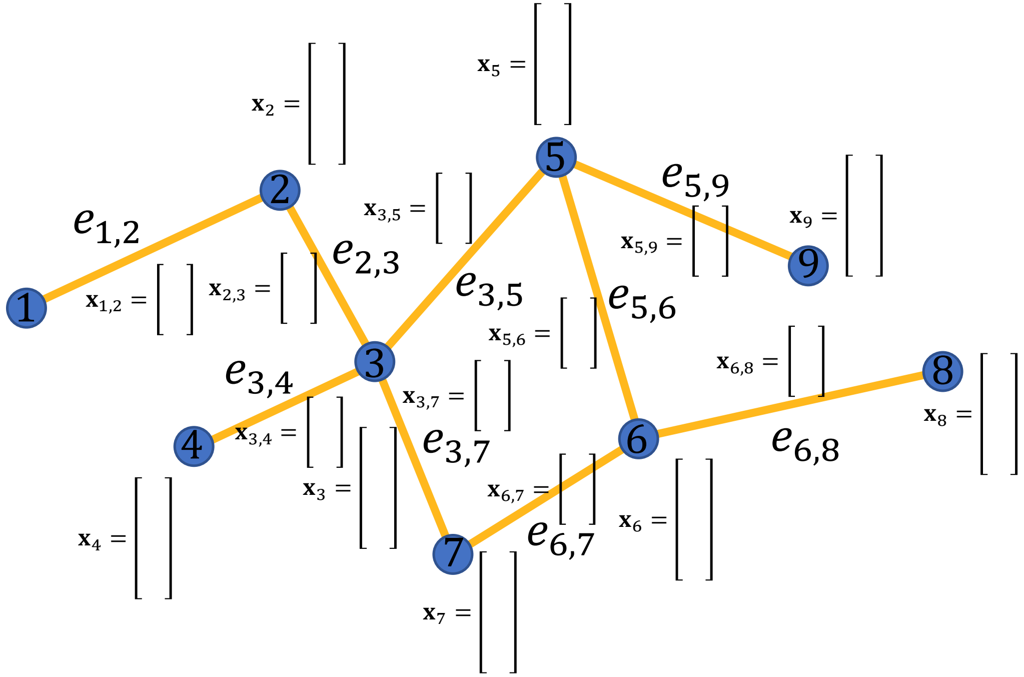
Graph representation of thermal resistance networks for chiplet thermal analysis

- Compact Thermal Model (Hotspot)



Thermal resistance networks in Hotspot

- Graph Structure

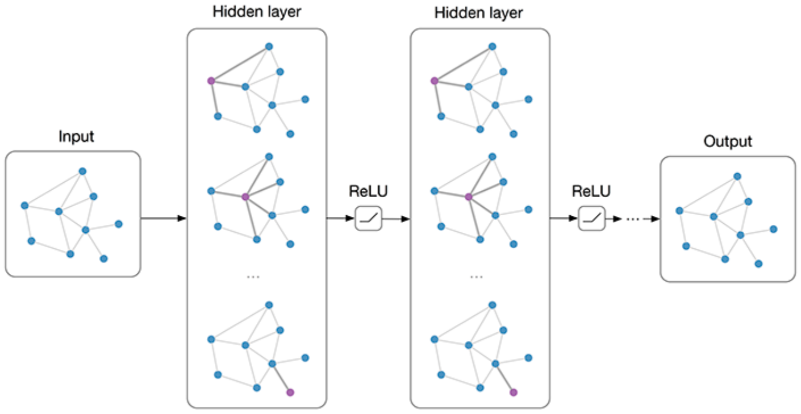


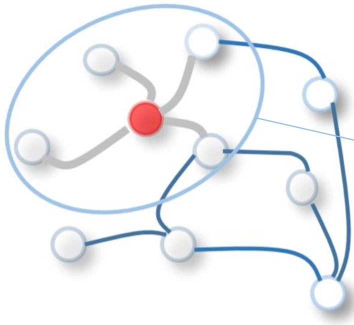
Graph $G=(V, E)$

- Undirected Graph
- Node embedding features (Power, Temperature)
- Edge embedding features(Resistance)

Several new techniques of graph neural networks

- Graph Convolution Networks[11]

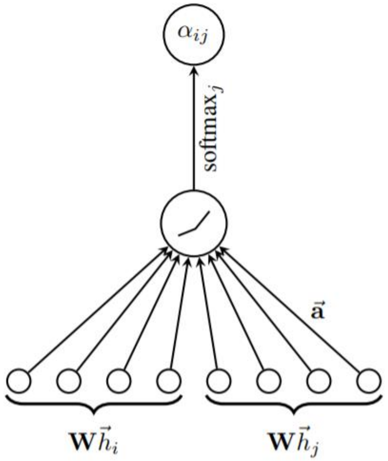


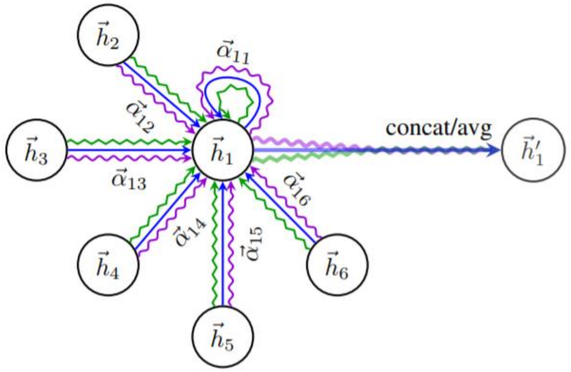


$$\mathbf{h}_v = f(\boldsymbol{\Theta}^T(\sum_{u \in \{N(v) \cup v\}} \bar{A}_{v,u}\ \mathbf{x}_u))$$

Convolution Operation

- Graph Attention Networks[14]





$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)} \qquad \vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\vec{h}_j\right).$$

Attention coefficient to consider the connection effect

[14] P. Velickovic *et al.*, “Graph attention networks,” in *Proc. ICLR*, 2017, pp. 1–12.

Several new techniques of graph neural networks

- Principal Neighborhood Aggregation[15]

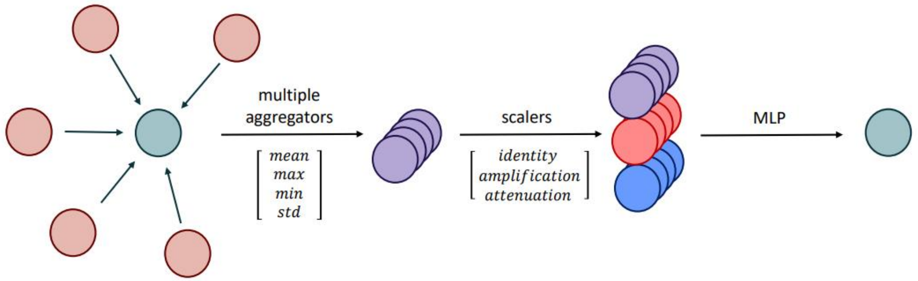
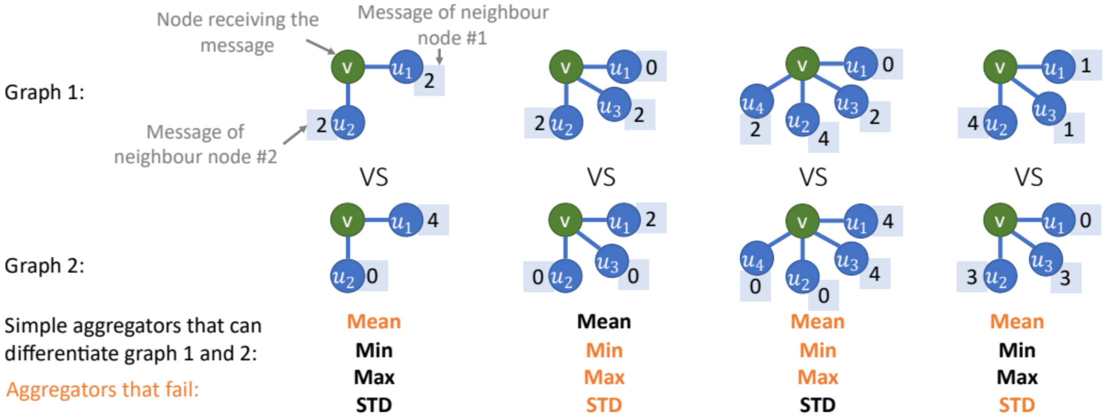


