TTNNM: Thermal- and Traffic-Aware Neural Network Mapping on 3D-NoC-based Accelerator

Xinyi Li
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Wenjie Fan
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Heng Zhang
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Jinlun Ji
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Tong Cheng
School of Electronic Science and Engineering, Nanjing University
Nanjing, China

Shiping Li
Jiangsu Huachuang Microsystem Co., Ltd
Nanjing, China

Li Li*
School of Electronic Science and Engineering, Nanjing University
Nanjing, China
lili@nju.edu.cn

Yuxiang Fu*
School of Integrated Circuits, Nanjing University
Suzhou, China
yuxiangfu@nju.edu.cn

ABSTRACT

3D Network on Chips (3D-NoCs) have ample on-chip wiring resources and high bandwidth, yet face numerous hotspots and higher temperature gradients due to increased integration and power density. This could lead to device failure, impacting system stability. Our paper introduces a thermal- and traffic-aware mapping method for 3D-NoC-based neural network accelerators. Firstly, based on the average load of different neural network layer, we determine their mapping sequences and suitable dies. Secondly, to minimize delay and alleviate hotspot temperatures, we allocate groups to appropriate nodes. Compared with previous works, TTNNM reduces the average temperature by 3.0°C, 2.2°C, 2.4°C, temperature variance by 58.4%, 64.8%, 73.0%, maximum temperature by 9.3°C, 7.9°C, 12.0°C, and packet latency by 31.7%, 17.2%, 25.1%.

CCS CONCEPTS

• **Hardware → 3D integrated circuits**; **Application specific integrated circuits**.

*Corresponding authors.

This work was supported in part by the National Nature Science Foundation of China under Grant No. 62104098 and in part by the Natural Science Foundation of Jiangsu Province for Youth under Grant No. BK20210178 and in part by the Joint Funds of the National Nature Science Foundation of China under Grant No. U21B2032 and in part by the National Key Research and Development Program of China (No. 2021YFB3600104) and in part by the National Key Research and Development Program of China (No. 2023YFB2806802). The authors would like to express their gratitude to the Interdisciplinary Research Center for Future Intelligent Chips (Chip-X) and Yachen Foundation.

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658703

KEYWORDS

3D-NoC, Thermal-aware, Traffic-aware, Mapping, NoC-based Neural Network Accelerator

ACM Reference Format:
Xinyi Li, Wenjie Fan, Heng Zhang, Jinlun Ji, Tong Cheng, Shiping Li, Li Li, and Yuxiang Fu. 2024. TTNNM: Thermal- and Traffic-Aware Neural Network Mapping on 3D-NoC-based Accelerator. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649476.3658703

1 INTRODUCTION

In recent years, neural networks (NNs) have found widespread application in fields like facial recognition and semantic analysis [12]. The NNs have grown deeper with more layers and parameters, driving the development of specialized hardware circuits for acceleration. NoC is often chosen for its high bandwidth and scalability [2, 5, 14, 17]. However, compared with traditional 2D-NoC systems, 3D-NoC systems suffer from more hotspots and higher temperature gradients due to increased integration, power density, and longer heat dissipation paths [13]. Uneven thermal conductivity in 3D-NoC heterogeneous layers [21] accelerates heat build-up away from heat sinks, exacerbating thermal issues that lead to device failure, reduced lifespan, and compromised system stability and reliability. Heavy computational workload on each Processing Element (PE) and massive data communication between PEs exacerbate thermal problems in 3D-NoC-based NN accelerator. Hence, there is an urgent need for a rational temperature control method.

Data transmission in multi-core systems occupies a large amount of power consumption. To mitigate chip thermal issues, the data transmission of hotspots can be restricted, such as scheduling the tasks of hotspots [3]. Additionally, voltage and frequency scaling techniques [16] are another methods to manage chip temperature, but may inadvertently increase network delay.

