computers

MDPI

Review

A Comprehensive Review of Processing-in-Memory Architectures for Deep Neural Networks

Rupinder Kaur *,†, Arghavan Asad † and Farah Mohammadi †

Electrical, Computer and Biomedical Engineering Department, Toronto Metropolitan University, 350 Victoria St, Toronto, ON M5B 2K3, Canada; arghavan.asad@torontomu.ca (A.A.); fmohamma@torontomu.ca (F.M.)

* Correspondence: rupinder.kaur.ece@torontomu.ca

† These authors contributed equally to this work.

Abstract: This comprehensive review explores the advancements in processing-in-memory (PIM) techniques and chiplet-based architectures for deep neural networks (DNNs). It addresses the challenges of monolithic chip architectures and highlights the benefits of chiplet-based designs in terms of scalability and flexibility. This review emphasizes dataflow-awareness, communication optimization, and thermal considerations in PIM-enabled manycore architectures. It discusses tailored dataflow requirements for different machine learning workloads and presents a heterogeneous PIM system for energy-efficient neural network training. Additionally, it explores thermally efficient dataflow-aware monolithic 3D (M3D) NoC architectures for accelerating CNN inferencing. Overall, this review provides valuable insights into the development and evaluation of chiplet and PIM architectures, emphasizing improved performance, energy efficiency, and inference accuracy in deep learning applications.

Keywords: deep neural network (DNN); processing-in-memory (PIM); heterogeneous architecture; resistive ReRAM (ReRAM); network on chip (NoC); latency; power; accuracy

Citation: Kaur, R.; Asad, A.; Mohammadi, F. A Comprehensive Review of Processing-in-Memory Architectures for Deep Neural Networks. *Computers* **2024**, *13*, 174. https://doi.org/10.3390/computers13070174

Academic Editor: Paolo Bellavista

Received: 12 June 2024
Revised: 5 July 2024
Accepted: 9 July 2024
Published: 16 July 2024

1. Introduction

Deep learning has emerged as a powerful technique for solving complex problems across various domains, including computer vision, natural language processing, and robotics [1–3]. The success of deep neural networks (DNNs) in these domains has led to an increasing demand for efficient hardware architectures that can accelerate the training and inference processes. Traditional von Neumann architectures, with their separated CPU and memory units, struggle to meet the high computational demands and data movement requirements of deep learning workloads [4–10]. To address these challenges, researchers have been exploring innovative approaches such as chiplet-based architectures and processing-in-memory (PIM) techniques.

Chiplet-based architectures offer a promising solution to the limitations of monolithic chip designs in deep learning. Monolithic chips, with their large area and on-chip interconnection costs, face challenges in scaling up to accommodate the growing sizes of deep learning models. Chiplet-based architectures, on the other hand, divide the system into smaller interconnected units called chiplets, allowing for improved scalability, modularity, and flexibility. These chiplets can be designed and optimized independently, leading to better yield and reduced costs. Furthermore, chiplet-based architectures enable efficient utilization of resources by distributing the computational workload across multiple chiplets, resulting in improved performance and energy efficiency [11–16].

In parallel, processing-in-memory (PIM) techniques have gained significant attention as a means to overcome the memory bottleneck in deep learning. In traditional architectures, data movement between the processor and memory units consumes a significant amount of energy and time. PIM architectures aim to alleviate this bottleneck by integrating

processing units directly into the memory subsystem. By performing computations in close proximity to the data, PIM architectures minimize data movement, reduce latency, and improve energy efficiency. PIM architectures can leverage emerging memory technologies like resistive random-access memory (ReRAM) to achieve high-performance and energy-efficient acceleration of deep learning tasks [17–21].

1.1. Motivation behind Review

The limitations of monolithic chip designs, including their large area and on-chip interconnection costs, make it difficult for them to scale up and accommodate the growing sizes of deep learning models. In contrast, chiplet-based architectures offer a promising solution by dividing the system into smaller interconnected units called chiplets. This division enables improved scalability, modularity, and flexibility. Chiplet-based architectures allow for independent design and optimization of chiplets, leading to better yield and reduced costs. By distributing the computational workload across multiple chiplets, chiplet-based architectures can efficiently utilize resources, resulting in improved performance and energy efficiency [22–28].

Moreover, the memory bottleneck in traditional architectures, where data movement between the processor and memory units consumes significant energy and time, can be addressed through PIM techniques. PIM architectures integrate processing units directly into the memory subsystem, minimizing data movement, reducing latency, and improving energy efficiency. These architectures leverage emerging memory technologies like resistive random-access performance and energy-efficient acceleration of deep learning memory (ReRAM) to achieve high-tasks [29–33].

Therefore, the motivation behind this review is to explore the advancements in chiplet-based architectures and PIM techniques for deep learning applications, highlighting their potential to revolutionize deep learning hardware. By leveraging the benefits of chiplet-based architectures and PIM techniques, researchers and engineers can overcome the challenges faced by traditional von Neumann architectures and contribute to the development of efficient and powerful AI hardware.

1.2. Gaps in Current Research

This review aims to address several gaps in the current research on processing-in-memory (PIM) architectures for deep neural networks. Some of the identified gaps addressed by this review are as follows:

1. Lack of comprehensive understanding: This review acknowledges the need for a comprehensive understanding of PIM techniques and their potential in revolutionizing deep learning hardware. It provides valuable insights into the advancements in chiplet-based architectures and PIM techniques, emphasizing their benefits in terms of performance, energy efficiency, scalability, and flexibility.
2. Limited exploration of chiplet-based architectures: Traditional monolithic chip designs face challenges in scaling up to accommodate the growing sizes of deep learning models. This review highlights chiplet-based architectures as a promising solution to these limitations and discusses their advantages in terms of improved scalability, modularity, and flexibility. It emphasizes the efficient utilization of resources and distribution of computational workload across multiple chiplets for enhanced performance and energy efficiency.
3. Insufficient focus on dataflow-awareness and communication optimization: This review recognizes the importance of dataflow-awareness and communication optimization in the design of PIM-enabled manycore architectures. It discusses the tailored dataflow requirements of different machine learning workloads and emphasizes the optimization of PIM architectures to minimize latency and improve energy efficiency. It also addresses the challenges associated with on-chip interconnection networks and the need for scalable communication in chiplet-based architectures.

4. Limited exploration of thermal considerations: Thermal constraints pose significant challenges in the design of PIM architectures. This review highlights the importance of thermal considerations and discusses thermally efficient dataflow-aware monolithic 3D NoC architectures for accelerating CNN inferencing. It compares different architectures and emphasizes the advantages of thermally efficient designs, such as TEFLON (thermally efficient dataflow-aware 3D NoC), in terms of energy efficiency, inference accuracy, and thermal resilience.
5. Inadequate exploration of programming models and hardware utilization: This review presents a heterogeneous PIM system for energy-efficient neural network training. It addresses the significance of programming models that accommodate both fixed-function logics and programmable cores, providing a unified programming model and runtime system for efficient task offloading and scheduling. It emphasizes achieving balanced hardware utilization in heterogeneous systems with abundant operation-level parallelism.
6. Limited analysis of cybersecurity challenges: This review acknowledges the cybersecurity challenges associated with deep neural networks (DNNs) and their increased attack surface. It discusses adversarial attacks, model stealing attacks, and concerns regarding privacy and data leakage. While the focus of this review is primarily on hardware architectures, this section provides an important perspective on the security implications of deploying DNNs.

By addressing these gaps, this comprehensive review contributes to the existing knowledge by providing insights into the development and evaluation of chiplet and PIM architectures. It emphasizes the improved performance, energy efficiency, and inference accuracy in deep learning applications. This review's coverage of various aspects, including chiplet-based architectures, PIM techniques, dataflow-awareness, thermal considerations, programming models, and cybersecurity challenges, makes it a valuable resource for researchers and engineers working in the field of deep learning hardware.

1.3. Key Insights

This comprehensive review aims to provide insights into the advancements in chiplet-based architectures and processing-in-memory techniques for deep learning applications. It explores the challenges faced by monolithic chip architectures and highlights the potential of chiplet-based designs in addressing these challenges [34–42]. This review familiarizes SIAM (scalable in-memory acceleration with mesh), a benchmarking simulator that evaluates chiplet-based in-memory computing (IMC) architectures, and showcases the flexibility and scalability of SIAM through benchmarking different deep neural networks.

Furthermore, this review delves into the design considerations of processing-in-memory architectures for deep learning workloads. It emphasizes the importance of dataflow-awareness and communication optimization in the design of PIM-enabled many-core platforms. By understanding the unique traffic patterns and data exchange requirements of different machine learning workloads, PIM architectures can be optimized to minimize latency and improve energy efficiency. This review also discusses the challenges associated with on-chip interconnection networks, thermal constraints, and the need for scalable communication in chiplet-based architectures.

Additionally, this review presents a heterogeneous PIM system for energy-efficient neural network training. This approach combines fixed-function arithmetic units and programmable cores on a 3D die-stacked memory, providing a unified programming model and runtime system for efficient task offloading and scheduling. This review highlights the significance of programming models that accommodate both fixed-function logics and programmable cores, as well as achieving balanced hardware utilization in heterogeneous systems with abundant operation-level parallelism [43–45].