Diagram for the Principal Neighbourhood Aggregation(PNA)

$$\bigoplus = \underbrace{\begin{bmatrix} I \\ S(D,\alpha=1) \\ S(D,\alpha=-1) \end{bmatrix}}_{\text{scalers}} \otimes \underbrace{\begin{bmatrix} \mu \\ \sigma \\ \max \\ \min \end{bmatrix}}_{\text{aggregators}}$$

PNA Operator


Node receiving the message
Message of neighbour node #1
Message of neighbour node #2
Graph 1:
VS
VS
VS
VS
Graph 2:
Simple aggregators that can differentiate graph 1 and 2:
Aggregators that fail:
Mean
Min
Max
STD
Mean
Min
Max
STD
Mean
Min
Max
STD
Mean
Min
Max
STD

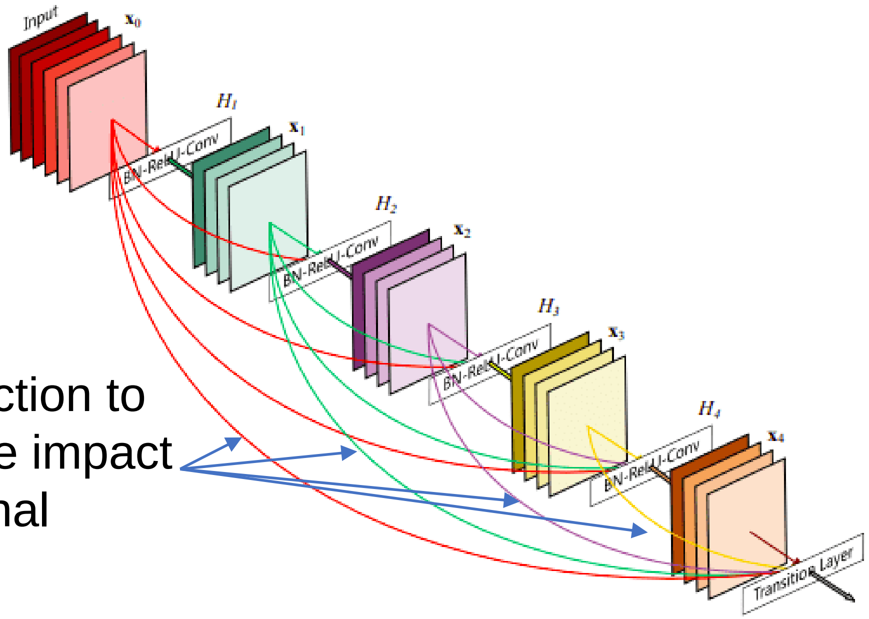
Examples where, for a single GNN layer and continuous input feature spaces, some aggregators fail to differentiate between neighborhood messages.

[15] G. Corso, L. Cavalleri, D. Beaini, P. Li `o, and P. Veliˇckovi´c, “Principal neighbourhood aggregation for graph nets,” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 13 260–13 271.

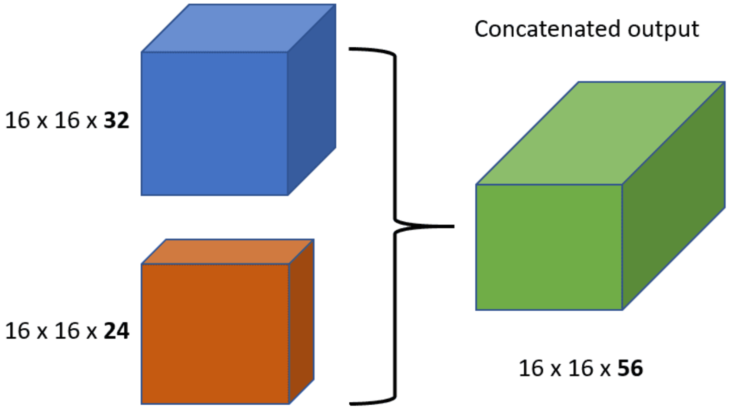
Several new techniques of graph neural networks

- Skip Connection via Concatenation[16]

Skip connection to enhance the impact of input signal



DenseNet

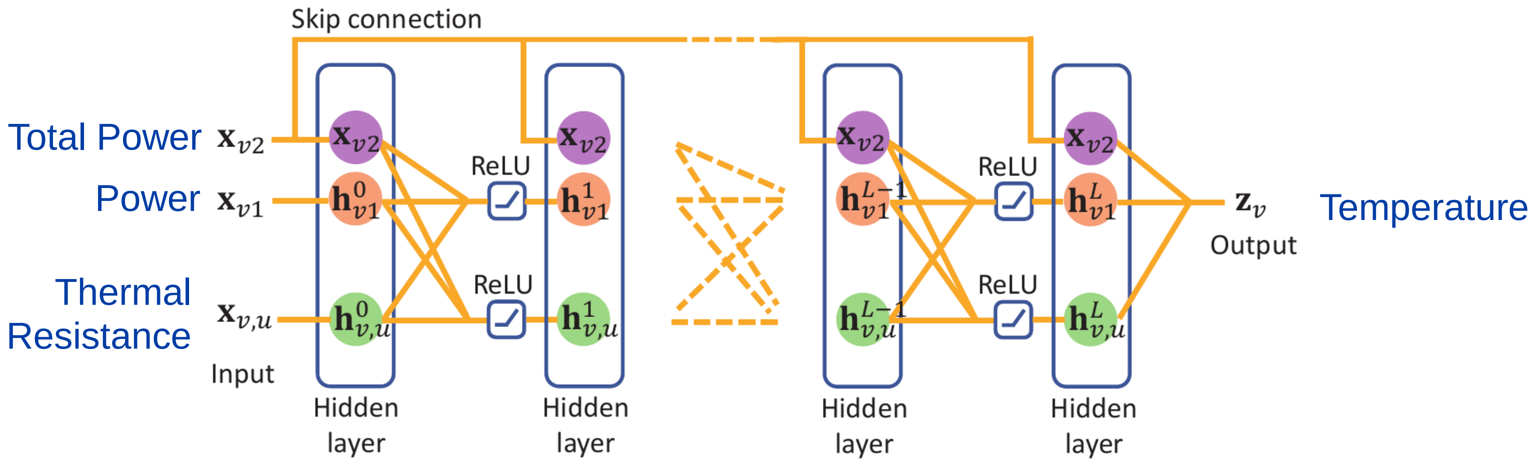


Concatenation Operation

[16] 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.