Apart from the temperature control methods mentioned above, mapping is another way to achieve a more uniform heat distribution in 3D-NoC. This article endeavors to propose a mapping

approach called Thermal- and Traffic-aware Neural Network Mapping (TTNNM) that balances the performance and temperature distribution in a 3D-NoC-based NN accelerator, which can be divided into three aspects. Firstly, we divide each NN layer's computing tasks into groups averagely. Secondly, we determine the mapping order and suited dies for NN layers. The NN layer with maximum computation, memory accessing, and communication load volume will be mapped on the dies closest to the heat sink first. Then the adjacent layer with higher load will be mapped to the available dies close to the heat sink. This process will iteratively continue until all NN layers have been mapped. Thirdly, to further minimize average packet latency and alleviate the temperature of hotspots, we allocate groups of each NN layer to suited nodes within the chosen dies. Our TTNNM is suitable for offline inference and similar application scenarios. Therefore, we only compare the optimization effects of various methods in terms of temperature and latency. The main contributions of this work are listed below:

1. TTNNM gives full consideration to the heterogeneous thermal transfer between 3D-NoC layers, and takes into account the distinct computational, memory accessing, and traffic loads of different NN layers. The NN layer with higher computation, memory accessing, and communication load volume tends to be mapped on the dies closer to the heat sink.

2. TTNNM further optimizes the traffic and temperature distribution by evaluating the traffic-load distribution, as well as considering the thermal dissipation and conduction effects of the tiles. It compares communication hop counts and high-load link numbers to limit packet transmission distance and prevent overcentralized mapping nodes and overlapping traffic paths.

3. Compared with random, NN-aware, cost model based mapping, TTNNM achieves a balance between latency and heat distribution, reducing the average temperature by 3.0°C, 2.2°C, 2.4°C, decreasing the temperature variance by 58.4%, 64.8%, 73.0%, reducing the maximum temperature by 9.3°C, 7.9°C, 12.0°C, and decreasing the packet latency by 31.7%, 17.2%, 25.1%.

The rest of this article are organized as follows. Section II introduces the related works about NN mapping. Section III presents our proposed TTNNM in detail. Experimental results are shown in Section IV. Finally, Section V concludes our work.

2 RELATED WORKS

With NoC gaining more and more popularity, researchers have increasingly devoted their attention to the mapping strategies. Mapping not only impacts the transmission latency between PEs but also affects the load distribution among dies in a 3D-NoC.

NN-aware represents a mapping algorithm built upon the Exhaustive Attack (EA) methodology [14]. It results in a mapping solution that significantly minimizes the routing distance and effectively distributes data traffic across the NoC, thereby achieving substantial latency reduction. GAMMA is a domain-specific genetic algorithm-based method designed specially for the HW-mapping problem proposed by Kao et al. [10]. GAMMA determines an optimized mapping with high sample efficiency, but the initial mappings can affect the potential search region of mappings, which may lead to sub-optimal solutions. The ConfuciuX [9], a reinforcement learning (RL) technique has been introduced to guide the NP-hard search process, but its application is more centered on optimizing hardware resource assignments as opposed to addressing hardware mapping challenges. Xue et al. propose an Autonomous Optimal Mapping Exploration (AOME) architecture to find optimal NN hardware mappings [20]. It is based on reinforcement learning framework and finds out better solutions. However, all the above strategies neglect to take temperature into account and can lead to local hotspots in 3D-NoC.

Titirsha et al. presented a novel thermal-aware mapping technique that leverages a hill-climbing algorithm to effectively minimize the average temperature of the chip [18]. But this technique is presented for 2D-NoC instead of the 3D-NoC. HotClus-ter [4] introduces a thermal-aware defect recovery approach for Through-Silicon Vias (TSVs) that clusters them according to their activity levels and maintains spare clusters to address defects. Despite its merits, this method is not specifically designed to handle the mapping of neurons within the chip. Li et al. proposed a mapping algorithm to optimize both communication and computational performance under a thermal constraint in 3D-NoC systems [11]. A thermal-aware defragmentation algorithm has also been put forth, aiming to minimize system fragmentation and reduce waiting times. But their application scenario focuses on runtime mapping for applications. In [19], a high-level thermal model is constructed by Wang et al. to capture the relationship between system thermal distribution and thread-to-core mapping. Based on this, a novel thermal-aware mapping algorithm that automatically allocates each thread of a forthcoming application to the suitable core is proposed. In their cost model, not only is temperature distribution taken into consideration, but also the hop count, which directly influences latency, is factored in. However, this approach is presented for traditional application thread mapping, not for the NN mapping and needs to be adapted. In [15], Maatar et al. introduces a novel premapping clustering technique tailored for NASH, a mixed-signal neuromorphic architecture built upon 3D-NoC. It effectively regulates the temperature of tiles within each layer, thereby reducing the overall peak temperature. But it is proposed for Spiking Neural Network (SNN) and primarily focuses on neuron clustering. In conclusion, our work comprehensively addresses thermal- and traffic-aware mapping for 3D-NoC based DNN accelerators.