Finally, this review explores thermally efficient dataflow-aware monolithic 3D (M3D) NoC architectures for accelerating CNN inferencing. It discusses the benefits of integrating processing-in-memory cores using ReRAM technology and emphasizes the importance

of efficient network-on-chip (NoC) designs to reduce data movement. This review compares different architectures and highlights the advantages of TEFLON (thermally efficient dataflow-aware 3D NoC) over performance-optimized space-filling curve (SFC)-based counterparts in terms of energy efficiency, inference accuracy, and thermal resilience.

In summary, the advancements in chiplet-based architectures and processing-in-memory techniques have the potential to revolutionize deep learning hardware. These approaches offer scalability, flexibility, improved performance, and energy efficiency, addressing the challenges faced by traditional monolithic chip designs. By leveraging the benefits of chiplet-based architectures and processing-in-memory techniques, researchers and engineers can pave the way for enhanced deep learning capabilities and contribute to the development of efficient and powerful AI hardware [30,31,46,47].

This review is divided into several sections, each focusing on different aspects of processing-in-memory architectures for deep neural networks. Figure 1 illustrates the layout of the article and highlights the key challenges associated with PIM-enabled manycore architectures. The figure provides a visual representation of the main challenges addressed in this review. It depicts a circular layout with various components and arrows connecting them. The components represent different aspects of PIM-enabled manycore architectures, while the arrows indicate the interconnected relationships and challenges between these components. Section 1 provides an overview of the challenges faced by traditional architectures and the potential solutions offered by chiplet-based designs and processing-in-memory (PIM) techniques. Section 2 then delves into the details of PIM, discussing its innovative approach of integrating computational units into the memory subsystem and the benefits it brings in terms of performance, energy efficiency, and scalability. The challenges associated with implementing PIM in heterogeneous CPU–GPU architectures are explored, including memory organization, programming models, data movement, and power/thermal constraints [48–50]. Section 3 highlights the importance of dataflow-awareness, communication optimization, and thermal considerations in designing PIM-enabled manycore architectures. Furthermore, it discusses a heterogeneous PIM system for energy-efficient neural network training and thermally efficient dataflow-aware monolithic 3D NoC architectures for accelerating CNN inferencing. Section 4 addresses the cybersecurity challenges associated with deep neural networks (DNNs). It discusses the increased attack surface due to the growth of AI capabilities and explores adversarial attacks, model stealing attacks, and concerns regarding privacy and data leakage. Finally, this review concludes by emphasizing the potential of PIM techniques in revolutionizing deep learning hardware and contributing to the development of efficient AI hardware [32].

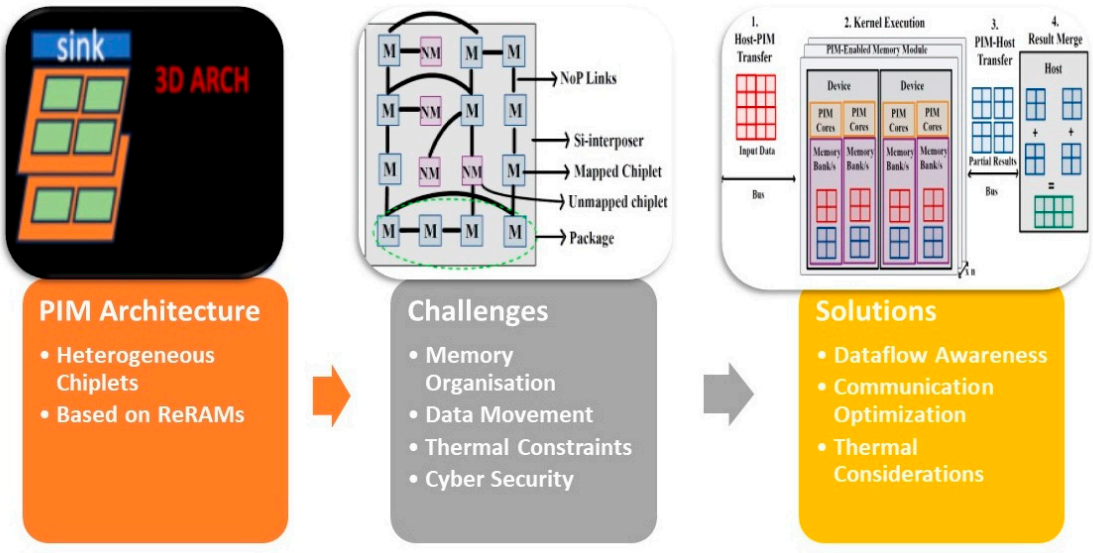

Figure 1. Layout of the article depicting key challenges of PIM-enabled manycore architectures.

1.4. Research Strategy and Data Extraction Methodology

The literature search strategy for this review involved searching various academic databases and search engines such as IEEE Xplore, ACM Digital Library, and Google Scholar. The search utilized keywords and search terms including PIM, chiplet-based architectures, deep learning, deep neural networks (DNNs), memory subsystem, dataflow-awareness, communication optimization, thermal considerations, resistive ReRAM, network-on-chip (NoC), latency, power, accuracy, heterogeneous architecture, machine learning workloads, computational units, memory technologies, Von Neumann bottleneck, performance optimization, energy efficiency, and scalability. The inclusion criteria focused on selecting articles directly related to PIM architectures, chiplet-based designs, and their applications in deep learning, including novel research, advancements, case studies, and experimental evaluations. The literature search covered several years to include both foundational works and recent research articles. The selection criteria for this review included relevance to PIM, recent advancements, comparative analysis of PIM architectures, and emphasis on dataflow-awareness and communication optimization. The data extraction process involved identifying and extracting key variables or data points such as performance metrics (training time, inference time, speedup, throughput, accuracy), energy efficiency (energy consumption, energy efficiency, power consumption), and scalability (chiplet-based designs, system size, resource utilization, performance scaling) from the selected studies.

2. Processing-in-Memory (PIM)

2.1. Introduction

Processing-in-memory (PIM) is an innovative approach that aims to overcome the memory bottleneck in traditional computer architectures by integrating computational units directly into the memory subsystem. With the rapid growth of data-intensive applications, such as deep learning, PIM has gained significant attention as a promising solution for improving performance, energy efficiency, and overall system scalability. In traditional computer architectures, the processor and memory units are separate entities, requiring frequent data movement between them. This data movement, often referred to as the von Neumann bottleneck, consumes a significant amount of energy and introduces latency, limiting the overall system performance. As the computational demands of modern applications continue to increase, the memory subsystem becomes a critical performance bottleneck. PIM architectures aim to address this bottleneck by bringing processing units closer to the data. By integrating computational units, such as arithmetic units or accelerators, into the memory cells or in close proximity to them, PIM architectures enable computations to be performed directly on the data, minimizing the need for data movement. This approach not only reduces energy consumption but also improves system performance by reducing memory access latency. Various memory technologies can be leveraged in PIM architectures, including static random-access memory (SRAM), dynamic random-access memory (DRAM), and emerging non-volatile memory technologies such as resistive random-access memory (ReRAM) and phase-change memory (PCM) [33,34]. These memory technologies offer different trade-offs in terms of density, access speed, power consumption, and endurance, and can be tailored to suit specific PIM design requirements. PIM architectures have shown promising results in a wide range of applications, particularly in data-intensive domains such as artificial intelligence, machine learning, and big data analytics. Deep learning, in particular, benefits greatly from PIM architectures as they can significantly reduce the data movement between the processor and memory during the training and inference processes, leading to improved energy efficiency and faster computation.

2.2. Challenges

Processing-in-memory (PIM) refers to the integration of processing elements within the memory subsystem of a computing system. Heterogeneous CPU–GPU architectures, which combine central processing units (CPUs) and graphics processing units (GPUs), can

benefit from PIM to improve performance and energy efficiency. However, there are several challenges associated with implementing PIM in heterogeneous CPU–GPU architectures. Here are some of the key challenges:

1. Memory organization: PIM requires a rethinking of memory organization to enable processing elements within the memory subsystem. CPUs and GPUs have different memory access patterns and requirements, which need to be accommodated in the design. Efficiently organizing and managing data in a PIM architecture can be complex, especially when dealing with heterogeneous processing units.
2. Programming model: PIM architectures require a programming model that allows developers to express data and task parallelism effectively. Developing software for PIM architectures can be challenging due to the need for explicit data placement and synchronization between the CPU and GPU components. The programming models need to be designed to fully exploit the potential parallelism offered by PIM while maintaining ease of use.
3. Data movement: Efficient data movement is crucial for PIM architectures. Moving data between the CPU and GPU components can incur significant overhead due to the communication between different memory spaces. Minimizing data movement and optimizing data transfer mechanisms become essential for achieving high performance in heterogeneous CPU–GPU architectures.
4. Power and thermal constraints: PIM architectures can potentially consume significant power due to the increased integration of processing elements within the memory subsystem. Managing power and thermal constraints in heterogeneous CPU–GPU architectures is critical to prevent overheating and ensure reliable operation. Designing efficient power management techniques that balance performance and energy consumption is a significant challenge.
5. Memory consistency and coherence: Maintaining memory consistency and coherence in PIM architectures is complex, particularly in heterogeneous CPU–GPU systems. CPUs and GPUs often have their own caches and memory hierarchies, which need to be synchronized to ensure data integrity and correctness. Developing efficient coherence protocols and memory consistency models for heterogeneous PIM architectures is a non-trivial task.
6. Hardware design and integration: Hardware design challenges arise when integrating processing elements within the memory subsystem. PIM architectures require modifications to the memory controller, cache hierarchy, and interconnects to enable efficient data processing within memory. Co-designing the hardware components and optimizing the integration of processing elements in a heterogeneous CPU–GPU architecture is a significant challenge.