Proposed graph convolution neural framework for chiplet thermal analysis

Proposed graph convolution network architecture



Formula for the proposed GCN architecture

Skip Connection PNA Operator Attention Mechanism

Input Layer:

$$\mathbf{h}_v^0 = \mathbf{x}_{v1}$$

$$\mathbf{h}_{v,u}^0 = \mathbf{x}_{v,u}$$

*l*th hidden Layers:

$$\mathbf{h}_v^{l+1} = \mathrm{ReLU}(\mathbf{b}^l + \mathbf{W}^l(\boxed{\mathbf{x}_{v2}||\mathbf{h}_v^l}|| \boxed{\begin{bmatrix} I \\ S(D,\alpha=1) \\ S(D,\alpha=-1) \end{bmatrix} \otimes \begin{bmatrix} \mu \\ \sigma \\ max \\ min \end{bmatrix}} \boxed{\alpha_{v,u}}\mathbf{h}_u^l)$$

$$\alpha_{v,u} = \frac{\exp(LeakyReLU(\mathbf{a}^T\mathbf{h}_{v,u}^l))}{\sum_{u\in N(v)} \exp(LeakyReLU(\mathbf{a}^T\mathbf{h}_{v,u}^l))}$$

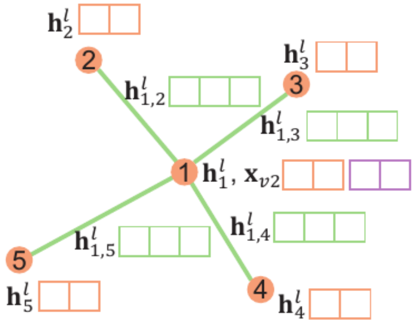
$$\mathbf{h}_{v,u}^{l+1} = \mathrm{ReLU}(\mathbf{b}^l + \mathbf{W}^l(\mathbf{x}_{v2}||\mathbf{h}_v^l||\mathbf{h}_u^l||\mathbf{h}_{v,u}^l))$$

Output Layer:

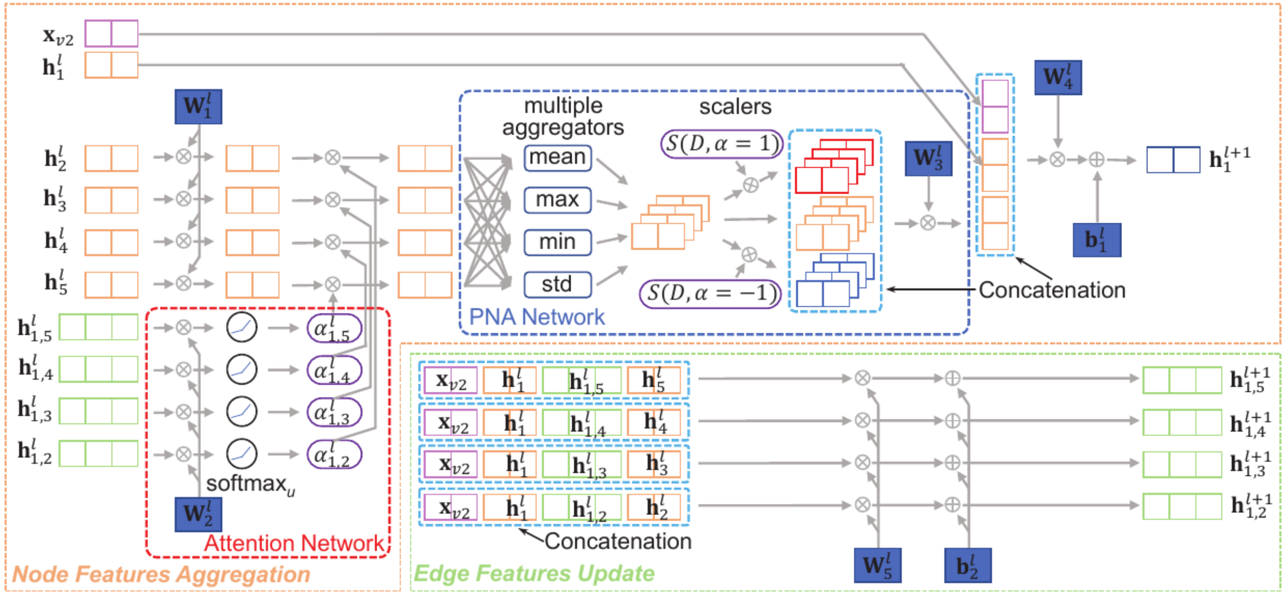
$$\mathbf{z}_v = \mathbf{b}^L + \mathbf{W}^L(\mathbf{x}_{v2}||\mathbf{h}_v^L|| \begin{bmatrix} I \\ S(D,\alpha=1) \\ S(D,\alpha=-1) \end{bmatrix} \otimes \begin{bmatrix} \mu \\ \sigma \\ max \\ min \end{bmatrix} \alpha_{v,u}\mathbf{h}_u^l)$$

Proposed graph convolution neural framework for chiplet thermal analysis

An example to demonstrate the operation of one proposed GCN layer



One node with four neighborhood nodes



Node feature aggregation and edge features update for the example

Highlight Features:

- Use global information (total power) as input.
- Apply the skip connection in graph convolution network
- Integrate PNA network into the model
- Use edge based attention network to represent the connection effect

Data format to be generated for proposed GCN method

Inputs and Outputs

Inputs(Features)

Undirected Graph

Node features

(NodeID, Power, totalPower)

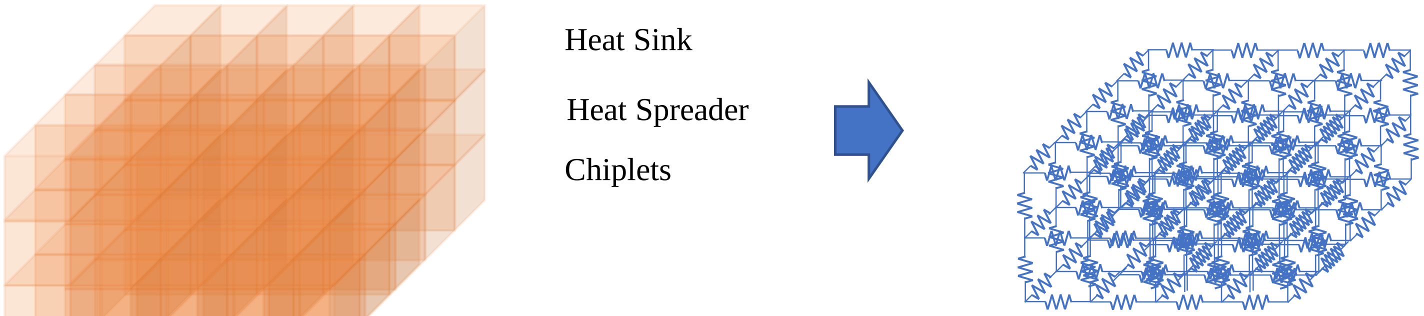
Edge features

(EdgeID, Thermal resistance)

Outputs(Label)

Node Label

(NodeID, Temperature)



