Chiplet-Based Architectures for Scalable DNN Accelerators

DARPAN GAUR, Indian Institute of Technology, Hyderabad, India
LAKSHAY ARORA, Indian Institute of Technology, Hyderabad, India

This paper explores the emerging role of chiplet-based architectures as a scalable and energy-efficient alternative to monolithic designs for accelerating deep learning (DL) workloads. As DL models grow in complexity, the limitations of single-die systems, in terms of rising manufacturing cost, lower fabrication yield, and lack of design reuse, have motivated the adoption of multi-chiplet solutions. Chiplet integration enables the decomposition of large accelerators into smaller, reusable components interconnected through high-bandwidth links, facilitating improved modularity and performance scalability. The paper reviews recent advancements in hardware, software, and interconnect designs for chiplet-based architectures through works including INDM [7], Gemini [1], Florets for Chiplets [5], and TEFLON [3]. These works highlight the architectural, algorithmic, and thermal challenges of chiplet-based systems while demonstrating substantial gains in performance, energy efficiency, and scalability. The paper concludes by emphasizing the promise of chiplets in redefining next-generation DL accelerator design, while acknowledging ongoing challenges in mapping, communication, and system-level integration and outlining future research directions to address these issues.

Additional Key Words and Phrases: Chiplets, dataflow mapping, 2.5D, 3D Integration, NoI/NoC, PIM, DNN

ACM Reference Format:
Darpan Gaur and Lakshay Arora. 2025. Chiplet-Based Architectures for Scalable DNN Accelerators. 1, 1 (May 2025), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Deep neural networks (DNNs) have become foundational to modern intelligent systems, enabling breakthroughs in areas such as image classification, speech recognition, natural language processing, and autonomous control. As models grow increasingly deeper and more complex, driven by innovations like neural architecture search (NAS) and transformer-based architectures, their computational and memory demands have surged dramatically. With the slowing of Moore's Law and the end of Dennard scaling, traditional monolithic silicon-based scaling strategies are no longer sufficient to meet the escalating performance requirements of DNN workloads.

To address these limitations, the semiconductor industry is transitioning toward chiplet-based architectures using advanced packaging technologies such as 2.5D interposers and organic substrates. These architectures decompose large chips into multiple smaller chiplets, offering advantages like improved yield, modularity, cost-efficiency, and enhanced scalability. High-bandwidth, low-latency interconnects at the package level enable effective communication between compute and memory chiplets, supporting scalable system design. Industry efforts like NVIDIA's SIMBA [4], and research efforts such as NN-Baton [6] and SIAM [2], highlight the viability of chiplet-based accelerators for large-scale DNN inference.

At the same time, In-Memory Computing (IMC) has emerged as a promising approach to address the memory wall by bringing computation closer to memory. IMC architectures exploit crossbar arrays of RRAM or SRAM to perform matrix-vector operations directly within memory, achieving high parallelism and energy efficiency. However, existing IMC solutions often assume all weights

Authors' Contact Information: Darpan Gaur, Indian Institute of Technology, Hyderabad, Hyderabad, India, co21btech11004@iith.ac.in; Lakshay Arora, Indian Institute of Technology, Hyderabad, Hyderabad, India, cs24resch11006@iith.ac.in.

2025. ACM XXXX-XXXX/2025/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

are stored in a single monolithic chip, which becomes infeasible as model sizes continue to scale. Chiplet-based IMC systems offer a scalable alternative by distributing IMC tiles across multiple chiplets, allowing larger models to be supported at lower cost and improved yield. These systems can harness the compute-density advantages of IMC and the modularity of chiplet-based integration, making them ideal for next-generation DNN workloads.

Key challenges in chiplet-based DNN accelerators include:

- **Communication bottlenecks:** Standard chiplet interconnects often exhibit lower bandwidth and higher latency than on-chip networks. The INDM architecture, for instance, shows that performance can be severely degraded without careful co-design of mapping strategies and interconnect topology.
- **Mapping complexity:** Allocating DNN layers across multiple chiplets becomes a high-dimensional optimization problem. The Gemini paper illustrates the need for intelligent mapping strategies that jointly consider data locality, load balancing, and inter-chiplet communication overheads.
- **Interconnect design:** As inter-chiplet dataflow becomes more critical, innovations like Floret's dataflow-aware Network-on-Interposer (NoP) are required to balance performance, latency, and physical wiring constraints.
- **Thermal constraints:** High-density packaging—especially in 3D-stacked IMC systems—can lead to thermal hotspots. The TEFLON framework highlights the importance of thermally-aware NoC design and mapping strategies to maintain energy efficiency and reliability.