3. PIM-Based Systems

Researchers and engineers are actively working on overcoming these obstacles to fully exploit the benefits of processing-in-memory in heterogeneous CPU–GPU architectures.

The following sub-sections provides a comprehensive review that addresses these obstacles and offers potential solutions for maximizing the benefits of processing-in-memory in heterogeneous CPU–GPU architectures.

3.1. Heterogeneous PIM Architecture

The challenges associated with training neural networks, particularly deep neural networks (DNNs), arise from the significant energy consumption and time overhead caused by frequent data movement between the processor and memory. Ongoing research aims to maximize the benefits of processing-in-memory in heterogeneous CPU–GPU architectures by overcoming these obstacles.

One such approach is proposed in [1] as a hardware design and involves integrating fixed-function arithmetic units and programmable cores on the logic layer of a 3D die-stacked memory. Figure 2 illustrates a 3D die-stacked memory configuration where

fixed-function arithmetic units and programmable cores are integrated on the logic layer. This integration forms a heterogeneous processing-in-memory (PIM) architecture, which is connected to the CPU. The aim is to minimize data movement and improve system performance by bringing processing capabilities closer to the memory. In addition to the hardware design, a software design is presented, which includes a programming model and runtime system. These components enable programmers to develop, offload, and schedule various neural network training operations across the CPU and the heterogeneous PIM architecture. The objective is to achieve program portability, facilitate program maintenance, enhance system energy efficiency, and improve hardware utilization. By combining the proposed hardware and software designs, a comprehensive solution is offered to address the challenges of energy consumption and data movement during neural network training. The heterogeneous PIM architecture, accompanied by the programming model and runtime system, provides an effective approach for efficient neural network training by leveraging the advantages of processing-in-memory techniques.

The challenges of programming processing-in-memory (PIM) architectures for neural network acceleration in heterogeneous systems with fixed-function logics and programmable cores are non-trivial. One key requirement is a unified programming model that can effectively handle the heterogeneity of PIM architectures. Achieving balanced hardware utilization in such heterogeneous systems is another challenge, particularly in harnessing operation-level parallelism for efficient execution of neural network training workloads. This architecture [1] aims to minimize data movement and enhance energy efficiency by performing computations in close proximity to the data. To enable programming flexibility, the OpenCL programming model has been extended to accommodate the heterogeneity of PIM architectures. This extension allows developers to express parallelism and take advantage of both fixed-function logics and programmable cores. Insights into the characteristics of neural network training workloads have been provided, showcasing profiling results of time-consuming and memory-intensive operations across different training models. The significance of reducing data movement is emphasized, motivating the adoption of PIM architectures. The combination of hardware and software design techniques aims to improve performance, energy efficiency, and hardware utilization in heterogeneous CPU–GPU systems with PIM capabilities.

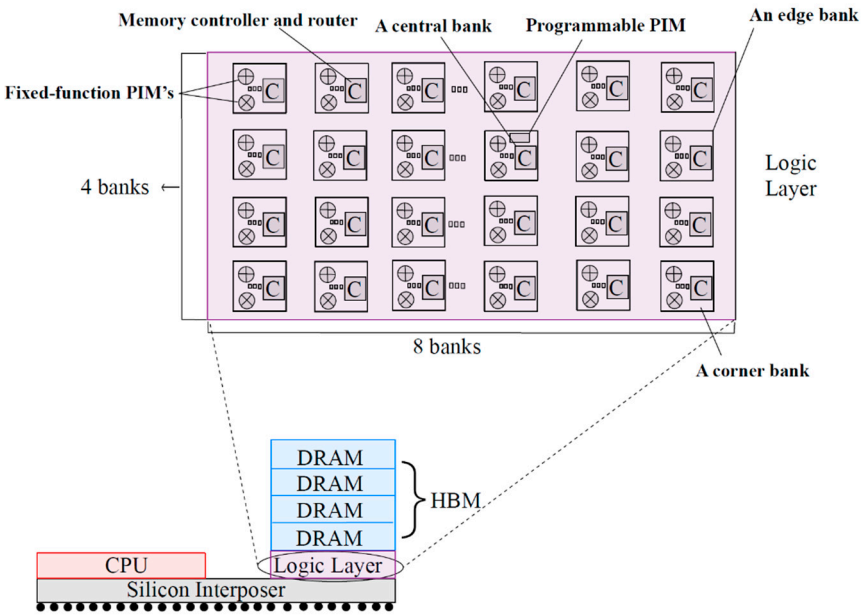

Figure 2. Illustration of a heterogeneous PIM architecture [1].

Another study [7] offers recommendations for software designers, insights into workload suitability for the PIM system, and suggestions for future hardware and architecture

designers of PIM systems. It discusses the concept of processing-in-memory (PIM) as a solution to the data movement bottleneck in memory-bound workloads. It introduces the UPMEM (Universal Processing Memory) PIM architecture, which combines DRAM memory arrays with in-order cores called DRAM processing units (DPUs) integrated in the same chip. Figure 3 depicts the architecture of the UPMEM PIM system, which is introduced as a solution to address the data movement bottleneck in memory-bound workloads. The figure illustrates how the DRAM memory arrays and DPUs are interconnected within the PIM system. This integration enables the DPUs to operate directly on the data stored in the DRAM memory arrays, minimizing the need for data movement between the CPU and memory. As a result, the UPMEM PIM architecture aims to improve system performance and reduce energy consumption. The study presents key takeaways from the comprehensive analysis of the UPMEM PIM architecture.

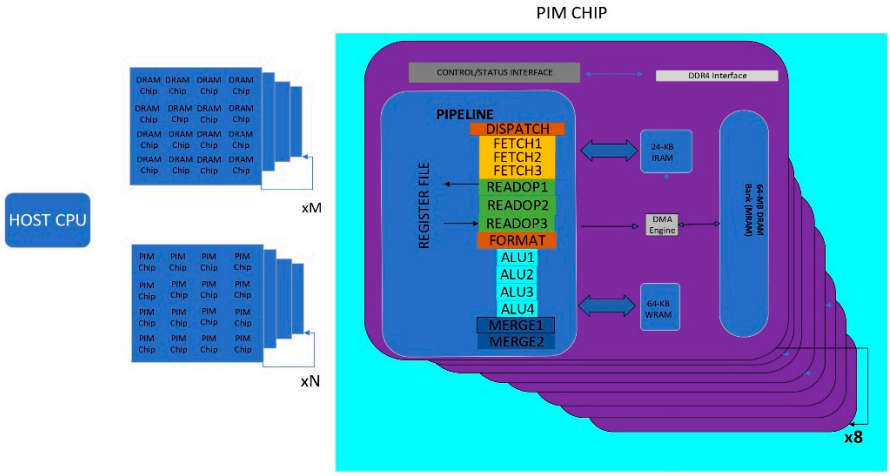

Figure 3. PIM-enabled system with a host CPU, standard main memory, and UPMEM-based PIM-enabled memory (**left**), and internal components of a UPMEM PIM chip (**right**) [7].

Firstly, it describes the experimental characterization of the architecture using microbenchmarks and introduces PrIM (processing-in-memory benchmarks), a benchmark suite consisting of 16 memory-bound workloads from various application domains. The analysis provides insights into the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture. It compares the architecture's performance and energy consumption to CPU and GPU counterparts. The evaluation is conducted on real UPMEM-based PIM systems with different numbers of DPUs.

Another study [10] discusses the development of a practical processing-in-memory (PIM) architecture using commercial DRAM technology. The proposed PIM architecture leverages 2.5D/3D stacking integration technologies and exploits bank-level parallelism in commodity DRAM to provide higher bandwidth and lower energy per bit transfer to processors. Importantly, the architecture does not require changes in host processors or application code, making it easily integrable with existing systems. The PIM architecture is implemented with a 20 nm DRAM technology and integrated with an unmodified commercial processor. A software stack is also developed to enable the execution of existing applications without modifications. System-level evaluations demonstrated significant performance improvements for memory-bound neural network kernels and applications, with speedups of 11.2× and 3.5×, respectively. Additionally, the proposed PIM architecture reduced the energy per bit transfer by 3.5× and improved the overall energy efficiency of the system by 3.2×.

The ever-increasing demand for high-performance machine learning applications has spurred a quest for more efficient and powerful processors. One of the key challenges in this domain lies in optimizing the data flow between memory and computing units within conventional architectures, which often leads to significant energy consumption and latency

issues. Addressing this challenge, Ref. [12] presents an innovative architecture called Lattice, which leverages nonvolatile processing-in-memory (NVPIM) based on resistive random-access memory (ReRAM) to accelerate deep convolution neural networks (DCNN). The primary objective of Lattice is to overcome the drawbacks associated with costly analog–digital conversions and excessive data copies or writes. To achieve this, the architecture introduces a novel approach to compute the partial sum of dot products between feature maps and weights in a CMOS peripheral circuit, effectively eliminating the need for analog–digital conversions. By doing so, Lattice not only reduces the energy overhead associated with these conversions but also enhances the overall system energy efficiency. Furthermore, Lattice incorporates an efficient data mapping scheme that aligns the feature map and weight data, minimizing unnecessary data copies or writes. This optimization helps to further reduce energy consumption and improve the overall performance of the system. In addition, the architecture introduces a zero-flag encoding scheme, specifically designed for sparse DCNNs, which enables energy savings during the processing of zero-values. To validate the effectiveness of the proposed architecture, extensive experiments were conducted, comparing Lattice to three state-of-the-art NVPIM designs: ISAAC, PipeLayer, and FloatPIM. The results clearly demonstrate that Lattice outperforms these existing designs, achieving substantial energy efficiency improvements ranging from 4× to 13.22×. The significance of Lattice extends beyond its immediate contributions. It sheds light on the pressing need for ultra-low power machine learning processors, especially in the era of resource-constrained edge devices and internet of things (IoT) applications. By addressing the challenges associated with data traffic between memory and computing units, Lattice paves the way for more energy-efficient and high-performance machine learning systems.