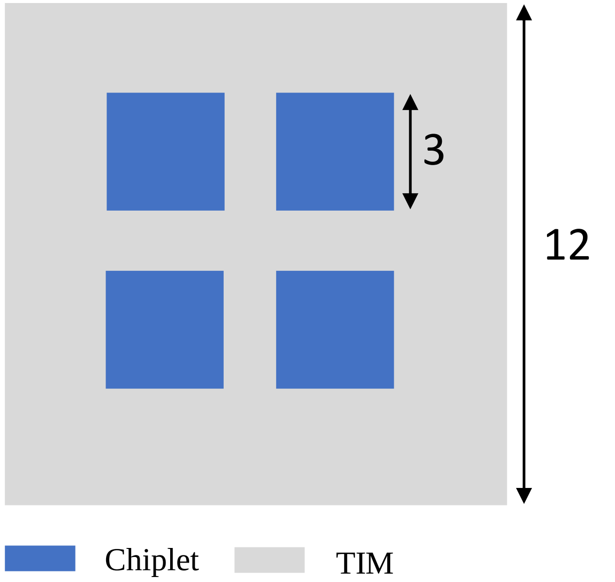
64×64×3

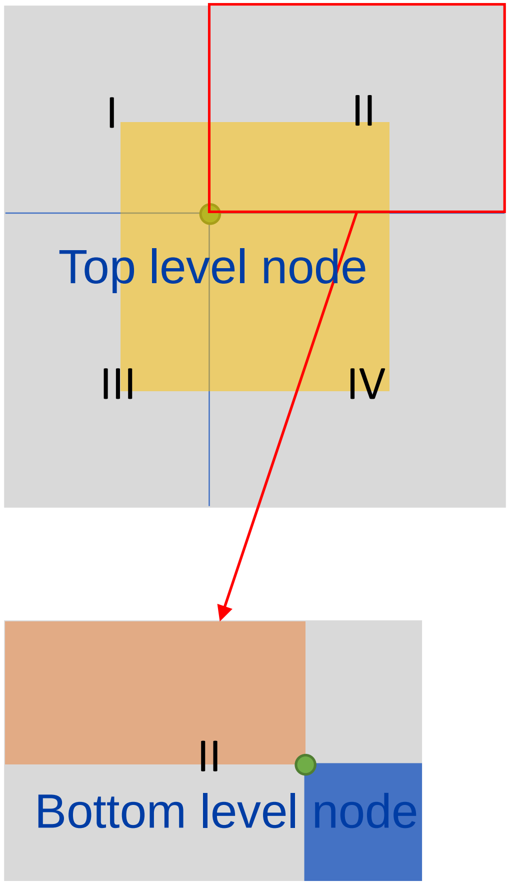
Nodes: 12288
Edges: 65807

Proposed algorithm to generate chiplet layout randomly

Data Generation

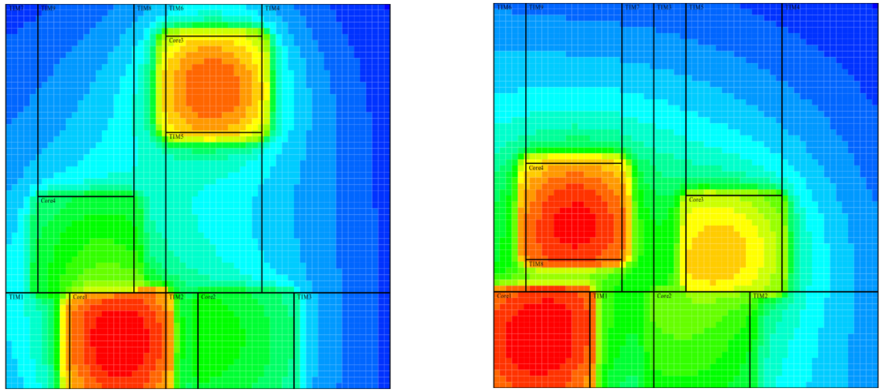
- Chiplet Layout





Top level node area: 6x6=36
Bottom level node area: (X-3)x(Y-3)
Power: [1,3,5,7,9]W

- Examples



- Dataset

Total samples: 400(Layout)x20(Power)=8000

Training set: 6800(85%)
Test set: 1200(15%)

Experimental Results

- GCN without PNA

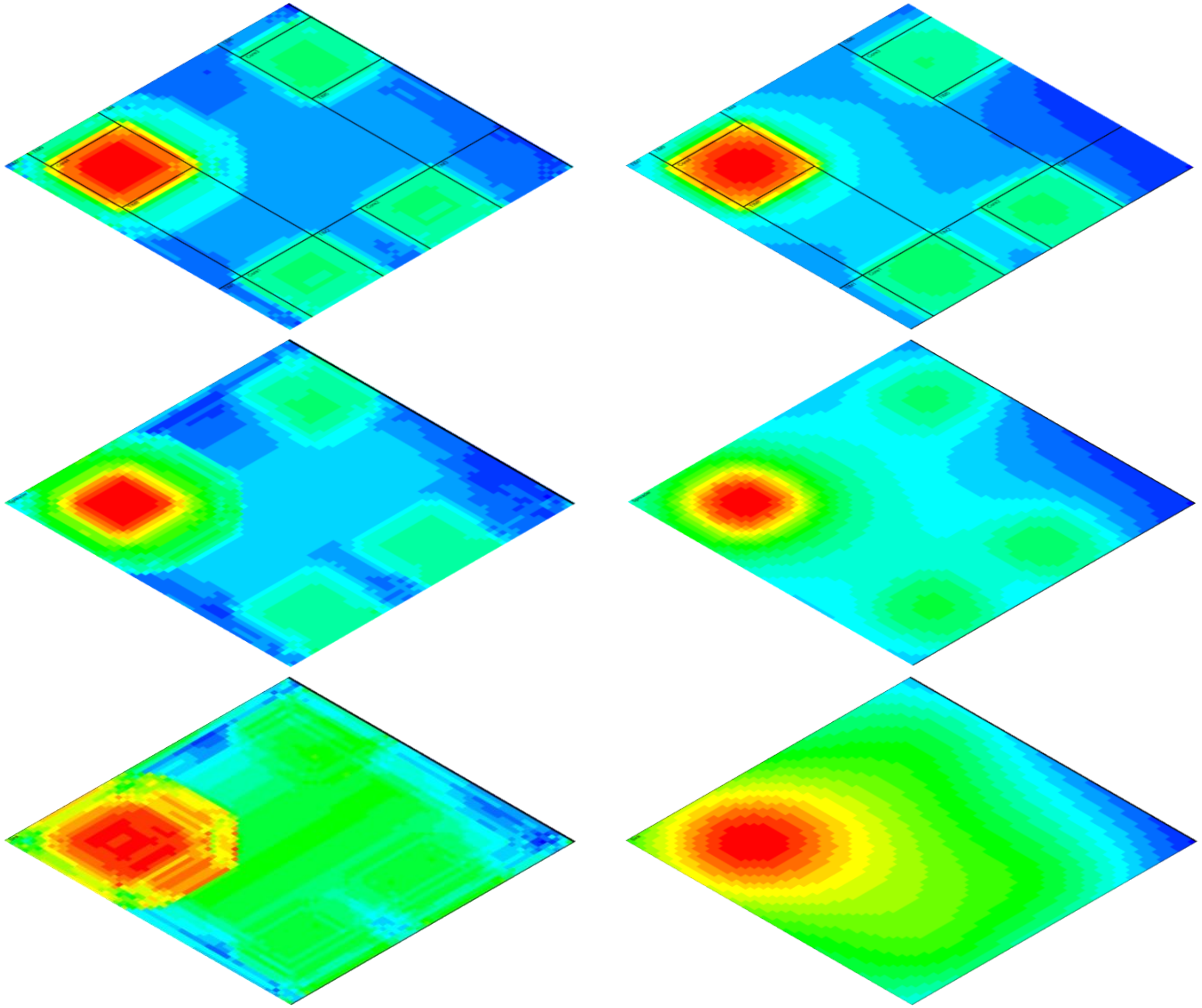
  - Graph Convolution Network.
  - Skip Connection
  - Graph Attention Network

- Accuracy

Relative Err = 0.91%(Test)