As DNN accelerators move toward scalable chiplet-based platforms, overcoming these challenges is essential for future progress. This paper investigates the core issues of communication, workload mapping, and thermal management in chiplet-based DNN inference systems and proposes architectural and algorithmic co-design strategies to unlock their full potential.

2 Background

The explosive growth in DNN model size and complexity has outpaced the capabilities of conventional monolithic hardware accelerators. Recent works have proposed chiplet-based architectures and in-memory computing paradigms that better align with DNN dataflow characteristics to address scaling challenges. This section reviews the three influential baseline architectures, SIMBA, NN-BATON, and SIAM, each exploring different dimensions of scalable DNN inference.

2.1 SIMBA

SIMBA is the first chiplet-based deep learning accelerator designed to enable large-scale DNN inference with high throughput and modular scalability. The architecture (illustrated in Fig. 1) consists of a 2D mesh of 36 chiplets, each with 16 processing elements (PEs), a Global PE, local buffers, and a RISC-V controller. These chiplets are connected through a hierarchical interconnect—network-on-chip (NoC) within chiplets and network-on-package (NoP) across chiplets—allowing SIMBA to scale from edge devices to data-center workloads. The system achieves up to 128 TOPS with a peak efficiency of 6.1 TOPS/W.

The system uses advanced tiling strategies to efficiently map DNN computations onto SIMBA. While uniform tiling assumes equal latency and bandwidth across all compute units, this breaks down in MCM systems due to inter-chiplet communication costs. SIMBA addresses this with three optimizations: non-uniform work partitioning to balance load based on data proximity, communication-aware data placement to reduce latency by optimizing data locations, and cross-layer pipelining to increase utilization by executing different layers on separate chiplet clusters. These techniques improve performance by up to 16% compared to baseline uniform tiling.

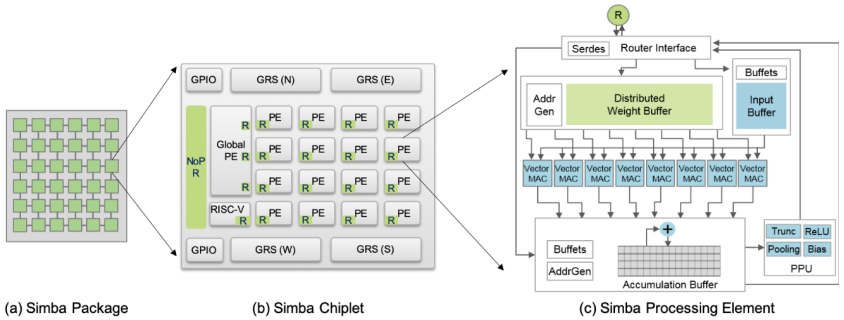

Fig. 1. Simba architecture from package to processing element (PE) [4]

2.2 NN-BATON

NN-BATON: Flexible System Mapping for Sparse DNN Accelerators NN-BATON (Neural Network Balancing And Task Organization for Networked accelerators), two fundamental challenges in chiplet-based DNN accelerator design: efficient workload mapping and optimal chiplet granularity. It introduces a systematic framework to reduce inter-chip communication overhead and improve resource utilization, addressing limitations found in single-die accelerators when scaled to multichip systems.

The architecture (illustrated in Fig. 2) models three levels of parallelism—package, chiplet, and core—where cores consist of PE arrays with local buffers, grouped into chiplets with shared memory and I/O, all interconnected via a ring network. To orchestrate computation, NN-Baton adopts an output-centric dataflow model using spatial primitives for partitioning and temporal primitives for loop unrolling, maximizing data reuse. For efficient evaluation, it introduces the C3P (Critical-Capacity Critical-Position) methodology, which analytically estimates memory access overhead based on loop structure and buffer sizes, enabling fast and informed design space exploration.