3 TTNNM: THERMAL- AND TRAFFIC-AWARE NEURAL NETWORK MAPPING

In this section, we will introduce our mapping strategy in detail. It achieves a balance between performance and heat distribution.

Our mapping methodology can be fundamentally divided into three steps. Initially, we evenly distribute the computational tasks of each NN layer into groups, avoiding creating obvious hotspots artificially. Secondly, we rank the NN layers based on the average computation, memory, and communication load volume of each layer. This process entails assigning the layer with the highest load to the dies proximate to the heat sink. Subsequently, the next NN layer for mapping is selected from one of the adjacent layers of the set of NN layers which have been mapped (S_{al}), always opting for the adjacent layer with a higher workload. This process will iteratively continue until all layers of the NN have been mapped. After that, we will obtain the mapping order of NN layers and

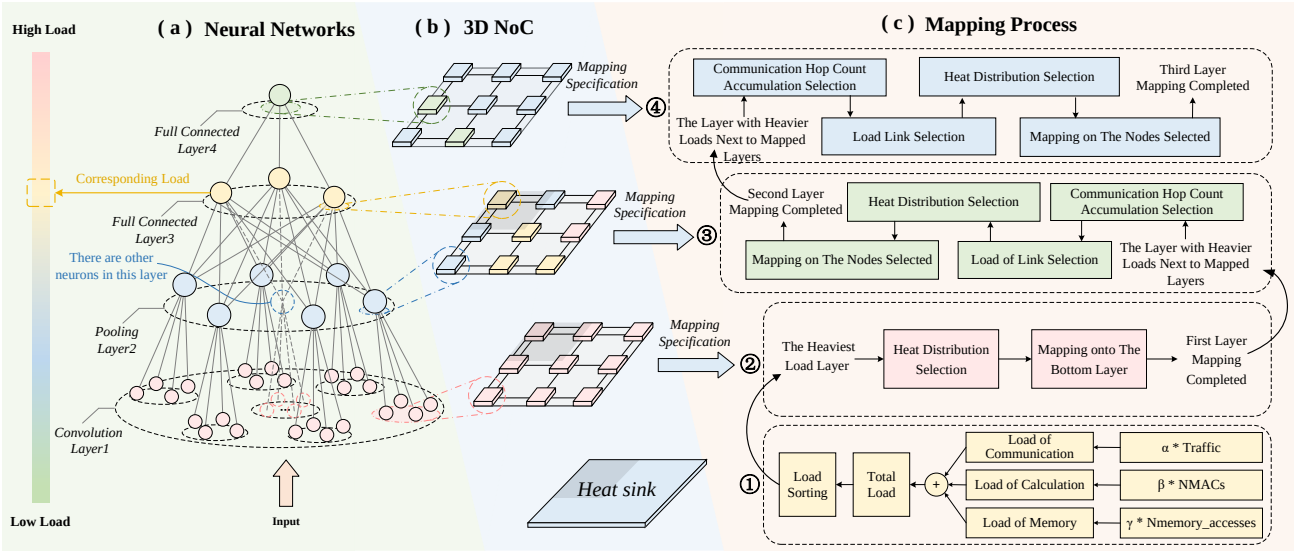


Figure 1: The total mapping process of TTNNM.

their suited dies. Finally, to further minimize delay and alleviate the temperature of hotspots, we allocate groups of each NN layer to appropriate nodes within the chosen dies.

In the above process, TTNNM not only gives full consideration to the heterogeneous thermal transfer between 3D-NoC layers, but also takes into account the distinct computational, memory, and traffic loads of different NN layers. We will delve into the second and third steps in detail in the following two subsections.

3.1 Determining the Mapping Order and Suited Dies for Different NN Layers

While we divide computational tasks of each NN layer into groups evenly, there inherently exists a disparity in computational complexity across different layers. For example, convolutional layers require a notably larger amount of computation compared to pooling layers. The memory access requirements and the number of data packets transmitted also vary greatly among different layers. As a result, the PE mapped with more loads on computing, memory accessing, and packets transmitting will consume more energy and cause faster heat accumulation. Based on the proportion of energy consumption on accelerators for MAC units, on-chip buffer memory accessing, and packet transmission[8], we have estimated approximate ranges for the values of α, β, and γ in Fig. 1 (c).