Mean RMSE= 0.35 K

Maximum Absolute Err = 2.54 K



GCN without PNA

HotSpot

Chiplets

Heat Spreader

Heat Sink

Experimental Results

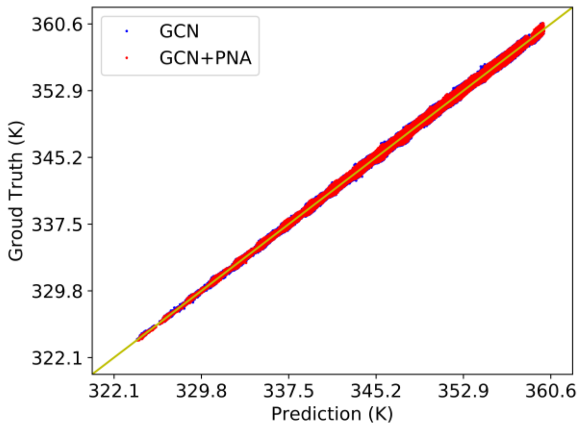
- GCN+PNA

Relative Err = 0.80%(Test) ⬇ 0.11%

Mean RMSE= 0.31 K ⬇ 0.04K

Maximum Absolute Err = 2.07 K ⬇ 0.47K

- Accuracy comparison of GCN and GCN+PNA



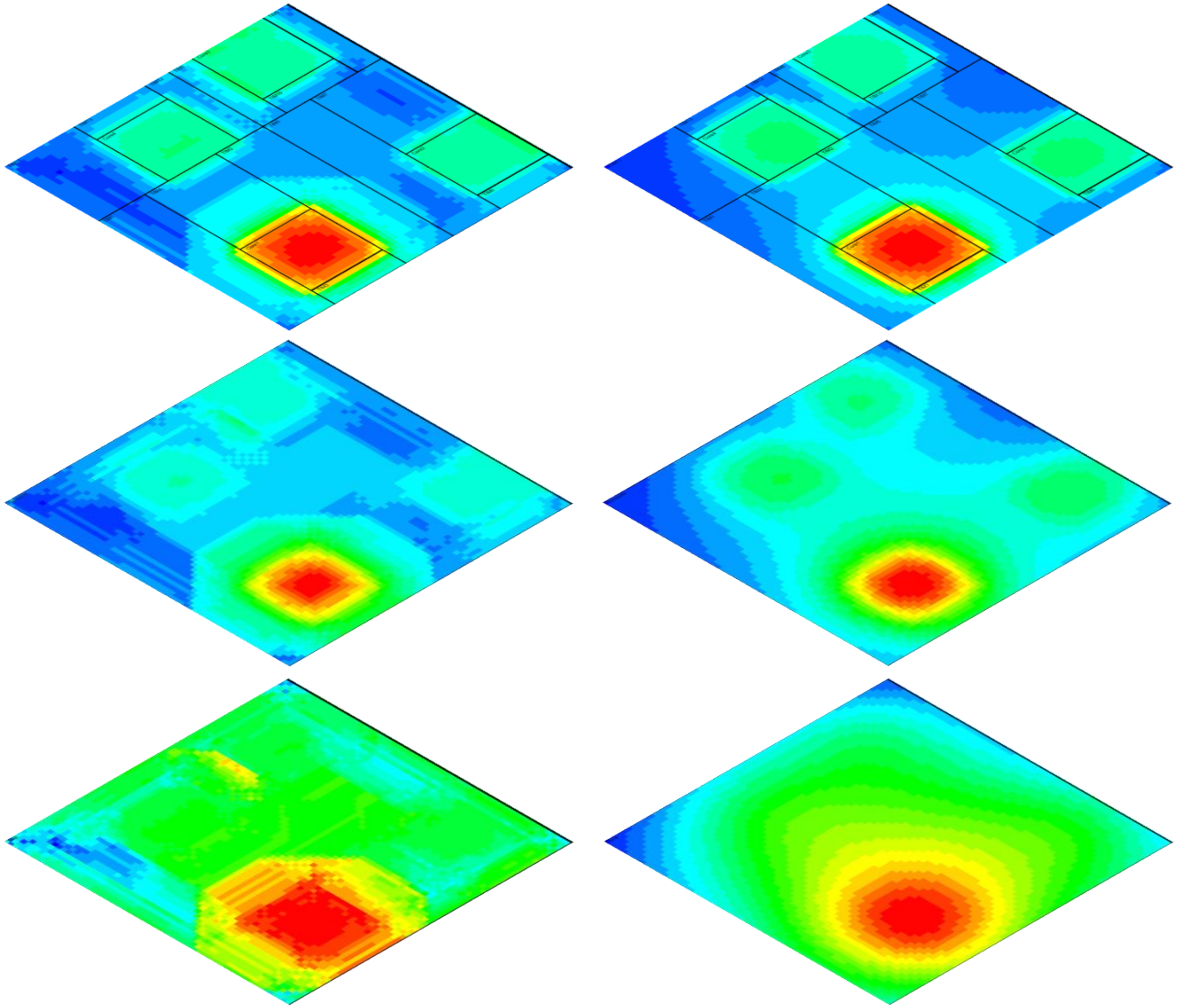
Temperature prediction vs ground truth

GCN+PNA HotSpot

Chiplets

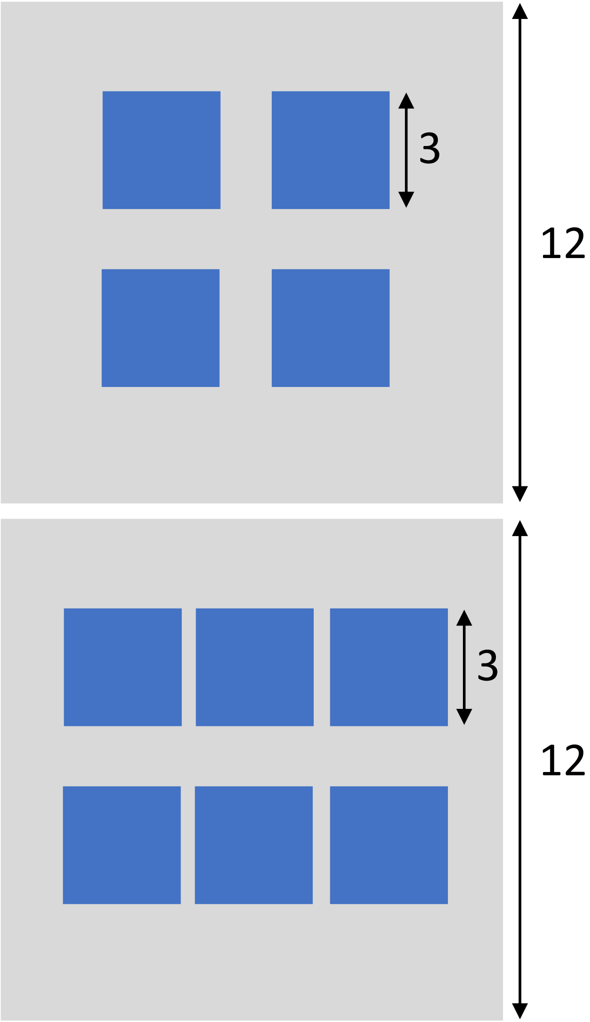
Heat Spreader

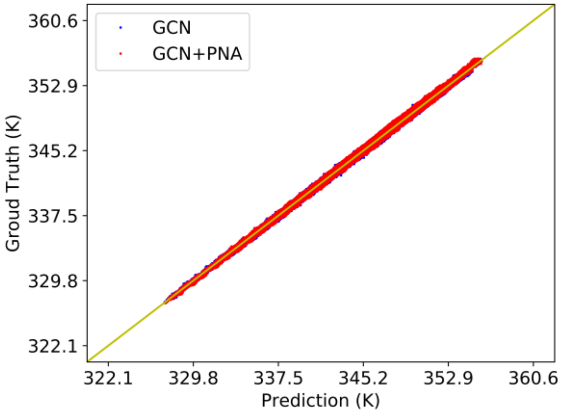
Heat Sink



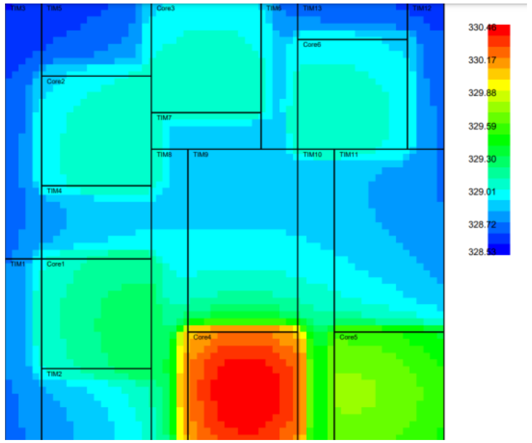
Generalization on unseen chips with same size