Overall, NN-Baton offers a practical and analytical approach to co-designing hardware and dataflow strategies in multichip DNN accelerators, making it a foundational reference for chiplet-aware mapping and architecture exploration.

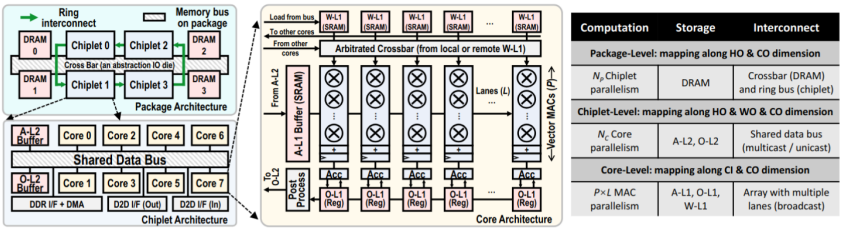

Fig. 2. NN-BATON: Three Level Architecture [6]

2.3 SIAM

SIAM (Scalable In-Memory Architecture for DNN Acceleration) is a first chiplet-based accelerator that uses analog in-memory computing (IMC) for matrix-vector multiplications in deep neural networks (DNNs). Unlike monolithic designs, SIAM employs a modular architecture with chiplets containing IMC cores for storing weights and performing analog Multiply-Accumulate (MAC) operations. It integrates digital interfaces, precision-tuning, and error-correction to address analog

non-idealities. Each chiplet includes buffers, ADCs, pooling, and activation units, connected via on-chip NoC and Network-on-Package (NoP) for efficient chiplet communication. A global controller manages DNN layer partitioning, scheduling, and memory access. SIAM offers significant energy efficiency, up to 130× improvement over traditional GPUs like NVIDIA V100 and T4, especially for networks like ResNet-50 on ImageNet.

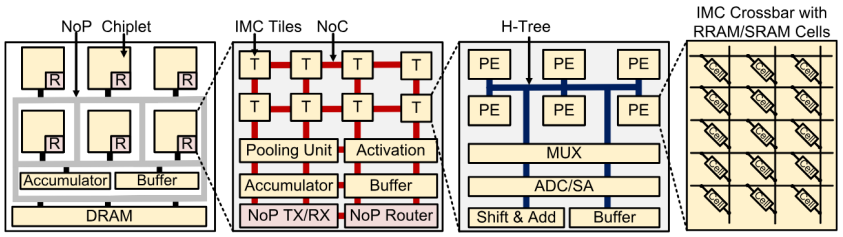

Fig. 3. SIAM architecture [2]

3 Methodology/Papers

3.1 INDM

Designing deep neural network (DNN) accelerators using chiplet-based architectures introduces several critical challenges that do not exist in monolithic designs. These include interdie communication overhead, inefficient dataflow mapping, and complex architectural partitioning. Interdie communication significantly increases latency and energy compared to monolithic designs. Traditional mapping algorithms, optimized for single-die architectures, fail to minimize cross-chiplet data movement. Additionally, partitioning compute and memory resources across multiple chiplets leads to a large design space, complicating system optimization. INDM addresses these issues through a co-optimized framework featuring a hierarchical interconnect network, design space exploration, and interdie-aware dataflow mapping.

3.1.1 Overall Architecture. The INDM architecture (illustrated in Fig. 4) organizes compute dies into clusters, with every four compute dies grouped into a single cluster. Each cluster is connected internally via a dedicated IO die, which handles communication among chiplets and with external memory. Clusters are interconnected through interdie links that pass through the IO dies, forming the backbone of the system-wide interconnect. DDR memory blocks are positioned around the periphery of the package substrate to reduce I/O path lengths and latency.

Within each compute die, processing elements (PEs) share L2 buffers and are connected via a multiring on-die network. The dataflow is managed to enable the reuse of weights and activations while reducing unnecessary data transfers between chiplets. The system supports flexible parallelism at the chiplet, die, and PE levels to match the characteristics of various DNN layers.