At the same time as heat accumulates, PEs also dissipate heat to the ambient air via the heat spreader and heat sink. The dies closer to the heat sink can dissipate heat to the air more quickly, which means that these dies can undertake more loads and energy consumption without creating ultra-high temperature spots.

Therefore, to distribute the heat more evenly, we decide the mapping order by comparing groups' average loads of each layer and select the one with more loads preferentially. We select dies nearer to the heat sink, allowing ample space for mapping each NN layer. This iterative process continues until all NN layers are mapped, as illustrated in Fig. 1 (a) and (b).

3.2 Allocating Groups of Each NN Layer to Appropriate Nodes of NoC

This section will introduce the process of allocating each group to the suitable tile, shown in Fig. 1 (c) and Algorithm 1. We use three optimization steps in turn to pick out mapping sequences with low temperature and transmission latency. Corresponding to the Communication Hop Count Accumulation Selection of ③ in Fig. 1 (c) and the 4th-6th lines in Algorithm 1, subsection 3.2.1 selects some mapping sequences with short packet transmission distances initially. Corresponding to the Load of Link Selection of ③ in Fig. 1 (c) and the 7th-17th lines in Algorithm 1, subsection 3.2.2 futher restricts traffic loads of links to optimize latency and heat distribution. Corresponding to the Heat Distribution Selection of ③ in Fig. 1 (c) and the 18th-23th lines in Algorithm 1, subsection 3.2.3 decides the final mapping sequence by evaluating thermal dissipation and conduction effects of tiles.

3.2.1 Selecting Mapping Sequences with Low Communication Hop Counts.

As communication latency is an important factor affecting the performance of multi-core processors, our approach needs to be optimized for it. In this paper, all packets are transmitted by using xyz dimensional-order routing algorithm. The number of NN layers and data packets transmitted by each NN layer are represented as N and $n_i (i \in [0, N-1])$. The hop number for one packet transmitted from the router of source node (S) to the router of destination node (D) can evaluate this packet's communication time roughly, as shown in formula (1). S_x, S_y, S_z represent the projection of node S in the x, y and z direction respectively. The same applies to D_x, D_y, D_z. Therefore, we evaluate the communication latency from NN layer i to $i+1$ by summing the number of their packets' transmitting hops, as shown in formula (2), where h_j is obtained in formula (1), and select some mapping sequences with the communication hop count smaller than the hop threshold (H_{th}). These sequences represent the mapping relationship between groups and nodes, which will

Algorithm 1 The process of allocating groups of NN layers to nodes of dies

Input: Set of unmapped NN layers (S_{un}), S_{al}
Output: Final Mapping Sequence (FMS)

while $S_{un} \neq \emptyset$ **do**
  Initialization: $S_{map1}, S_{map2}, S_{map3}, S_{map4} \leftarrow \emptyset$
  $S_{map1} = \{random\_map1, random\_map2, \cdots\}$
  for i in len(S_{map1}) **do**
    push $S_{map1}[i]$ with $H < H_{th}$ into S_{map2}
  end for
  for i in len(S_{map2}) **do**
    push $S_{map2}[i]$ with $l_{max} < l_{th}$ into S_{map3}
  end for
  for i in len(S_{map3}) **do**
    for j in 2L **do**
      if $l_j > n_{th}$ **then**
        $S_{map3}[i].N_l \leftarrow +1$
      end if
    end for
    push $S_{map3}[i]$ with $N_l < N_{th}$ into S_{map4}
  end for
  for i in len(S_{map4}) **do**
    for j in len(S_{al}) **do**
      $S_{map4}[i].TV \leftarrow \sum S_{al}[j].TV$
    end for
    push $S_{map4}[i]$ with the smallest TV into FMS
  end for
  update S_{al}, S_{un}
end while

be further screened by subsequent steps.