PIM-STM, a library developed by [18], offers a range of transactional memory (TM) implementations specifically designed for PIM systems. The library addresses the difficulties involved in efficiently implementing TM within PIM devices and assesses various design choices and algorithms. Additionally, it showcases experimental findings that highlight the performance and memory efficiency advantages attained by utilizing PIM-STM, as opposed to conventional CPU-based systems. The primary objective of this research is to furnish developers with valuable guidelines while also providing them with a library to experiment with alternative STM designs for PIM architectures.

In the study discussed in [19], a novel architecture named "Reconfigurable Processing-in-Memory" (PIM) is presented as a solution for data-intensive applications, specifically addressing the challenges posed by deep neural networks (DNNs) and convolutional neural networks (CNNs). These challenges primarily revolve around resource limitations and the overhead associated with data movement. While existing PIM architectures involve trade-offs in power, performance, area, energy efficiency, and programmability, the proposed architecture aims to achieve higher energy efficiency while maintaining programmability and flexibility. The proposed solution introduces a unique multi-core reconfigurable architecture integrated within DRAM sub-arrays. Each core comprises multiple processing elements (PEs) equipped with programmable functional units constructed using high-speed reconfigurable multi-functional look-up tables (M-LUTs). These M-LUTs enable the generation of multiple functional outputs in a time-multiplexed manner, eliminating the need for separate LUTs for each function. This architecture supports a wide range of operations necessary for CNN and DNN processing, including convolution, pooling, activation functions, and batch normalization. It offers enhanced efficiency and performance compared to conventional PIM architectures, making it particularly suitable for demanding applications involving big data and AI acceleration. Overall, the proposed reconfigurable PIM architecture aims to provide energy-efficient and high-performance solutions for data-intensive applications. It achieves this by leveraging the capabilities of multi-functional look-up tables and integrating them within DRAM sub-arrays.

In [20], a novel architecture named StreamPIM is introduced to tackle the memory wall problem and enhance the performance and energy efficiency of large-scale applications. The proposed architecture takes advantage of racetrack memory (RM) techniques, which in-

crease memory density and enable processing-in-memory (PIM) architectures. StreamPIM tightly integrates the memory core with computation units, creating a matrix processor using domain-wall nanowires instead of CMOS-based computation units. Additionally, it introduces a domain-wall nanowire-based bus to eliminate the need for electromagnetic conversion. By leveraging the internal parallelism of RM, the architecture optimizes performance. The StreamPIM architecture effectively addresses issues related to data transfer overheads and conversion inefficiencies, resulting in improved performance and energy efficiency for matrix computations. It provides a promising solution to overcome the limitations imposed by the memory wall, enabling more efficient and powerful processing in large-scale applications. In order to compare and analyze different architectures used in processing-in-memory (PIM) systems, a table has been compiled (Table 1) outlining the significant features of various PIM architectures. The table provides a comprehensive overview of the key characteristics and functionalities of each architecture, facilitating a better understanding of their respective advantages and limitations.

Table 1. Significant features of various PIM architectures.

Paper	Approach/Architecture	Description	Key Features	Advantages	Challenges
[1]	Hardware Design with 3D Stacked Memory	Integration of fixed-function arithmetic units and programmable cores on a 3D die-stacked memory	Minimizes data movement, improves system performance, programming model and runtime system for offloading and scheduling	- Reduced data movement between processor and memory - Improved system performance - Enables efficient offloading and scheduling	- Complex hardware design and integration - Programming model and runtime system development
[7]	UPMEM PIM Architecture	DRAM memory arrays combined with in-order cores (DRAM processing units—DPUs) on the same chip	Improves performance and energy efficiency in memory-bound workloads, benchmarking against CPU and GPU counterparts	- Enhanced performance in memory-bound workloads - Improved energy efficiency - Direct integration of processing units in memory	- Limited scalability for certain workloads - Programming and software support for DPUs
[10]	Practical PIM Architecture with Commodity DRAM	Exploits bank-level parallelism in commercial DRAM and 2.5D/3D stacking integration technologies	Higher bandwidth, lower energy per bit transfer, no changes to host processors or application code	- Increased memory bandwidth - Reduced energy consumption per bit transfer - Seamless integration with existing systems	- Overcoming stacking and integration challenges - Ensuring compatibility with diverse memory systems
[12]	Lattice Architecture with NVPIM	Utilizes nonvolatile processing-in-memory (NVPIM) based on resistive random-access memory (ReRAM) for accelerating DCNNs	Eliminates analog–digital conversions, reduces data copies/writes, improved energy efficiency and performance	- Eliminates costly analog–digital conversions - Reduced data copies and writes - Improved energy efficiency and performance	- Integration and compatibility with existing systems - Achieving high-density ReRAM arrays

Table 1. *Cont.*

Paper	Approach/Architecture	Description	Key Features	Advantages	Challenges
[18]	PIM-STM Library	Library providing various implementations of transactional memory (TM) for PIM systems	Efficient TM implementation in PIM devices, evaluation of different design choices and algorithms	- Efficient implementation of transactional memory (TM) in PIM devices - Provides guidelines and alternative design choices for TM in PIM architectures	- Ensuring TM consistency and correctness - Overhead of TM implementations on PIM systems
[19]	Reconfigurable PIM Architecture	PIM architecture integrated within DRAM sub-arrays, leveraging multi-functional look-up-tables	Higher energy efficiency, programmability, and flexibility for CNN and DNN processing	- Increased energy efficiency - Programmability and flexibility for CNN and DNN processing - Utilizes multi-functional look-up-tables for operations	- Designing efficient and scalable lookup-table-based architectures - Memory access and data dependencies
[20]	StreamPIM Architecture	Utilizes racetrack memory (RM) techniques and domain-wall nanowires to address memory wall issue	Improved performance and energy efficiency in large-scale applications, tight coupling of memory core and computation units	- Addresses memory wall issue in large-scale applications - Improved performance and energy efficiency - Tight coupling of memory core and computation units	- Overcoming challenges in RM fabrication and integration - Ensuring reliable and efficient data movement in RM

3.2. Dataflow Aware Architecture

As deep convolutional neural networks (DNNs) become more complex, the need for a manycore architecture with multiple ReRAM-based processing elements (PEs) on a single chip arises. However, traditional PIM-based architectures often prioritize computation and overlook the crucial role of communication. Merely increasing computational resources without addressing the communication infrastructure's limitations can hamper overall performance. The use of chiplet-based 2.5D architectures has gained attention in recent years. These architectures involve the integration of multiple smaller dies through a network-on-interposer (NoI) [2]. The motivation behind this approach has been to achieve energy efficiency and cost advantages compared to monolithic planar chips. Additionally, the exploration of 3D integration techniques, such as through-silicon vias (TSVs) or monolithic inter-tier vias (MIVs), offers opportunities for improved performance and energy efficiency. In the context of machine learning workloads, it is crucial to consider the specific traffic patterns and data exchange requirements. Real-world scenarios often involve the simultaneous execution of multiple machine learning applications with varying inputs. To address this, dataflow-awareness becomes essential in manycore accelerators designed for machine learning applications. Different machine learning workloads, such as convolutional neural networks (CNNs), graph neural networks (GNNs), and transformer models, exhibit unique on-chip traffic patterns when mapped onto a manycore system. Optimizing the dataflow between chip-lets or processing elements (PEs) is critical for reducing latency and improving energy efficiency. One approach involves mapping consecutive neural layers onto neighboring chip-lets or PEs to minimize long-range and multi-hop data exchange

as stated in [2]. Figure 4 provides a visual representation of the SWAP architecture and illustrates the arrangement of chiplets within the system. This architecture incorporates both mapped (M) and unmapped (NM) chiplets, as shown in the diagram. The inclusion of a limited number of mapped and unmapped chiplets enables the system to optimize its performance.

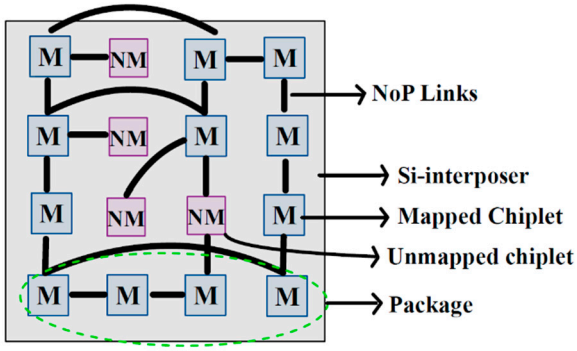

Figure 4. SWAP architecture for a chiplet-based system with a few mapped (M) and unmapped (NM) chiplets [2].