3.1.2 Hierarchical Interconnect Network. To support efficient data movement and reuse under typical DNN workloads, INDM proposes a two-level hierarchical interconnect network:

- On-die communication is handled through a multiring network, which supports high-bandwidth multicast and unicast communication among PEs and shared L2 buffers. This topology reduces router complexity and facilitates efficient weight/activation distribution.
- Interdie communication is managed via a cluster-based topology that links compute chiplets through IO dies. Each IO die integrates DDR PHYs and controllers, minimizing the area overhead while supporting high-bandwidth DRAM access.

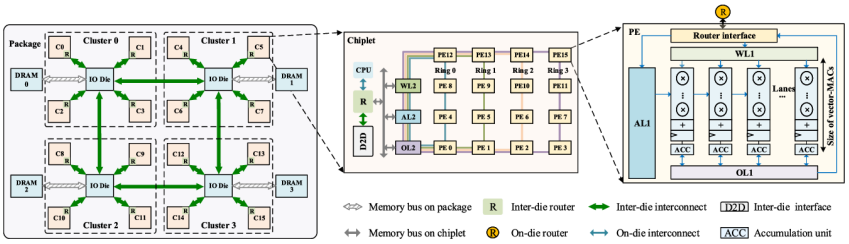

Fig. 4. INDM Architecture [7]

This network design takes advantage of DNN data reuse patterns, such as weight sharing and activation locality, to minimize traffic and balance bandwidth across the hierarchy.

3.1.3 Exploration of Architecture Partitioning. To navigate the vast design space introduced by chiplet partitioning, INDM incorporates a design space exploration (DSE) framework that jointly evaluates architectural configurations and dataflow mapping strategies. This framework optimizes parameters such as the number of chiplets, buffer sizes, and interconnect topology based on target DNN workloads and hardware constraints. It balances latency, energy, and area by systematically assessing trade-offs across design points. A key feature of the framework is its latency estimation model, which offers rapid yet accurate performance predictions, significantly reducing evaluation time compared to cycle-accurate simulations. By co-optimizing architecture and mapping, the DSE framework ensures that the resulting design is scalable, energy-efficient, and well-suited to deep learning applications' communication and computation demands.

3.2 Gemini

As chiplet-based architectures become essential for scaling DNN accelerators in the post-Moore era, they also introduce new design complexities—particularly in mapping workloads and balancing cost, energy, and performance. Gemini addresses these challenges through a unified framework that co-explores spatial mapping and architecture design, specifically targeting large-scale DNN inference accelerators.

3.2.1 Overall Architecture and Mapping Challenges. Gemini is built upon a highly configurable hardware template that models realistic chiplet systems with computing and I/O chiplets connected via a mesh NoC and D2D links. This template enables various architectural configurations with tunable parameters like core counts, chiplet division (XCut, YCut), memory bandwidth, and buffer sizes. A key insight motivating Gemini is the trade-off between chiplet granularity and cost/performance efficiency: smaller chiplets improve yield but suffer from higher packaging and D2D overheads, while larger chiplets reduce interconnect cost but risk area-related yield loss.

3.2.2 Architecture and Mapping Co-Exploration Framework. Gemini introduces a co-optimization framework with two main engines: a Mapping Engine and a Monetary Cost (MC) Evaluator. The Mapping Engine uses a Simulated Annealing (SA) algorithm with five custom operators to explore the encoded mapping space. It iteratively improves LP mapping schemes while automatically minimizing D2D communication. Alongside, an intra-core optimizer fine-tunes tiling and loop ordering for each core.

The MC Evaluator models silicon, DRAM, and packaging costs based on architectural parameters and chiplet layouts, providing accurate cost feedback during exploration. Gemini evaluates each design using a multi-objective function combining monetary cost, energy consumption, and delay (MC·E·D), guiding the search toward practical trade-offs.

3.3 Florets for Chiplets

Designing chiplet-based manycore systems for concurrent CNN inference introduces unique challenges in communication efficiency, data locality, and task mapping. Traditional Network-on-Interposer (NoI) designs use multi-hop topologies like mesh and torus, which are regular and workload-agnostic. While simple to implement, these topologies perform poorly when executing diverse, large-scale CNN workloads with high contiguous inter-layer data movement. They fail to exploit the communication patterns inherent to deep learning models, resulting in excessive inter-chiplet hops, increased latency and energy consumption, and poor scalability.