$$h = |S_x - D_x| + |S_y - D_y| + |S_z - D_z|. \tag{1}$$

$$H_i = \sum_{j=0}^{n_i} h_j. \tag{2}$$

3.2.2 Limiting the Maximum-load of Links and the Number of High-load Links.
Mapping sequences with short packet transmission distance will result in overcentralized mapping nodes and overlapping traffic paths, such as some links painted in red in Fig. 2 (c), which may lead to over-high link congestion finally, because latency will sharply escalate once the injection rate surpasses the saturation point, as shown in Fig. 2 (b). Besides, uneven traffic-load distribution will lead to over-high temperature of the chip locally, causing serious damage to the lifespan of the device and stability of the system. To alleviate congestion in NoC links, we will further select some mapping sequences according to the maximum-load of links (l_{max}) and the number of high-load links (N_l), which are used by currently mapped nodes, as lower loads of links can represent lower congestion and delay roughly. The number of all physical links in the dies are L. As all physical links allow packets transmissing bidirectionally, without mutual interference, the number of directional links (DLs) is $2{\times}L$. The load of each link is named in $[l_0, l_{2L-1}]$. The initial congestion numbers of all DLs are zero. If there are some packets (n'_i) being transmitted in one DL (l_i), its congestion number will

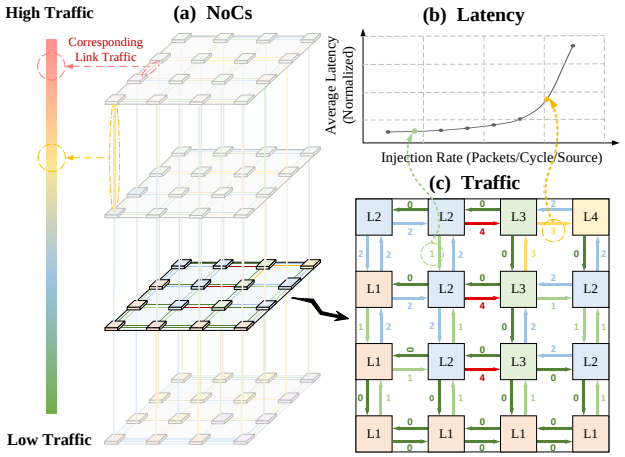


Figure 2: The traffic load of different links and their impact on the latency.

increase correspondingly, shown in formula (3). l_{max} and N_l are shown in formula (4), (5). n_{th} is links' threshold of traffic loads. N_l is the number of links with traffic loads more than n_{th}. As a result, we select some mapping sequences with their $l_{max} < l_{th}$ (another threshold of links' traffic loads) and $N_l < N_{th}$ (the threshold of the high-load links' quantities).

$$l_0, \cdots, l_{2L-1} = n'_0, \cdots, n'_{2L-1}. \tag{3}$$

$$l_{max} = MAX(l_0, \cdots, l_{2L-1}). \tag{4}$$

$$N_l = \begin{cases} N_l + 1 & l_i > n_{th}, \\ N_l & l_i \leqslant n_{th}. \end{cases} \tag{5}$$

3.2.3 Temperature Evaluation and Prediction for NoC Tiles.
As the temperature of the NoC tile is affected by its energy consumption and heat dissipation, our method conducts temperature evaluation and prediction of each tile roughly according to these two indicators:

- As the heat generation exhibits a positive correlation with the energy consumption, we sum each tile's energy consumption (E_i, $i \in [0, N-1]$) created by completing some tasks, including computing, memory accessing, and packets transmitting, to evaluate its heat generation.
- As each tile transfers heat with air and other tiles, its temperature is affected by them. Both of their thermal influence decay exponentially with the Euclidean distance [6].

As a result, our method evaluates each tile's temperature value (TV_i, $i \in [0, N-1]$) as formula (6). Finally, our method will select the mapping sequence with the minimum sum of already mapped nodes' temperature values, as shown in formula (7). η is a parameter used to adjust the weight of heat dissipation with the ambient air. d_j is the distance between the node j and the heat sink. $d_{i,j}$ is the distance between the node i and the node j.

$$TV_i = \sum_{j=0}^{N-1} [E_j \times (1 - \eta \times e^{-d_j})] \times e^{-d_{i,j}}. \tag{6}$$

$$TV = \sum_{i \in S_{al}} TV_i. \quad (7)$$

4 EVALUATION

4.1 Experimental Setup

We run the experiment based on a cycle-accurate traffic-thermal co-simulation platform called AccessNoxim [7]. It integrates the NoC simulator Noxim and the architecture-level thermal model Hotspot. The basic parameters of the NoC are listed in Table 1.