Increase the number of Chiplets from 4 to 6

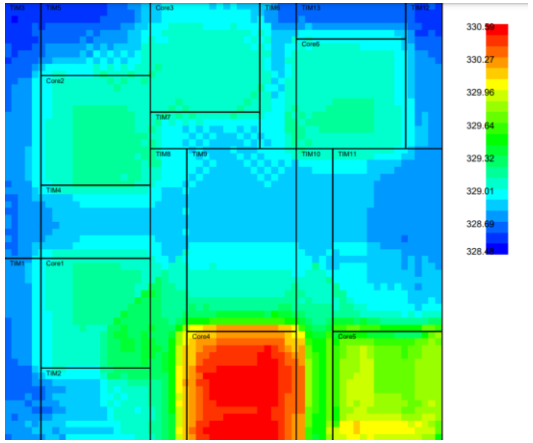




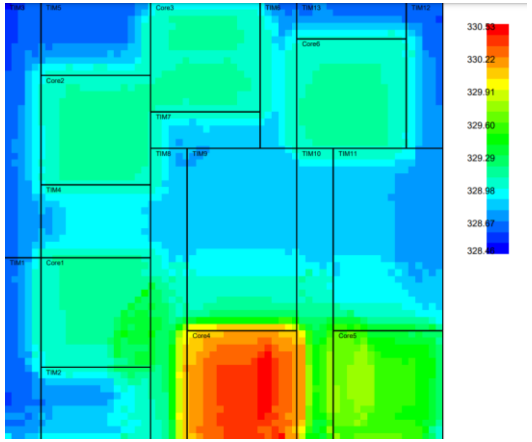
Temperature prediction vs ground truth



HotSpot



GCN+PNA



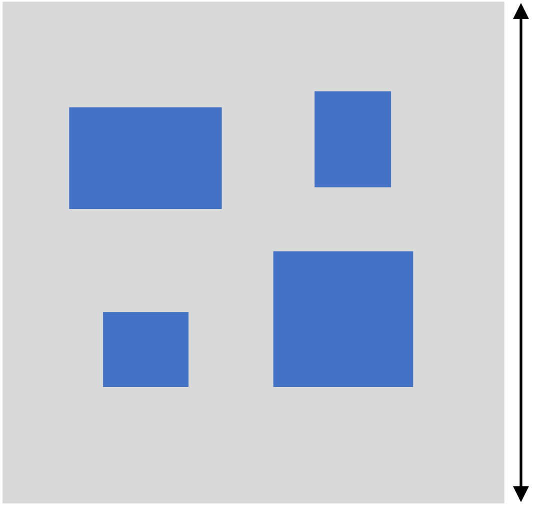
GCN

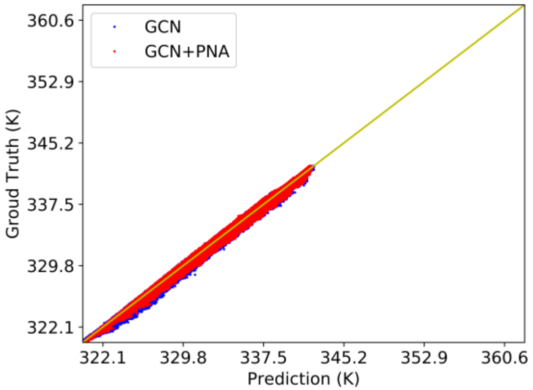
Table: Accuracy Comparison on 1200 unseen design with six chiplets

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%

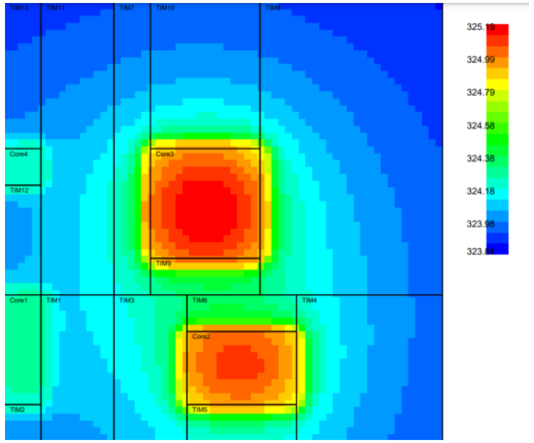
Generalization on unseen chips with same size

Change the size of 4 chiplets

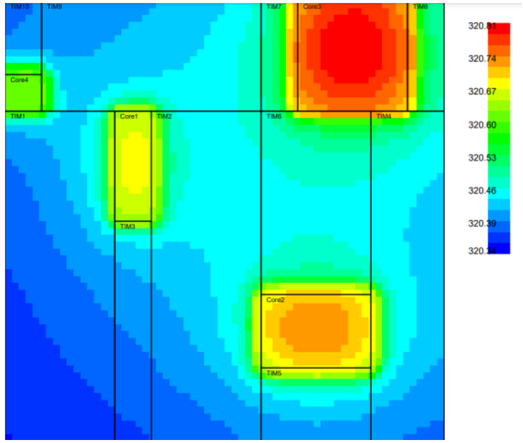




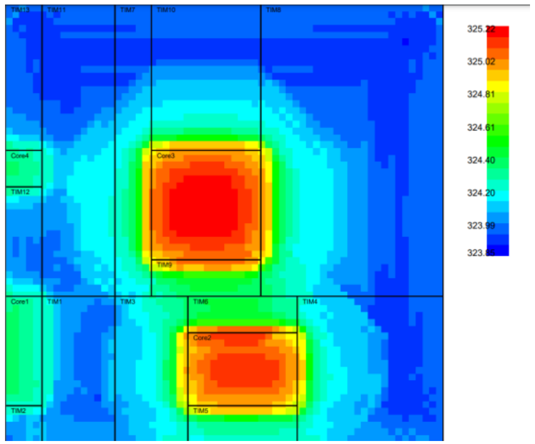
Temperature prediction vs ground truth



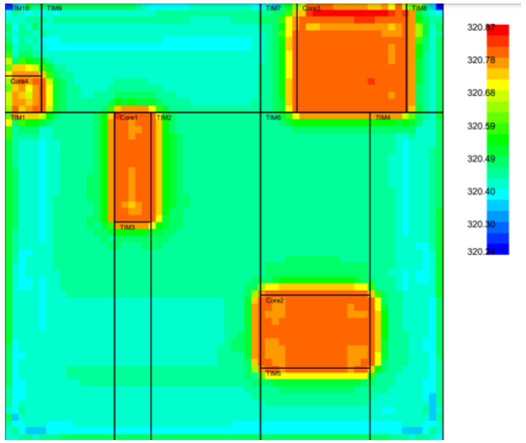
HotSpot



HotSpot



GCN+PNA



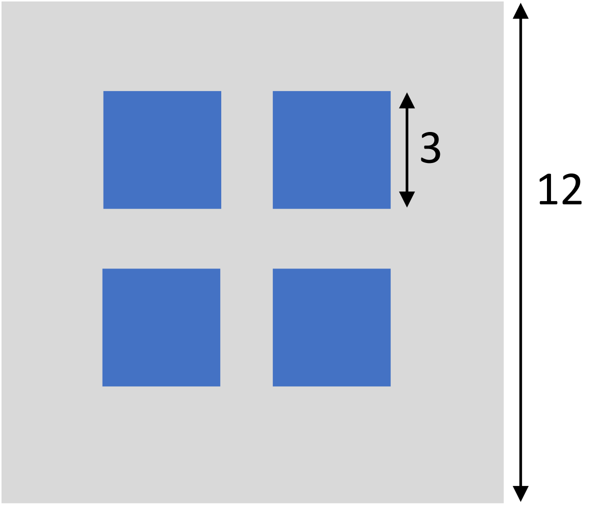
GCN

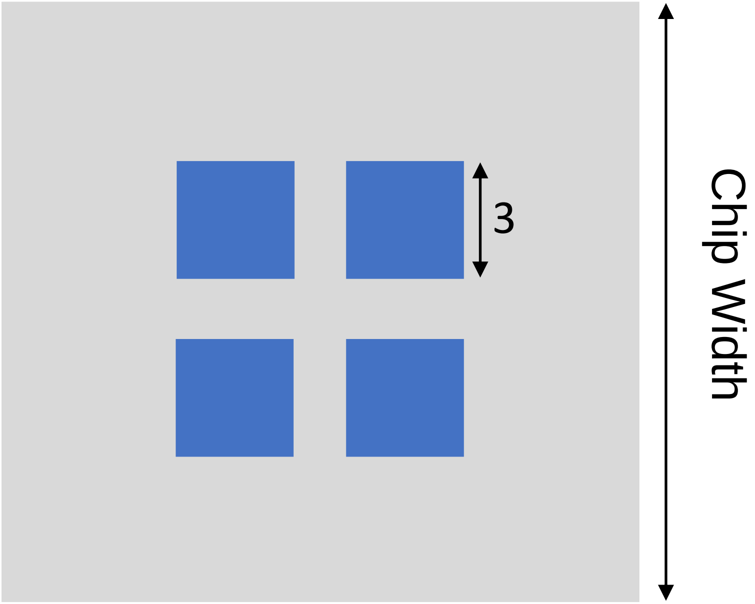
Table: Accuracy Comparison on 1200 unseen design with different sizes of chiplets