Florets addresses these limitations by leveraging *Space-Filling Curves (SFCs)* to construct a workload-aware, locality-preserving NoI topology optimized for 2.5D chiplet systems. Importantly, Florets is built atop a ReRAM-based Processing-in-Memory (PIM) architecture. By integrating ReRAM PIM within each chiplet, the design enables in-situ computation, significantly reducing data movement between compute and memory. This tight coupling of compute and memory complements the communication-efficient NoI design, enhancing both performance and energy efficiency for CNN inference workloads.

3.3.1 Overall Architecture. The Florets architecture organizes chiplets into multiple *SFC groups*, called Florets, where each floret (or petal) cluster represents a linear contiguous path through a subset of chiplets. These florets enable efficient mapping of CNN layers to chiplets such that successive layers are placed on neighboring chiplets, reducing communication hops. Each SFC (petal) is a disjoint path with a dedicated head and tail, and inter-SFC links connect the tail of one Floret to the head of another, forming a scalable, hierarchical NoI structure.

Florets avoid uniform SFCs by assigning variable-length paths to different florets based on task requirements. A Traveling Salesman Problem (TSP)-based heuristic is used to generate low-cost SFC paths. CNN inference tasks are dynamically mapped to available chiplet sequences along these curves, ensuring minimal inter-floret transitions.

3.3.2 Space-Filling Curve-Based NoI. The NoI is constructed using multiple SFCs (shown in Fig. 5), each optimized for locality and minimal communication overhead. Features include:

- Multiple disjoint SFCs that enable concurrent CNN execution by dividing the chiplets into separate linear paths.
- Workload-aware mapping aligns the CNN layer graph with the SFC paths to reduce hops.
- Inter-SFC links for flexibility and redundancy, enabling layers that cannot fit within one SFC to spill over with minimal latency.

This design improves the spatial locality of neural layer execution and minimizes long-distance data movement, significantly enhancing throughput and energy efficiency.

3.3.3 CNN-Aware Task Mapping Strategy. Florets includes a dynamic layer mapping algorithm that:

- Assigns CNN layers to contiguous chiplets along SFCs while balancing chiplet utilization.
- Supports multi-layer-per-chiplet and multi-chiplet-per-layer configurations.
- Minimizes inter-SFC communication by optimizing path lengths between head-tail pairs.

The algorithm ensures deadlock-free execution through sequential task allocation and considers resource reuse for future tasks. Evaluations show that Florets reduces communication latency and energy by up to **58%** and **64%**, while executing a diverse workload of CNN inference tasks. It also significantly reduces the fabrication costs by up to **82%** compared to existing NoI architectures.

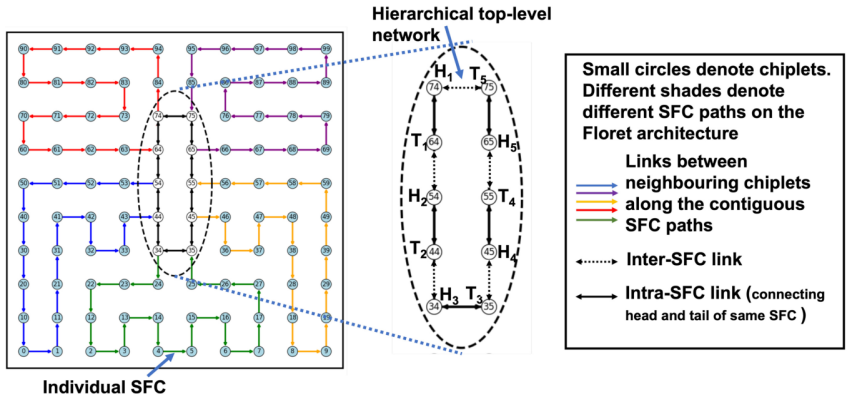

Fig. 5. Illustration of the SFC-based architecture called Floret for a 100-chiplet-based system with five SFCs on the interposer network. [5]

3.4 TEFLON

While SFC-based NoI designs like Florets improve communication efficiency, they do not address thermal constraints, which are critical in tightly integrated 3D manycore architectures, especially those based on ReRAM-based processing-in-memory (PIM). CNN inference on 3D systems introduces thermal hotspots that degrade performance and inference accuracy due to ReRAM conductance drift. TEFLON, an extension of Florets, proposes a thermally aware 3D NoC design methodology that optimizes for *latency, energy, temperature*, and *accuracy* using multi-SFC paths in a monolithic 3D (M3D) environment.