To validate the effectiveness of our proposed method, we totally mapped 4 different neural networks' trace to the AccessNoxim platform. The trace of these networks were obtained from a high-level abstract model in which the accelerator employs a fully unfolded pipeline dataflow like the Tianjic chip [5] and the output of each PE is parallelized across the channel dimension.

Table 1: 3D-NoC-based NN Accelerator Configurations.

Parameters	Settings	Parameters	Settings
Mesh Size	$8 \times 8 \times 4$	MACs / PE	256
Buffer Depth	4 Flits	Packet Length	8 Flits
Routing	XYZ	Simulation Time	3×10^6 cycles
Initial Temperature	65°C	Warm-up Time	10000 cycles

4.2 Experiment Results

We compare our proposed TTNNM with other three mapping strategies, namely random, NN-aware [14] and cost model [19]. The cost model originates from [19], and we map the NN layers in their original sequence just like the conventional applications coming in sequence in this strategy. The following parts will present our experiment results from different aspects.

4.2.1 *Temperature and Latency Overview.*

Table 2 shows the average temperature, temperature variance, maximum temperature, and average packet latency of different mapping strategies for AlexNet, VGG11, VGG13 and VGG16 in a $8 \times 8 \times 4$ 3D-NoC. According to the experimental results in [1], the increase in average packet latency leads to an increase in communication delay, thereby exacerbating the total latency.

- AlexNet: Compared with random, NN-aware, cost model based mapping, TTNNM reduces the average temperature by 2.4°C, 1.4°C, 1.0°C, the temperature variance by 56.9%, 67.0%, 71.6%, the maximum temperature by 6.6°C, 8.3°C, 14.8°C, and the packet latency by 33.3%, 8.7%, 27.7%.
- VGG11: Compared with random, NN-aware, cost model based mapping, TTNNM reduces the average temperature by 3.0°C, 2.6°C, 2.6°C, the temperature variance by 52.1%, 69.4%, 77.6%, the maximum temperature by 11°C, 10.1°C, 14.8°C, and the packet latency by 30.7%, 18.4%, 28.3%.
- VGG13: Compared with random, NN-aware, cost model based mapping, TTNNM reduces the average temperature by 3.9°C, 2.9°C, 2.9°C, the temperature variance by 72.4%, 71.7%, 78.0%, the maximum temperature by 15.1°C, 9.7°C, 11.8°C, and the packet latency by 34.0%, 20.2%, 26.7%.
- VGG16: Compared with random, NN-aware, cost model based mapping, TTNNM reduces the average temperature by 2.7°C, 2°C, 3°C, the temperature variance by 52.4%, 51.1%, 64.6%, the maximum temperature by 4.3°C, 3.3°C, 6.5°C, and the packet latency by 28.6%, 21.4%, 17.8%.

Table 2: Average temperature, temperature variance, maximum temperature, and latency of different mapping strategies for AlexNet, VGG11, VGG13 and VGG16 in a 3D-NoC.

Net	Strategy	T_{avg} (°C)	T_{var} (°C^2)	T_{max} (°C)	Latency (cycle)
AlexNet	Random	70.7	13.7	83.8	20.4
	NN-aware [14]	69.7	17.9	85.5	14.9
	Cost Model [19]	69.3	20.8	92.0	18.8
	TTNNM	68.3	5.9	77.2	13.6
VGG11	Random	73.7	14.6	90.3	17.9
	NN-aware [14]	73.3	22.9	89.4	15.2
	Cost Model [19]	73.3	31.2	94.1	17.3
	TTNNM	70.7	7.0	79.3	12.4
VGG13	Random	77.8	22.8	96.5	19.1
	NN-aware [14]	76.8	22.3	91.1	15.8
	Cost Model [19]	76.8	28.7	93.2	17.2
	TTNNM	73.9	6.3	81.4	12.6
VGG16	Random	79.3	14.5	89.6	17.5
	NN-aware [14]	78.6	14.1	88.6	15.9
	Cost Model [19]	79.6	19.5	91.8	15.2
	TTNNM	76.6	6.9	85.3	12.5

In summary, compared with random, NN-aware, cost model based mapping, our proposed TTNNM reduces the average temperature by 3.0°C, 2.2°C, 2.4°C, temperature variance by 58.4%, 64.8%, 73.0%, maximum temperature by 9.3°C, 7.9°C, 12.0°C, and packet latency by 31.7%, 17.2%, 25.1%.