To cater to machine learning workloads, the design of a dataflow-aware network-on-interposer (NoI) architecture suited for 2.5D/3D integration is important. However, several challenges arise when communicating between chiplets, including dealing with large physical distances, mitigating issues with poor electrical wires, and managing power constraints. Achieving ultra-high bandwidth, energy-efficient, and low-latency inter-chiplet data transfer is a significant consideration. Furthermore, thermal challenges need to be addressed when designing dataflow-aware manycore architectures.

In [11], the advantages and challenges of utilizing resistive random-access memory (ReRAM)-based processing-in-memory (PIM) architectures for deep learning applications are discussed. ReRAM-based architectures have demonstrated potential in accelerating deep learning algorithms while achieving higher energy efficiency compared to traditional GPUs. However, they also present certain limitations in terms of model accuracy and performance. The document highlights the design challenges specific to ReRAM-based PIM architectures for convolutional neural networks (CNNs) and graph neural networks (GNNs). These challenges include the precision sensitivity of CNNs and the communication-intensive nature of GNNs. Moreover, the authors address the non-idealities of ReRAMs, such as noise, hard faults, process variations, and limited write endurance, which can impact the implementation of large-scale deep learning algorithms. To overcome these challenges and shortcomings, the authors propose ReRAM-based heterogeneous manycore PIM designs as a potential solution.

In [14], the focus is on implementing processing-in-memory (PIM) technology to accelerate deep learning (DL) workloads. To overcome the increasing fabrication costs associated with monolithic PIM accelerators, the paper proposes a 2.5-D system that integrates multiple PIM chiplets using a network-on-package (NoP) approach. However, existing NoP architectures often overlook the communication requirements of DL workloads. To address this issue, the SWAP architecture is introduced, which takes into account the traffic characteristics of DL applications. The SWAP architecture exhibits significant improvements in performance and energy consumption while also reducing fabrication costs compared to state-of-the-art NoP topologies. The paper presents an optimization methodology for designing an irregular NoP architecture specifically tailored to DL workloads and provides experimental evaluations that demonstrate the superior performance of the SWAP architecture.

In [15], the challenges of on-chip training for large-scale deep neural networks (DNNs) are addressed, and a mixed-precision RRAM-based compute-in-memory (CIM) architecture

called MINT is proposed. MINT leverages analog computation within the memory array to accelerate vector-matrix multiplications (VMM) and tackles issues related to weight precision and ADC resolution. By splitting weights into MSBs and LSBs, MINT employs CIM transposable arrays for forward and backward propagations of MSBs, while regular memory arrays store LSBs for weight updates. The impact of ADC resolution on training accuracy is analyzed, and experimental evaluations on a convolutional VGG-like network using the CIFAR-10 dataset demonstrate the superior accuracy and energy efficiency of the MINT architecture compared to baseline CIM architectures. Overall, MINT offers a promising solution to the challenges of on-chip training in large-scale DNNs, showcasing improved accuracy and energy efficiency.

To minimize execution time, energy consumption, and overall cost, Ref. [16] highlights the importance of hardware-mapping co-optimization in multi-accelerator systems and the need for exploring the multi-objective space. It introduces MOHaM, a framework for multi-objective hardware-mapping co-optimization. MOHaM addresses these requirements and provides an open-source infrastructure for designing multi-accelerator systems with known workloads. MOHaM utilizes a specialized multi-objective evolutionary algorithm to select suitable sub-accelerators, configure them, determine their optimal placement, and map the layers of DNNs spatially and temporally. The framework is evaluated against existing design space exploration (DSE) frameworks and demonstrates Pareto optimal solutions with significant improvements in latency and energy reduction. It introduces custom genetic operators and an optimization algorithm, making it faster and more efficient than exhaustive search methods. The results show substantial latency and energy reductions compared to state-of-the-art approaches.

3.3. Thermally Aware Architecture

The increased integration density and higher power dissipation in dataflow-aware architectures require efficient thermal management techniques to ensure reliable operation and prevent overheating. The design of dataflow-aware manycore architectures must therefore tackle thermal challenges.

One such study in [3] introduces a thermally optimized dataflow-aware monolithic 3D (M3D) network-on-chip (NoC) architecture for enhancing convolutional neural network (CNN) inferencing. The proposed design aims to integrate multiple processing-in-memory (PIM) cores using resistive random-access memory (ReRAM) technology on a single chip. Figure 5 illustrates the proposed thermally optimized dataflow-aware monolithic 3D (M3D) network-on-chip (NoC) architecture for enhancing convolutional neural network (CNN) inferencing. The architecture is designed to address the challenges of improving the efficiency of CNN inferencing by integrating multiple processing-in-memory (PIM) cores using resistive random-access memory (ReRAM) technology on a single chip. The key component in the figure is the PIM cores, represented as rectangular blocks. These cores are responsible for performing computations directly in the memory subsystem, leveraging the benefits of PIM. The integration of PIM cores with ReRAM technology enables efficient and high-performance processing of neural network operations. It emphasizes the importance of efficient communication in ReRAM-based architectures and underscores the need for an effective network-on-chip (NoC) solution. It focuses on the concept of mapping CNN layers to ReRAM-based PEs and the significance of maintaining contiguity among PEs to minimize communication latency. It discusses the use of space-filling curves (SFCs) to achieve dataflow-awareness in designing the NoC architecture. More importantly, it addresses the thermal constraints of ReRAMs, particularly the impact of temperature on conductance and inference accuracy. It emphasizes the importance of avoiding thermal hotspots and distributing high-power consuming cores effectively in the 3D architecture.

In [2], Floret is mentioned as an SFC-enabled network-on-interposer (NoI) topology for 2.5D chiplet-based integration, which achieves high performance by mapping neural layers of CNN models to contiguous chip-lets. It is stated that Floret outperforms other existing NoI architectures. However, Ref. [3] introduces TEFLON, which is described as

a thermally efficient dataflow-aware monolithic 3D (M3D) NoC architecture designed to accelerate CNN inferencing without creating thermal bottlenecks. TEFLON is claimed to reduce the energy-delay-product (EDP) and improve inference accuracy compared to performance-optimized SFC-based counterparts.

It is also observed in this study that CNNs like GN and RN34* exhibit higher reduction in energy-delay-product (EDP) compared to linear VGG CNNs such as VGG11, VGG19, VGG19*, and VGG16*. This is attributed to the presence of additional bypass links for the CNN neural layers that are spatially split among multiple processing elements (PEs) in GN and RN34*. These additional bypass links contribute to improved efficiency and reduced energy consumption in the inference process, resulting in higher EDP reduction compared to the linear VGG CNNs.

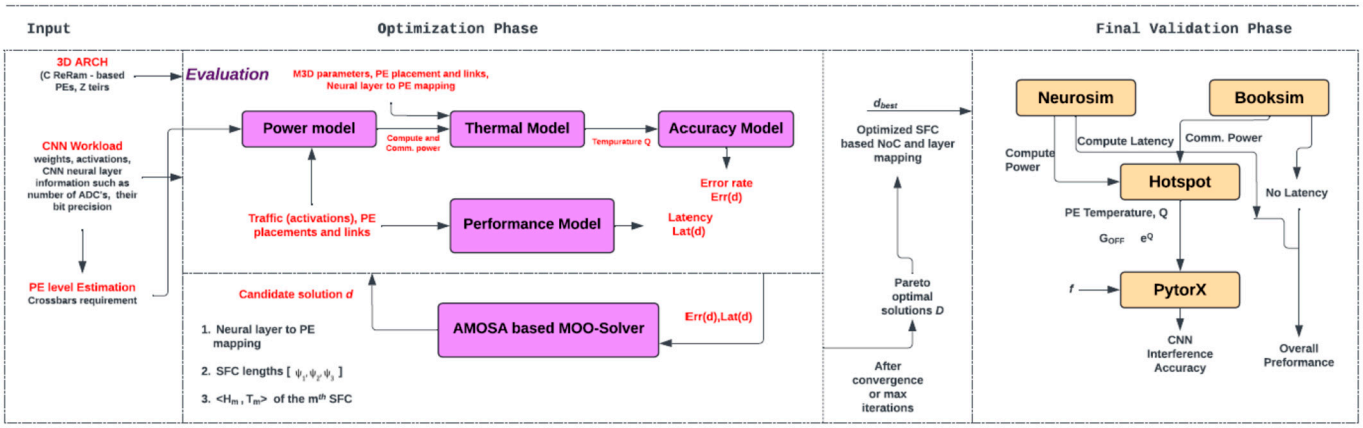

Figure 5. The TEFLON architecture described in [3].

(The asterisk (*) next to the CNN models (GN, RN34, VGG19, VGG19, VGG16) indicates that there might be some variations or modifications to the original models.)

A comparison of inference accuracy on the CIFAR-10 dataset is made between different implementations:

(a) Software-only implementation without any impact of thermal noise.

(b) Floret on a 100 PE system size, considering the impact of reduced noise margin and thermal noise, with varying PE frequency (10 MHz and 100 MHz).

It indicates that the impact of thermal noise on the inference accuracy at 100 MHz is significant on Floret for all the CNNs. For instance, the inference accuracy of the RN34 model in the Floret-enabled NoC drops by 13.4% compared to the software-only implementation. On the other hand, TEFLON-enabled NoC shows more resilience to thermal noise even at high frequencies, with an average accuracy loss ranging from 0.5% to 2% only.