3.4.1 Overall Architecture. TEFLON structures its architecture around multiple 3D SFCs, where each SFC links a group of PEs across planar tiers using both horizontal and vertical links. Each SFC has a defined head and tail, and inter-SFC links enality and redundancy. TEFLON avoids stacking hivertically gh-power PEs (e.g., early CNN layers) vertically to prevent thermal hotspots. Instead, it places these layers closer to the heat sink and spreads the thermal load across tiers and planes.

Each PE integrates ReRAM crossbars and routers, and TEFLON adapts the placement of layers based on their power consumption profiles, ensuring both thermal safety and communication efficiency.

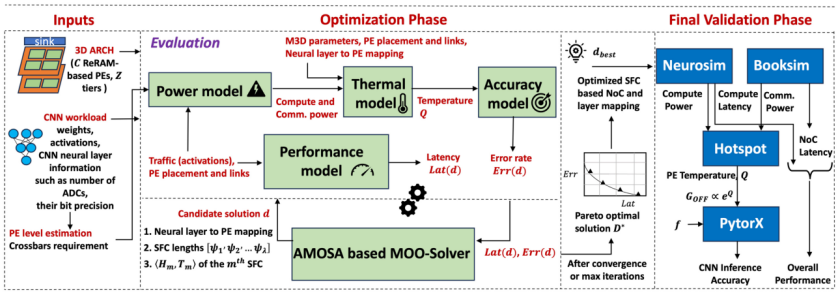

Fig. 6. Overall Workflow of TEFLON [3]

3.4.2 Multi-Objective SFC-Based NoC Design. TEFLON uses a multi-objective optimization (MOO) framework to jointly minimize:

- Latency (Lat) — based on activation dataflow between layers.
- Inference error (Err) — estimated from the thermal impact on ReRAM accuracy.

Key features include:

- Dynamic PE placement along SFCs using a TSP-based heuristic.
- Trade-off tuning between performance and accuracy using scalarization parameters.
- Selection of the optimal number and lengths of SFCs to balance load, robustness, and power.

The NoC is evaluated using latency, temperature, power, and error estimation models, and Pareto-optimal configurations are selected for deployment.

3.4.3 Thermal-Accuracy-Performance Tradeoffs. TEFLON integrates:

- HotSpot thermal modeling to prevent overheating.
- Weight deviation modeling in ReRAM due to thermal noise.
- Pruned CNN models (via coarse-to-fine pruning) to reduce power demands.

TEFLON improves inference accuracy (e.g., by up to 11% over Florets), reduces peak chip temperature (by up to 25K), and achieves a 42–46% reduction in Energy-Delay Product (EDP) over traditional mesh NoCs. It also introduces bypass links and router port optimization for further performance gains.

4 Experimental Trends

INDM architecture is evaluated against two leading designs: SIMBA, which uses a mesh-based interconnect, and NN-Baton, which employs ring and crossbar topologies. Both serve as baselines but struggle with scalability and interdie communication efficiency. INDM conducts an exhaustive design space exploration under a strict 3 mm^2 die area constraint. Despite this, it identifies an optimal architecture with only 3.7% EDP overhead compared to the unconstrained design. Area analysis reveals L1/L2 buffers and PE MACs as the primary components, while energy is mainly consumed by DRAM and interdie communication. As the chiplet count scales from 8 to 16 (8–128 TOPS), INDM consistently reduces latency, maintains energy efficiency, and improves overall EDP. It achieves an average 26.93%-79.78% latency and 26%-73.81% EDP reduction over SIMBA and NN-Baton, demonstrating its effectiveness across deployment scenarios.