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%

Generalization on unseen chips with different size

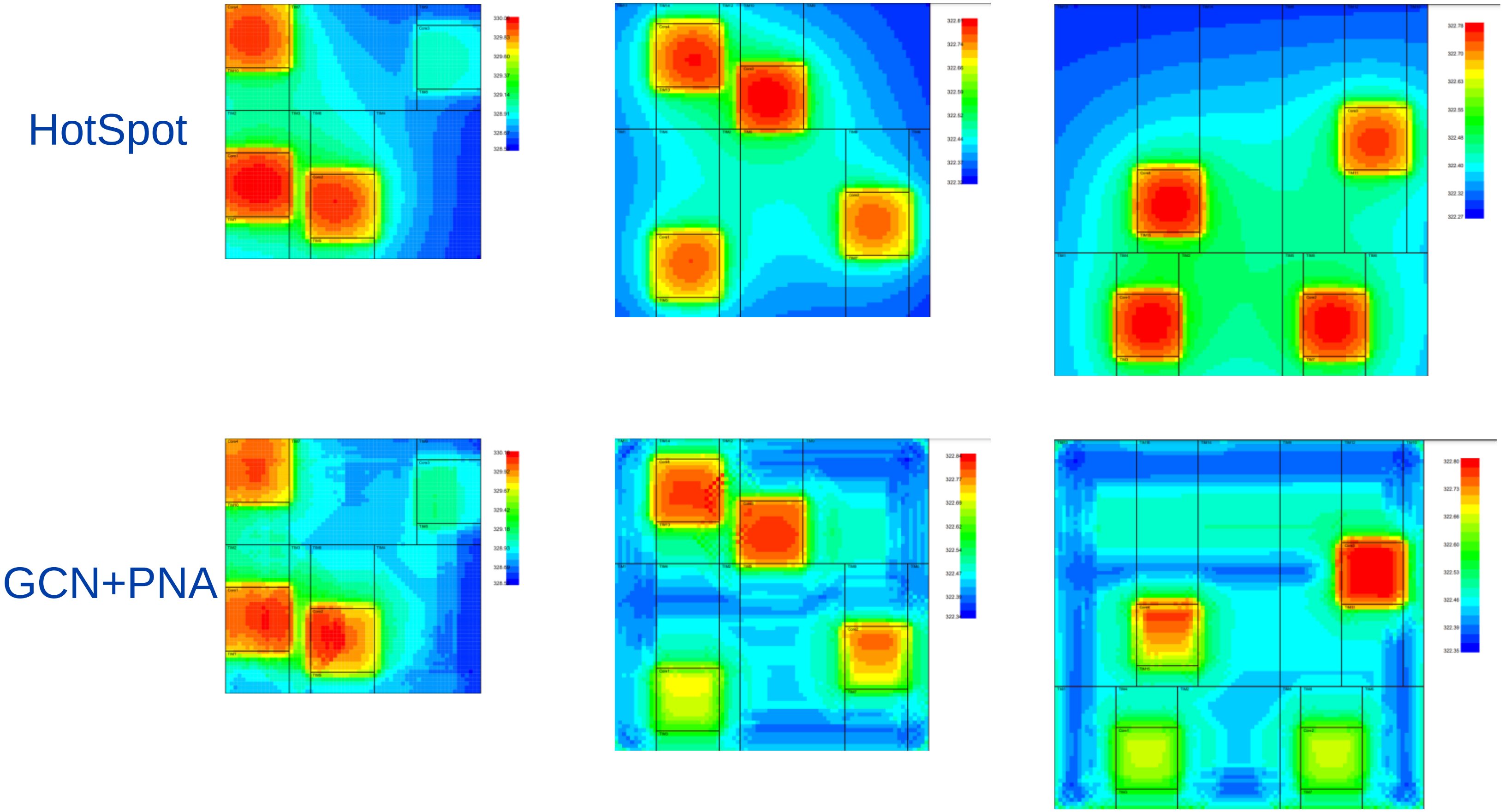
Increase the size of chip





Thermal map

Chip Width (mm)	12	15	18
Number of Nodes	12288	19200	27648



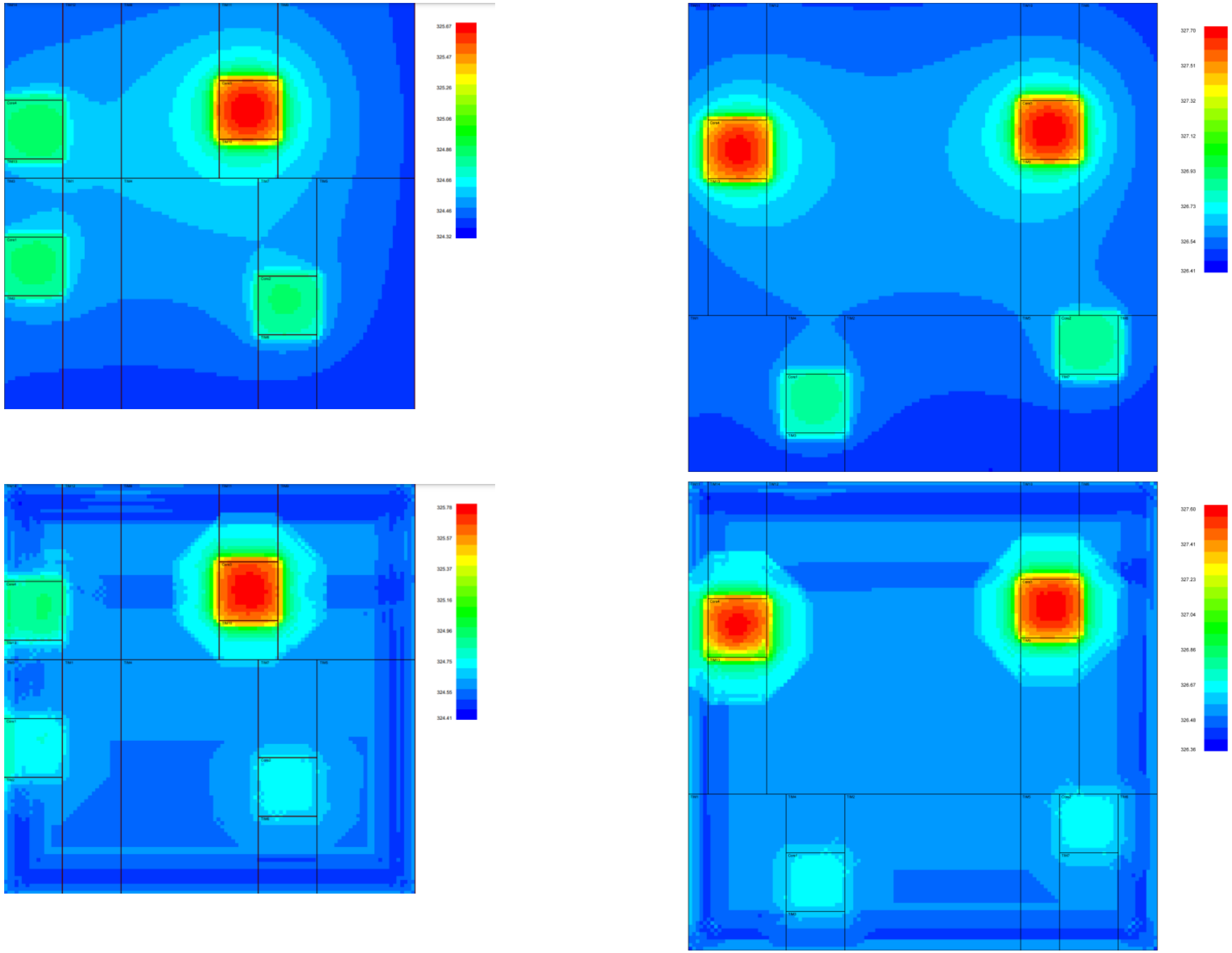
Generalization on unseen chips with different size