Another study in [4] also discusses a design methodology for a heterogeneous 3D NoC that handles the communication requirements between CPUs and GPUs efficiently while reducing thermal issues caused by high power density. It highlights the challenges of training CNNs on heterogeneous manycore platforms and emphasizes the benefits of using 3D ICs and NoCs in improving performance and reducing data transfer latency. It discusses the need to optimize both performance and thermal characteristics in manycore systems and explores the role of CPU, GPU, and memory controller placement in achieving better performance and temperature profiles. The authors present their proposed design methodology and evaluate its effectiveness in reducing temperature while maintaining performance. They conduct experiments using LeNet and CIFAR CNNs and demonstrate a significant reduction in maximum temperature with only a minimal degradation in the full-system energy-delay-product compared to traditional 3D NoCs optimized solely for performance.

To gain insights into various PIM architectures, challenges, and proposed solutions, refer to Table 2.

Table 2. Overview of PIM architectures, challenges, and proposed solutions.

Paper	Architecture	Challenges	Proposed Solutions
[2]	Chiplet-based 2.5D architectures	Communication limitations, energy efficiency, cost advantages	Integration of multiple smaller dies through a network-on-interposer (NoI)
[3]	Thermally optimized dataflow-aware monolithic 3D (M3D) NoC architecture	Efficient communication, thermal challenges of ReRAMs	Space-filling curves (SFCs) for dataflow-awareness, avoiding thermal hotspots, distributing high-power consuming cores
[11]	ReRAM-based processing-in-memory (PIM) architectures	Model accuracy, performance, noise, hard faults, process variations, limited write endurance	ReRAM-based heterogeneous manycore PIM designs
[14]	Network-on-package (NoP) architecture for DL workloads	Communication requirements, fabrication costs	SWAP architecture based on DL traffic characteristics
[15]	Mixed-precision RRAM-based compute-in-memory (CIM) architecture	Higher weight precision, ADC resolution	MINT architecture with analog computation inside memory array
[16]	Multi-accelerator systems, hardware-mapping co-optimization	Latency, energy consumption, cost	MOHaM framework for multi-objective hardware-mapping co-optimization

3.4. Processing-in-Memory Systems Applications

3.4.1. Graph Neural Networks

Graph neural networks (GNNs) are machine learning models used for analyzing graph-structured data. The execution of GNNs involves both compute-intensive and memory-intensive operations, with the latter being a significant bottleneck due to data movement between memory and processors. PIM systems aim to alleviate this bottleneck by integrating processors close to or inside memory arrays.

In [5], the focus is on accelerating graph neural networks (GNNs) using processing-in-memory (PIM) systems. The paper introduces PyGim, a machine learning framework specifically designed to accelerate GNNs on real PIM systems. PyGim aims to harness the power of PIM architectures to improve the performance and efficiency of GNN computations. By leveraging the capabilities of PIM, PyGim enables efficient processing of graph data directly within the memory, reducing data movement and latency. The framework provides an interface for developers to easily integrate and accelerate GNN models on PIM systems, allowing for faster and more efficient graph-based computations. It proposes intelligent parallelization techniques for memory-intensive GNN kernels and develops a Python API for them. The framework enables hybrid execution of GNNs, where compute-intensive and memory-intensive operations are executed on processor-centric and memory-centric systems, respectively. Figure 6 depicts the execution of the aggregation step in a real processing-in-memory (PIM) system, as presented in the study referenced as [5]. This figure provides a visual representation of the practical implementation of the aggregation process within the context of a PIM system. PyGim is extensively evaluated on a real-world PIM system, outperforming its CPU counterpart and achieving higher resource utilization than CPU and GPU systems. It emphasizes the potential of PIM architectures in accelerating GNNs and presents several key innovations. These include the combination of accelerators (CoA) scheme, which utilizes different accelerators for compute-intensive and memory-intensive operations, and hybrid parallelism (HP) techniques for efficient parallelization of GNN aggregation on PIM systems. A PIM backend is developed, integrated with PyTorch, and made available through a user-friendly Python API. The evaluation of PyGim on a commercial PIM system demonstrates its superior performance compared to CPU-based approaches. PyGim is intended to be open-sourced to facilitate the widespread use of PIM systems in GNN applications.

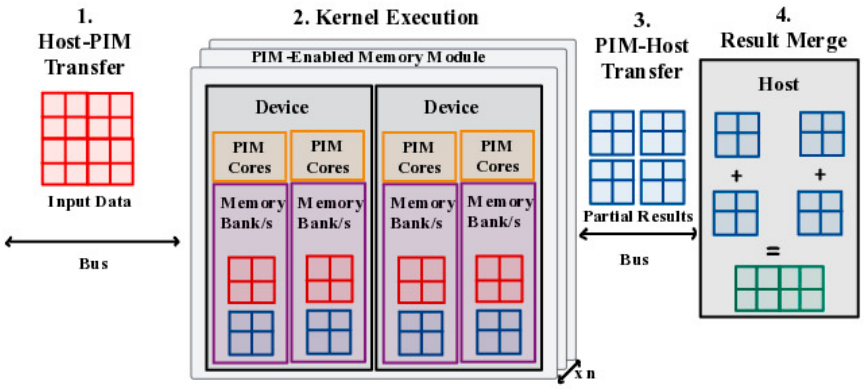

Figure 6. Executing the aggregation step on an actual PIM system [5].

There is another study in [8] that discusses the challenges of training graph neural networks (GNNs) on large real-world graph datasets in edge-computing scenarios. It also proposes the use of resistive random-access memory (ReRAM)-based processing-in-memory (PIM) architectures, which offer energy efficiency and low latency. However, ReRAM-based PIM architectures face issues of low reliability and performance when used for GNN training with large graphs. To overcome these challenges, it introduces a learning-for-data-pruning framework. This framework utilizes a trained binary graph classifier (BGC) to prune subgraphs early in the training process, reducing the size of the input data graph. By reducing redundant information, the overall training process is accelerated, the reliability of the ReRAM-based PIM accelerator is improved, and the training cost is reduced. Experimental results demonstrate that, using this data pruning framework, GNN training can be accelerated, the reliability of ReRAM-based PIM architectures can be improved by up to 1.6 times, and the overall training cost can be reduced by 100 times compared to state-of-the-art data pruning techniques.

Another study in [9] proposes a fault-aware framework for training graph neural networks (GNNs) on edge platforms using resistive random-access memory (ReRAM)-based processing-in-memory (PIM) architecture. ReRAM-based PIM architectures have gained popularity for high-performance and energy-efficient neural network training on edge devices. They leverage the crossbar array structure of ReRAMs for efficient matrix-vector multiplication operations.

However, ReRAMs are prone to hardware faults, particularly stuck-at-faults (SAFs), which make the resistance of ReRAM cells unchangeable. These faults can lead to unreliable training and poor test accuracy. The fault-tolerant methods for neural networks, such as weight pruning and retraining, are not effective in addressing faults in ReRAM-based architectures storing both adjacency and weight matrices. Ref. [9] introduces FARe, a novel fault-tolerant framework specifically designed for ReRAM-based PIM architectures. FARe considers the distribution of SAFs in ReRAM crossbars and maps the graph adjacency matrix accordingly. It also utilizes weight clipping to address faults in the GNN weight matrix. Experimental results demonstrate that FARe outperforms existing approaches in terms of both accuracy and timing overhead. It can restore GNN test accuracy by 47.6% on faulty ReRAM hardware with only a ~1% timing overhead compared to the fault-free counterpart. FARe is model- and dataset-agnostic, making it applicable to different types of GNN workloads and graph datasets.

Graph processing is important for various applications such as social networks, recommendation systems, and knowledge graphs. Traditional architectures face difficulties in handling the irregular data structure of graphs and memory-bound graph algorithms. The authors of [13] discuss the challenges and solutions related to processing large-scale graphs using processing-in-memory (PIM) architectures. They propose a degree-aware graph partitioning algorithm called GraphB for balanced partitioning and introduce tile buffers with an on-chip 2D-Mesh for efficient inter-node data transfer. GraphB also in-

corporates datalow design for computation–communication overlap and dynamic load balancing. In performance evaluations, GraphB achieves significant speedups compared to state-of-the-art PIM-based graph processing systems.

3.4.2. NN Inference

Utilizing processing-in-memory (PIM) architectures offers significant potential for enhancing both the performance and energy efficiency of neural network (NN) inference. PIM architectures integrate computational capabilities directly into the memory units, enabling computations to be performed in close proximity to the data. This proximity minimizes data movement and communication overhead, which are typically the major bottlenecks in traditional computing systems. A similar study in [6] analyzes three state-of-the-art PIM architectures: UPMEM, Mensa, and SIMDRAM. The analysis reveals that PIM architectures significantly benefit memory-bound NNs. UPMEM shows 23 times the performance of a high-end GPU when the GPU requires memory oversubscription for a general matrix-vector multiplication kernel. Figure 7 displays the design of the Mensa-G accelerator as depicted in [6]. It provides a visual representation of the architecture and components of the accelerator. Mensa improves energy efficiency and throughput by 3.0 times and 3.1 times, respectively, compared to the Google Edge TPU for 24 Google Edge NN models. SIMDRAM outperforms a CPU/GPU by 16.7 times and 1.4 times for three binary NNs. It concludes that the ideal PIM architecture for NN models depends on the specific attributes of the model, considering the inherent design choices. It emphasizes the need for programming models and frameworks that can unify the benefits of different PIM architectures into a single heterogeneous system. PIM is identified as a promising solution to improve the performance and energy efficiency of various NN models.