For **Gemini**, we perform DSE on accelerators with 72, 128, and 512 TOPs to assess its architecture and mapping co-exploration capabilities. We fix compute power and vary architectural parameters like Chiplet/Core count, DRAMBW, NoCBW, D2DBW, GBUF/Core, and MAC/Core. Core arrays are arranged in near-square grids with only valid XCut/YCut values. The default setup includes TSMC 12nm, 1 GHz, organic substrate, and GRS-based D2D. We use Transformer as the main workload, with additional evaluations on ResNet-50, ResNeXt, Inception-ResNet, and PNASNet. Throughput- and latency-centric scenarios are tested using batch sizes of 64 and 1, respectively. We compare Gemini's explored design (G-Arch, G-Map) against baselines: Simba-based (S-Arch, T-Map) and Tenstorrent-based (T-Arch, T-Map).

- G-Arch vs. S-Arch: G-Arch with G-Map reduces delay by 46.8% and energy by 28.8% over S-Arch with T-Map, with only 14.3% more MC. This is achieved by reducing chiplet count, increasing interconnect bandwidth, and doubling buffer capacity—minimizing D2D overhead.
- G-Arch vs. T-Arch: Compared to T-Arch with T-Map, Gemini improves performance by 1.74×, energy efficiency by 1.13×, and reduces MC by 40.1%, demonstrating its effectiveness and generality.

Floret architecture is evaluated against state-of-the-art Network-on-Interposer (NoI) topologies such as *SIMBA* and *NN-Baton* in the context of 2.5D chiplet-based systems executing concurrent CNN inference tasks. These baselines are optimized for single-task performance but do not scale well under multi-task workloads due to lack of dataflow awareness.

Floret introduces a novel space-filling curve (SFC)-based interconnect across chiplets, enabling contiguous mapping of successive CNN layers and minimizing long-range communication. The evaluation uses ReRAM-based chiplets for processing and runs multiple CNN variants including ResNet, VGG, and DenseNet.

Compared to existing mesh-based and ring topologies, Floret consistently reduces latency by up to 58% and energy consumption by up to 64%, even in datacenter-scale workloads. It introduces multiple SFCs (Florets) that provide inherent redundancy, scalability, and performance isolation across tasks. It also significantly reduces the fabrication costs by up to 82% compared to existing NoI architectures. Experimental studies demonstrate Floret's effectiveness in maintaining performance across increasing system sizes and concurrent inference loads, outperforming all baselines in terms of latency, energy efficiency, and chiplet utilization.

TEFLON is evaluated under a 3D manycore Processing-in-Memory (PIM) system with 36-, 64-, and 100-core configurations to assess its joint optimization of performance, temperature, and inference accuracy. It is benchmarked against conventional 3D mesh-based NoC architectures and the Floret-enabled SFC NoC which lacks thermal awareness.

TEFLON applies a thermally-aware, multi-SFC mapping strategy tailored for ReRAM-based CNN inference. It reduces Energy-Delay Product (EDP) by 42%, 46%, and 45% on average for the 36-, 64-, and 100-PE systems respectively. Additionally, it lowers peak chip temperature by up to 25 K and improves inference accuracy by up to 11% compared to the Floret baseline.

TEFLON dynamically adjusts PE placement and layer mapping along the 3D SFC paths to avoid thermal hotspots while maintaining contiguity for high-activation layers. The framework uses a multi-objective optimization algorithm to jointly minimize latency and error rate under thermal constraints. It's scalable performance and thermal robustness are validated across CIFAR-10/100 workloads using diverse CNN models including *VGG*, *ResNet*, and *DenseNet*. This robustness across varied CNN topologies highlights TEFLON's generalizability and effectiveness in real-world, high-throughput many-core inference scenarios.

5 Challenges and Future Directions

Chiplet-based in-memory computing offers significant potential, but several challenges remain. Inter-chiplet communication is a major bottleneck, requiring advanced, dataflow-aware interconnect designs. Efficient dataflow mapping and workload partitioning across heterogeneous chiplets remain complex and require co-optimized hardware-software strategies. Standardization of chiplet interfaces is lacking, limiting scalability and integration. Additionally, thermal management in dense 2.5D/3D IMC systems is critical, as overheating can degrade performance and accuracy—especially in analog ReRAM-based designs.