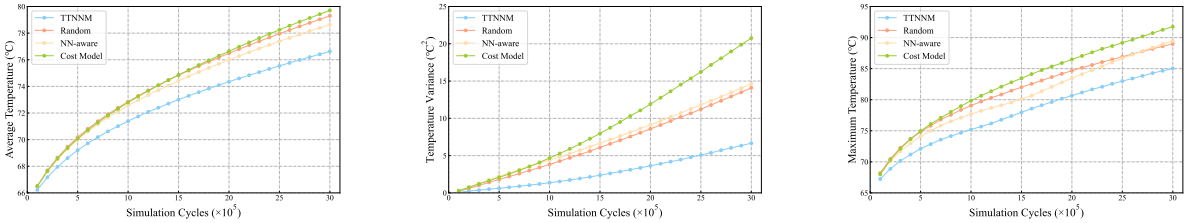
Compared to the random mapping strategy, the NN-aware mapping strategy has been optimized for latency but exhibits subpar performance in terms of temperature, a deficiency that can be rectified through TTNNM. In some networks, the performance of the cost model-based mapping strategy deteriorates in both latency and temperature due to its original mapping sequence of NN layers. This issue is addressed in TTNNM, where the distinct computational, memory access, and traffic loads of various NN layers are taken into account for optimization.

4.2.2 *Temperature Changes Over Time.*

Fig. 3 presents the variation of average temperature, temperature variance and maximum temperature throughout the entire simulation time of VGG16. Our proposed TTNNM always achieves the lowest average temperature and temperature variance throughout the whole simulation, and achieves the lowest maximum temperature when simulation time increases.

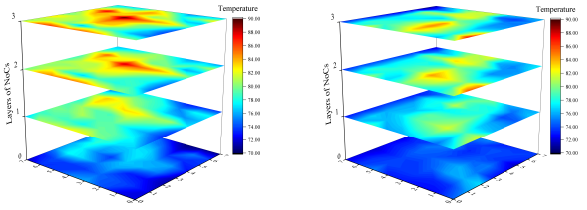
4.2.3 *Temperature Distribution across NoC.*

Fig. 4 shows us the temperature distribution across various layers of the NoC at the end of simulation for VGG16 using NN-aware



(a) Average Temperature Change (b) Temperature Variance Change (c) Maximum Temperature Change

Figure 3: Temperature Change throughout the entire duration of simulation.



(a) NN-aware (b) TTNNM

Figure 4: The temperature distribution across various dies of the NoC at the end of simulation for VGG16 using NN-aware mapping and our TTNNM.

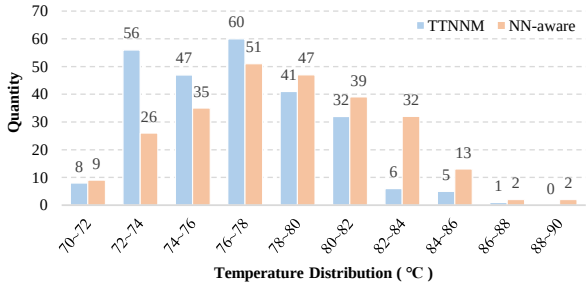


Figure 5: The number of tiles in different temperature ranges at the end of simulation for VGG16 using NN-aware mapping and our TTNNM.

mapping and our TTNNM. Our proposed TTNNM achieves a more balanced temperature distribution across different layers.

Additionally, we have counted the number of tiles in different temperature ranges. Fig. 5 shows the results at the end of simulation for VGG16 using NN-aware mapping and our TTNNM. Compared with the NN-aware mapping, our proposed TTNNM results in a smaller number of tiles within the high-temperature range.

5 CONCLUSION

Thermal issues are an inescapable topic in the context of 3D-NoC technology. The mapping strategy not only influences the communication latency among PEs but also plays a crucial role in shaping the temperature distribution across layers within a 3D-NoC. In this article, our proposed TTNNM gives full consideration to the heterogeneous thermal transfer between 3D-NoC layers, and take into account the distinct loads of different NN layers. Compared with random, NN-aware, cost model based mapping, our proposed TTNNM reduces the average temperature by 3.0°C, 2.2°C, 2.4°C, decreases the temperature variance by 58.4%, 64.8%, 73.0%, reduces the maximum temperature by 9.3°C, 7.9°C, 12.0°C, and decreases the packet latency by 31.7%, 17.2%, 25.1%. Additionally, we can utilize heuristic algorithms in our future work to narrow down the search space and shorten the search time for the optimal mapping.