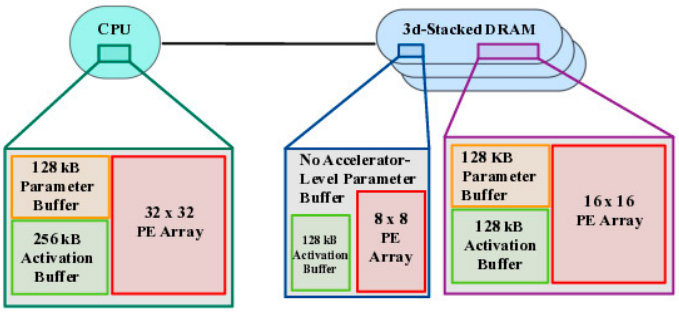

Figure 7. Mensa-G accelerator design depicted in [6].

In [17], the paper focuses on the exploration and characterization of a commercial processing-in-memory (PIM) technology called UPMEM-PIM. It highlights the need for PIM architectures to address the growing demand for memory-intensive workloads in areas such as scientific computing, graph processing, and machine learning. It mentions the challenges faced by PIM, including programmability and flexible parallelization. UPMEM-PIM is identified as a general-purpose PIM technology that offers programmability and flexibility for parallel programming. General-purpose PIM designs, like UPMEM-PIM, have the potential to become important computing devices as their hardware and software stack matures.

In [21], the focus is on accelerating reinforcement learning (RL) algorithms using processing-in-memory (PIM) systems. RL algorithms often face performance challenges due to memory-bound bottlenecks and high execution latencies when dealing with extensive and diverse datasets. To address these issues, the paper introduces SwiftRL, a framework that explores the potential of real-world PIM architectures for accelerating RL workloads and training phases. The study presents a roofline model highlighting the memory-bound nature of RL workloads, demonstrates the benefits of in-memory computing systems, conducts scalability tests on thousands of PIM cores, compares performance with traditional CPU and GPU implementations, and provides open-source PIM implementations of RL

training workloads. By showcasing the advantages and scalability of PIM architectures, the study contributes to advancing RL research and applications in memory-bound scenarios, offering new possibilities for efficient RL algorithm execution.

4. Necessity of Cyber Security in PIM

Deep neural networks (DNNs) have revolutionized various fields, including computer vision, natural language processing, and pattern recognition. However, with the increasing adoption of DNNs in critical applications, cybersecurity has emerged as a significant concern. This section explores the challenges and opportunities in enhancing cybersecurity in deep neural networks, drawing insights from recent research papers and industry practices.

The rapid advancement of artificial intelligence (AI) algorithms, such as large language models, has led to increased computing demands in data centers. This growth in AI capabilities has expanded the attack surface for cybercriminals, who exploit vulnerabilities in DNN architectures and training processes. Understanding the evolving threat landscape is crucial for developing effective cybersecurity measures.

Adversarial attacks pose a significant threat to the integrity and reliability of DNNs. These attacks involve manipulating input data through imperceptible perturbations, causing DNNs to make incorrect predictions or misclassify inputs. Defending against adversarial attacks requires robust training methodologies, such as defensive distillation and adversarial training, and the development of adversarial defense mechanisms.

Deep neural networks trained on proprietary datasets can be vulnerable to model stealing attacks. Malicious actors can extract sensitive information from deployed models, including proprietary algorithms, training data, and trade secrets. Protecting intellectual property within DNN models necessitates the implementation of secure model sharing and deployment techniques, such as watermarking and encryption.

Deep learning models often require large amounts of data for training, raising concerns regarding privacy and data leakage. Adversaries may attempt to extract sensitive information by exploiting vulnerabilities in the training process or by intercepting data during inference. Implementing privacy-preserving techniques, such as differential privacy and secure multi-party computation, can mitigate these risks and ensure the confidentiality of user data.

As deep neural networks continue to advance and find widespread adoption, addressing cybersecurity challenges becomes paramount. Enhancing cybersecurity in DNNs requires a multi-faceted approach, encompassing robust training methodologies, adversarial defense mechanisms, secure model sharing, privacy preservation, continuous monitoring and patching, and explainable AI techniques. By proactively addressing these challenges and leveraging the opportunities presented in recent research papers and industry practices, the potential of deep neural networks can be harnessed while mitigating the risks associated with cyber threats.

In [22], the paper discusses the use of heterogeneous chiplets as a solution for enabling large-scale computing in data centers, driven by the increasing demands of artificial intelligence (AI) algorithms, particularly large language models. It emphasizes the advantages of heterogeneous computing with domain-specific architectures (DSAs) and chiplets in scaling up and scaling out computing systems while reducing design complexity and costs compared to traditional monolithic chip designs. The key challenges of interconnecting heterogeneous chiplets, addressing diverse AI workloads, and ensuring chiplet interface standards, packaging, security, and software programming are explored. The paper also highlights infrastructure challenges related to communication and computation in AI task acceleration and introduces metrics to characterize different AI algorithms. Chiplets are presented as a solution for rapid development, offering performance, energy efficiency, cost, and time-to-market advantages. The success of chiplet technology is exemplified through the AMD EPYC CPU processor and other chiplet-based products. Overall, the paper provides insights into the opportunities and benefits of heterogeneous chiplets for

large-scale computing in AI workloads, addressing challenges in system integration and driving advancements in data center computing.

In [23], the paper addresses the significance of social network security and introduces the application of deep convolutional neural networks (DCNN) for topic mining and security analysis. It addresses the increasing concerns regarding network information security in social networks, such as network attacks, data leakage, and theft of confidential information. The research aims to develop a Weibo security topic detection model using DCNN and big data technology. The model utilizes the long short-term memory (LSTM) structure in the memory intelligence algorithm to extract Weibo topic information, while the DCNN learns the grammar and semantic information of Weibo topics for in-depth data features. Comparative analysis of the improved DCNN model with other models, such as AlexNet, convolutional neural networks (CNN), and deep neural networks (DNN), shows superior accuracy, recall, and F1 values. The experimental results demonstrate that the improved DCNN model achieves a recognition accuracy peak of 96.17% after 120 iterations, outperforming the other models by at least 5.4%. The intrusion detection model also exhibits high accuracy, recall, and F1 values. Furthermore, the improved DCNN security detection model shows lower training and testing time consumption compared to similar approaches in the literature. The research concludes that the improved DCNN model, based on deep learning, exhibits lower delay and good network data security transmission. Overall, the paper emphasizes the significance of timely and effective social network security topic mining and analysis models for ensuring data and information security in social networks. The utilization of DCNN and big data technology in this context provides valuable insights for enhancing network security performance and improving the security and transmission of social network data.

In [24], the paper explores the application of deep learning techniques in the field of cybersecurity. It highlights the challenges faced by computer systems in terms of security and explores how advancements in machine learning, particularly deep learning, can address these challenges. The paper presents three distinct cybersecurity problems: spam filtering, malware detection, and adult content filtering. It describes the use of specific deep learning techniques such as long short-term memory (LSTMs), deep neural networks (DNNs), and convolutional neural networks (CNNs) combined with transfer learning to tackle these problems. The experiments conducted show promising results, with an area under ROC curve greater than 0.94 in each scenario, indicating excellent performance. The paper emphasizes the importance of creating future-proof cybersecurity systems in the face of the evolving threat landscape, particularly with the rise of the internet of things (IoT). It discusses the potential of deep learning techniques to enhance the effectiveness of security solutions by leveraging artificial intelligence and machine learning advancements. In the related works section, the paper reviews previous research on malicious software detection, spam filtering, adult content filtering, and neural network architecture. It highlights the use of neural networks, including convolutional neural networks, in detecting and classifying malware. Various machine learning algorithms such as decision trees, logistic regression, random forests, AdaBoost, artificial neural networks, and convolutional neural networks are discussed in the context of spam detection. Overall, the document provides a comprehensive overview of applying deep learning in cybersecurity, evaluates the status of experiments conducted in spam filtering, malware detection, and adult content filtering, and discusses their simplicity and applicability in real-world environments. It aims to inspire more individuals to explore and utilize the potential of deep learning techniques in addressing cybersecurity challenges.

In [25], the paper aims to detect and protect cloud systems from malicious attacks by introducing a new deep learning model. The research recognizes the increasing importance of safeguarding cloud systems against various forms of cyber threats. To address this, the paper proposes a novel deep learning approach specifically designed for detecting and mitigating malicious attacks in cloud environments. The proposed model utilizes transfer learning and deep neural networks for intelligent detection of attacks in network

traffic. It converts the network traffic into 2D preprocessed feature maps, which are then processed using transferred and fine-tuned convolutional layers. The model achieves high classification accuracies, with 89.74% for multiclass and 92.58% for binary classification, as evaluated on the NSL-KDD test dataset. The paper also provides an overview of various state-of-the-art studies and techniques in the field of intrusion detection systems (IDS) using deep learning. These include models based on CNN, LSTM, autoencoders, and other deep learning architectures. Different datasets such as NSL-KDD, KDD Cup'99, and UNSW-NB15 have been utilized for training and evaluating the performance of these models. In addition, the paper mentions the use of techniques like data preprocessing, reinforcement learning, information gain (IG) filter-based feature selection, and swarm-based optimization to enhance the performance of IDS systems. It also discusses the effectiveness of deep learning approaches in improving the accuracy and efficiency of intrusion detection. Overall, the research article highlights the significance of deep transfer learning in addressing the challenges of cyber security, particularly in cloud systems. The proposed model demonstrates promising results in detecting and classifying various types of attacks, contributing to the advancement of cyber security technologies.