Future work should focus on holistic co-design frameworks that jointly optimize architecture, mapping, and interconnects. Thermal-aware NoC and dynamic layer placement are essential for reliability, while standardized interfaces and adaptive runtime systems can improve portability and efficiency. Advancements in compiler support and reconfigurable scheduling will also be key to unlocking the full potential of chiplet-based accelerators for DL workloads.

6 Conclusion

As Deep Learning models continue to scale in size and complexity, traditional monolithic accelerators face growing limitations in performance, energy efficiency, and manufacturability. This paper

has examined chiplet-based architectures combined with in-memory computing as a promising alternative, offering modularity, scalability, and reduced communication overheads. By analyzing recent advances—including INDM's hierarchical interconnects and interdie-aware dataflow mapping, Gemini's architecture-mapping co-exploration, Floret's communication-aware NoI design, and TEFLON's thermally optimized 3D layout—we have highlighted how thoughtful hardware-software co-design can mitigate challenges such as dataflow mapping, workload partitioning, inter-chiplet communication, and thermal constraints. Experimental results demonstrate significant improvements in latency, energy efficiency, and fabrication cost, reinforcing the potential of chiplet-based IMC for next-generation DNN accelerators. While integration and standardization remain ongoing challenges, this paradigm is well-positioned to drive the development of scalable, high-performance DL hardware systems.

Acknowledgments

This paper was completed under the guidance of Dr. Rajesh Kedia, whose support and insights were essential throughout its development.

References

[1] Jingwei Cai, Zuotong Wu, Sen Peng, Yuchen Wei, Zhanhong Tan, Guiming Shi, Mingyu Gao, and Kaisheng Ma. 2024. Gemini: Mapping and Architecture Co-exploration for Large-scale DNN Chiplet Accelerators. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 156–171. doi:10.1109/HPCA57654.2024.00022

[2] Gokul Krishnan, Sumit K. Mandal, Manvitha Pannala, Chaitali Chakrabarti, Jae-Sun Seo, Umit Y. Ogras, and Yu Cao. 2021. SIAM: Chiplet-based Scalable In-Memory Acceleration with Mesh for Deep Neural Networks. *ACM Trans. Embed. Comput. Syst.* 20, 5s, Article 68 (Sept. 2021), 24 pages. doi:10.1145/3476999

[3] Gaurav Narang, Chukwufumnanya Ogbogu, Janardhan Rao Doppa, and Partha Pratim Pande. 2024. TEFLON: Thermally Efficient Dataflow-aware 3D NoC for Accelerating CNN Inferencing on Manycore PIM Architectures. *ACM Trans. Embed. Comput. Syst.* 23, 5, Article 78 (Aug. 2024), 23 pages. doi:10.1145/3665279

[4] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel Emer, C. Thomas Gray, Brucek Khailany, and Stephen W. Keckler. 2019. Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* (Columbus, OH, USA) *(MICRO '52)*. Association for Computing Machinery, New York, NY, USA, 14–27. doi:10.1145/3352460.3358302

[5] Harsh Sharma, Lukas Pfromm, Rasit Onur Topaloglu, Janardhan Rao Doppa, Umit Y. Ogras, Ananth Kalyanraman, and Partha Pratim Pande. 2023. Florets for Chiplets: Data Flow-aware High-Performance and Energy-efficient Network-on-Interposer for CNN Inference Tasks. *ACM Trans. Embed. Comput. Syst.* 22, 5s, Article 132 (Sept. 2023), 21 pages. doi:10.1145/3608098

[6] Zhanhong Tan, Hongyu Cai, Runpei Dong, and Kaisheng Ma. 2021. NN-baton: DNN workload orchestration and chiplet granularity exploration for multichip accelerators. In *Proceedings of the 48th Annual International Symposium on Computer Architecture* (Virtual Event, Spain) *(ISCA '21)*. IEEE Press, 1013–1026. doi:10.1109/ISCA52012.2021.00083

[7] Jinming Zhang, Xi Fan, Yaoyao Ye, Xuyan Wang, Guojie Xiong, Xianglun Leng, Ningyi Xu, Yong Lian, and Guanghui He. 2024. INDM: Chiplet-Based Interconnect Network and Dataflow Mapping for DNN Accelerators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 43, 4 (2024), 1107–1120. doi:10.1109/TCAD.2023.3332832