Chip Width (mm) 21 24

Number of Nodes 37632 49152

HotSpot

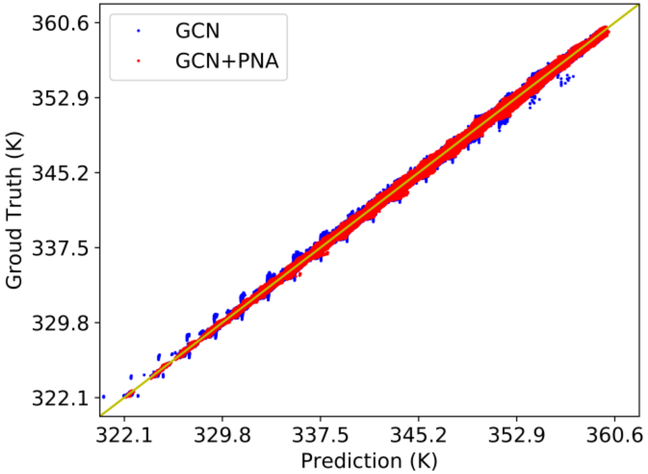
GCN+PNA



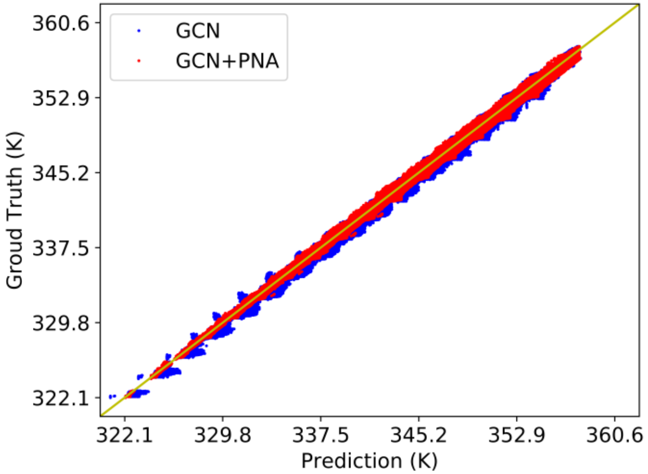
The large unseen chip is the double size of the original one.

Generalization on unseen chips with different size

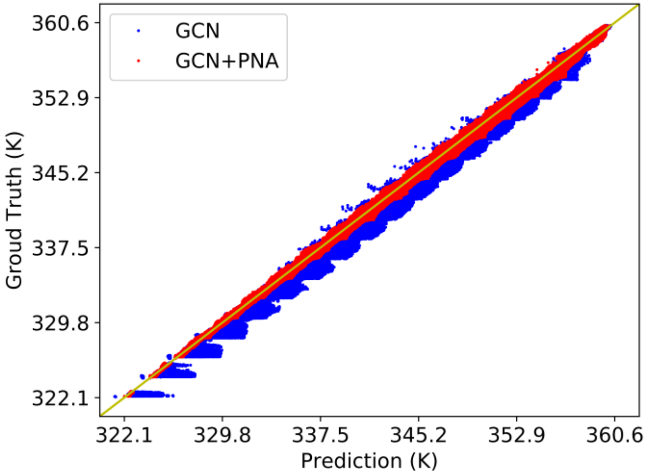
Predictions vs Truths



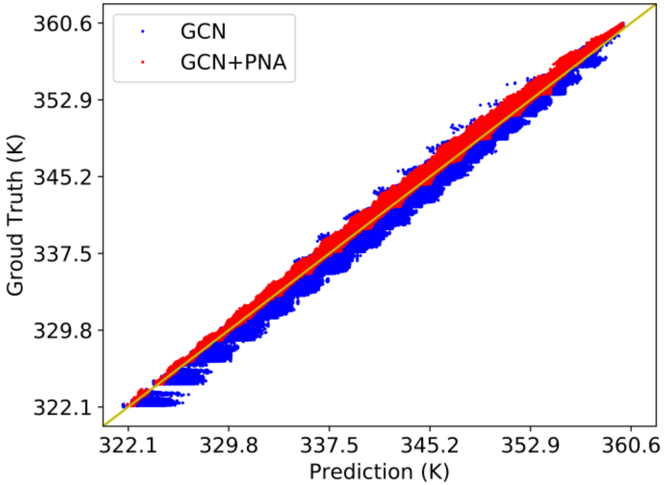
Chip Width of 15 mm



Chip Width of 18 mm



Chip Width of 21 mm

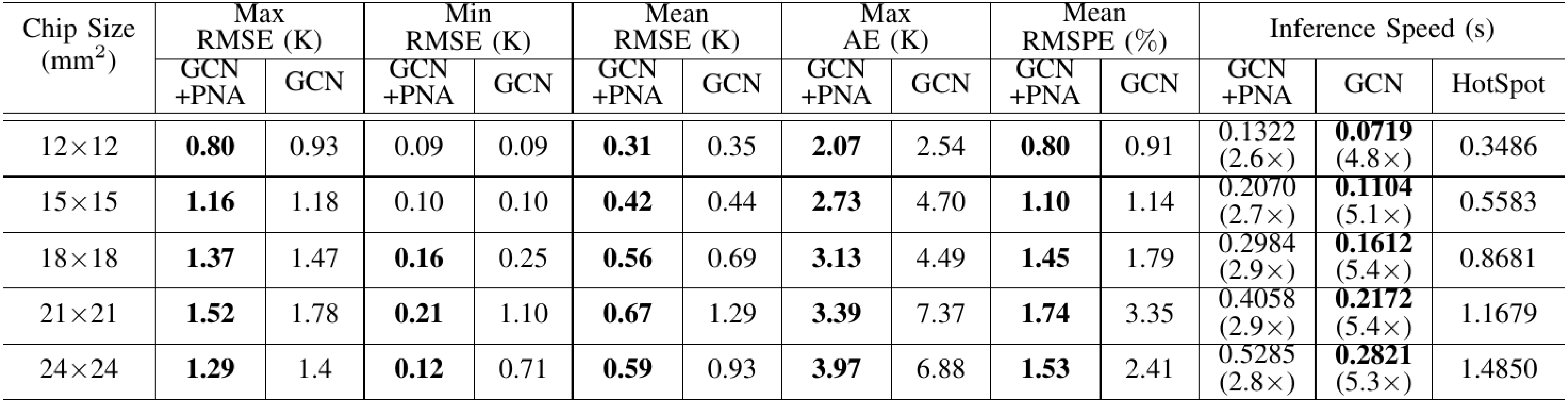


Chip Width of 24 mm

PNA can improve the accuracy of GCN on large unseen chip

Generalization on unseen chips with different size

Table: Accuracy Comparison on unseen designs with different size of chip



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

Speed Comparison (in a Linux server with Xeon E5 2.2 GHz CPU and NVIDIA Titan RTX GPU with 24GB memory)

- GCN model can achieve 4.8× speedup over HotSpot based on SuperLU.
- GCN+PNA model can achieve 2.6x speedup over HotSpot based on SuperLU.

Summary and conclusion

- In this work,
  - We 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 opensource HotSpot based on the CTM
  - 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.
  - In addition, the trained GCN with PNA can also be capable 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
- 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.

ASP-DAC 2022

THANKS