REFERENCES

[1] Kun-Chih Jimmy Chen et al. 2020. A NoC-based simulator for design and evaluation of deep neural networks. *Microprocess. Microsyst.* 77 (2020), 103145.
[2] Yu-Hsin Chen et al. 2019. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 9, 2 (2019), 292–308.
[3] Ayse Kivilcim Coskun et al. 2008. Proactive temperature balancing for low cost thermal management in MPSoCs. In *Proc. ICCAD*. IEEE, 250–257.
[4] Khanh N Dang et al. 2022. HotCluster: A Thermal-Aware Defect Recovery Method for Through-Silicon-Vias Toward Reliable 3-D ICs Systems. *IEEE Trans. Comput. Aided Des. Integr Circuits Syst* 41, 4 (2022), 799–812.
[5] Lei Deng et al. 2020. Tianjic: A unified and scalable chip bridging spike-based and continuous neural computation. *IEEE J Solid State Circuits* 55, 8 (2020), 2228–2246.
[6] Wei Huang et al. 2006. HotSpot: a compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. Very Large Scale Integr VLSI Syst* 14, 5 (2006), 501–513.
[7] Kai-Yuan Jheng et al. 2010. Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip. In *Proc. VLSI-DAT*. IEEE, 135–138.
[8] Neethu K et al. 2022. Impact Analysis of Communication Overhead in NoC based DNN Hardware Accelerators. In *Proc. INDICON*. 1–6.
[9] Sheng-Chun Kao et al. 2020. ConfuciuX: Autonomous Hardware Resource Assignment for DNN Accelerators using Reinforcement Learning. In *Proc. MICRO*. 622–636.
[10] Sheng-Chun Kao et al. 2020. GAMMA: Automating the HW Mapping of DNN Models on Accelerators via Genetic Algorithm. In *Proc. ICCAD*. 1–9.
[11] Bing Li et al. 2019. On Runtime Communication and Thermal-Aware Application Mapping and Defragmentation in 3D NoC Systems. *IEEE Trans. Parallel Distrib Syst* 30, 12 (2019), 2775–2789.
[12] Zewen Li et al. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans. Neural Networks Learn. Sys.* (2021).
[13] Shu-Yen Lin et al. 2011. Traffic-and thermal-aware routing for throttled three-dimensional Network-on-Chip systems. In *Proc. VLSI-DAT*. IEEE, 1–4.
[14] Xiaoxiao Liu et al. 2018. Neu-NoC: A high-efficient interconnection network for accelerated neuromorphic systems. In *Proc. ASP-DAC*. IEEE, 141–146.
[15] Mohamed Maatar et al. 2023. Thermal-Aware Task-Mapping Method for 3D-NoC-Based Neuromorphic Systems. In *Proc. ICET*. IEEE, 1067–1076.
[16] M Momeni et al. 2020. Energy optimization in 3D networks-on-chip through dynamic voltage scaling technique. In *Proc. ICEE*. IEEE, 1–4.
[17] Yakun Sophia Shao et al. 2019. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *Proc. MICRO*. 14–27.
[18] Twisha Titirsha et al. 2020. Thermal-aware compilation of spiking neural networks to neuromorphic hardware. In *Proc. LCPC*. Springer, 134–150.
[19] Jian Wang et al. 2016. A High-Level Thermal Model-Based Task Mapping for CMPs in Dark-Silicon Era. *IEEE Trans. Electron Devices* 63, 9 (2016), 3406–3412.
[20] Yongqi Xue et al. 2022. AOME: Autonomous Optimal Mapping Exploration Using Reinforcement Learning for NoC-based Accelerators Running Neural Networks. In *Proc. ICCD*. IEEE; IEEE Comp Soc; Natl Sci Fdn, 364–367.
[21] Inchoon Yeo et al. 2008. Predictive dynamic thermal management for multicore systems. In *Proc. DAC*. 734–739.