In [26], the paper addresses the challenges associated with improving the efficiency of malware detection using machine learning techniques and proposes potential solutions. The authors address the increasing security threats posed by malware in embedded systems and the need for robust detection methods. The paper introduces the concept of processing-in-memory (PIM) architecture, where the memory chip is enhanced with computing capabilities. This architecture minimizes memory access latency and reduces the computational resources required for model updates. The authors propose a PIM-based approach for malware detection, incorporating precision scaling techniques tailored for convolutional neural network (CNN) models.

The proposed PIM architecture (as shown in Figure 8) demonstrates higher throughput and improved energy efficiency compared to existing lookup table (LUT)-based PIM architectures. The combination of PIM and precision scaling enhances the performance of malware detection models while reducing energy consumption. This approach offers a promising solution to the resource-intensive nature of malware detection model updates and contributes to more efficient and sustainable cybersecurity practices. The paper highlights the three-fold contributions of the research: memory-efficient malware detection using in-memory computation, precision scaling to decrease power consumption, and scaling malware samples to lower bit integer types while maintaining high detection accuracy. The related work section discusses various malware detection techniques, including static and dynamic analysis, image processing, and the use of neural networks, emphasizing the advantages and limitations of each approach. It also provides an overview of processing-in-memory (PIM) designs and their benefits in terms of throughput and energy efficiency for deep learning applications. Overall, the paper presents a novel approach to improving the efficiency of malware detection through the integration of processing-in-memory architecture and precision scaling techniques. The proposed methodology shows promising results and addresses the challenges associated with training models on evolving malware data.

In [27], the paper explores the security implications associated with processing-in-memory (PIM) architectures. PIM architectures aim to enhance performance and energy efficiency by enabling direct access to main memory, but this convenience can potentially introduce vulnerabilities. The research introduces a set of high-throughput timing attacks called IMPACT, which exploit PIM architectures to establish covert and side channels. It highlights two covert-channel attack variants that leverage PIM architectures to achieve high-throughput communication channels. It also presents a side-channel attack on a DNA sequence analysis application that leaks private characteristics of a user's sample genome. The results show significant improvements in communication throughput compared to existing covert-channel attacks. It discusses the challenges and limitations of traditional defense mechanisms against PIM-based attacks and proposes potential countermeasures. It evaluates two defense mechanisms and analyzes their performance and security trade-offs.

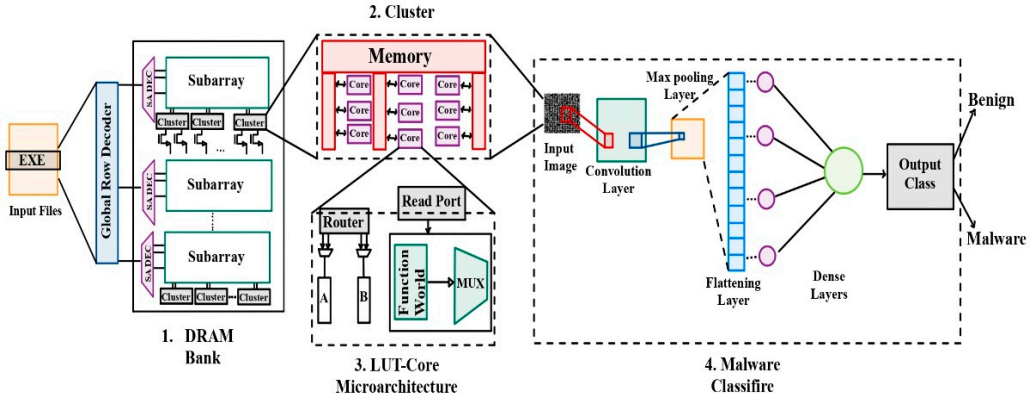

Figure 8. PIM architecture-based malware detection model depicted in [26].

Refer to Table 3 for an overview of papers addressing the importance of cybersecurity in processing-in-memory (PIM) systems. It offers insights into different approaches and perspectives taken by researchers to tackle security challenges associated with PIM technologies.

Table 3. Analysis of papers on cybersecurity in PIM.

Paper	Methodology	Advantages	Challenges
[22]	Heterogeneous chip-lets	- Scaling up and scaling out computing systems - Reduced design complexity and costs	- Chip-let interface standards - Packaging and security issues - Software programming
[23]	Deep convolutional neural networks (DCNN)	- Superior accuracy and performance - Timely and effective social network security topic mining and analysis models	--
[24]	Deep learning techniques (LSTMs, DNNs, CNNs) combined with transfer learning	- Effective application in cybersecurity - Promising experimental results	- Challenges in spam filtering, malware detection, and adult content filtering
[25]	Deep neural networks and transfer learning	- High classification accuracies - State-of-the-art techniques in intrusion detection systems using deep learning	--
[26]	Processing-in-memory (PIM) architecture	- Efficient malware detection - Higher throughput and improved energy efficiency	
[27]	Processing-in-memory (PiM) architectures and timing attacks	--	- Security implications of PiM architectures - High-throughput timing attacks exploiting PiM architectures

Overall, this section provides insights into the challenges, opportunities, and benefits associated with cybersecurity measures in various domains, including deep neural networks, large-scale computing, social media platforms, and cloud systems. It highlights the importance of robust techniques and advanced technologies in protecting against cyber threats and preserving data security.

5. Summary of the Review

This article provides a comprehensive review of the latest advancements in processing-in-memory (PIM) techniques for deep learning applications. It addresses the limitations of

traditional von Neumann architectures and highlights the benefits of chiplet-based designs and PIM in terms of scalability, modularity, flexibility, performance, and energy efficiency.

This article begins by discussing the challenges faced by monolithic chip architectures and how chiplet-based designs offer improved scalability and resource utilization. It then delves into the concept of processing-in-memory, which aims to overcome the memory bottleneck by integrating computational units directly into the memory subsystem. PIM architectures reduce data movement, minimize latency, and improve energy efficiency by performing computations in close proximity to the data. Various memory technologies, such as SRAM, DRAM, ReRAM, and PCM, can be leveraged in PIM architectures.

This review emphasizes the significance of dataflow-awareness, communication optimization, and thermal considerations in designing PIM-enabled manycore architectures. It explores different machine learning workloads and their specific dataflow requirements. The document also presents a heterogeneous PIM system for energy-efficient neural network training and discusses thermally efficient dataflow-aware monolithic 3D NoC architectures for accelerating CNN inferencing.

There are several areas of future research and development in the field of processing-in-memory architectures for deep neural networks. Some potential future directions include:

1. Exploring advanced memory technologies: further investigation into emerging memory technologies, such as memristors or spintronics, can offer new opportunities for enhancing the performance and energy efficiency of PIM architectures.
2. Optimizing communication and interconnectivity: continued research on efficient on-chip interconnection networks and communication protocols can further reduce data movement and latency in PIM architectures.
3. Integration with emerging technologies: exploring the integration of PIM architectures with other emerging technologies, such as neuromorphic computing or quantum computing, can lead to novel and more efficient computing systems.
4. Security and privacy considerations: addressing the cybersecurity challenges associated with deep neural networks and PIM architectures, including adversarial attacks, model stealing attacks, and privacy concerns, is crucial for the widespread adoption of these technologies.
5. Hardware–software co-design: further exploration of hardware–software co-design approaches can enable better optimization and utilization of PIM architectures, considering the unique characteristics of deep learning workloads.
6. Real-world application deployment: conducting practical experiments and case studies to evaluate the performance, energy efficiency, and scalability of PIM architectures in real-world deep learning applications can provide valuable insights for their adoption.

Table 4 provides a comprehensive collection of papers that delve into the topic of processing-in-memory (PIM). It encompasses discussions on architecture, challenges, proposed solutions, and future scope, as explored in this review.

Table 4. Compilation of papers on PIM: architecture, challenges, proposed solutions, and future scope.

Paper	Architecture	Challenges	Proposed Solutions	Future Scope
[2]	Chiplet-based 2.5D architectures	Communication limitations, energy efficiency, cost advantages	Integration of multiple smaller dies through a network-on-interposer (NoI)	Exploring advanced interconnect technologies, optimizing power efficiency further
[3]	Thermally optimized dataflow-aware monolithic 3D (M3D) NoC architecture	Efficient communication, thermal challenges of ReRAMs	Space-filling curves (SFCs) for dataflow-awareness, avoiding thermal hotspots, distributing high-power consuming cores	Investigating advanced thermal management techniques, extending to new memory technologies

Table 4. *Cont.*

Paper	Architecture	Challenges	Proposed Solutions	Future Scope
[11]	ReRAM-based processing-in-memory (PIM) architectures	Model accuracy, performance, noise, hard faults, process variations, limited write endurance	ReRAM-based heterogeneous manycore PIM designs	Enhancing error tolerance, exploring novel training algorithms for PIM architectures
[14]	Network-on-package (NoP) architecture for DL workloads	Communication requirements, fabrication costs	SWAP architecture based on DL traffic characteristics	Exploring advanced packaging technologies, optimizing for heterogeneous workloads
[15]	Mixed-precision RRAM-based compute-in-memory (CIM) architecture	Higher weight precision, ADC resolution	MINT architecture with analog computation inside memory array	Investigating novel analog computing schemes, optimizing for large-scale deployment
[16]	Multi-accelerator systems, hardware-mapping co-optimization	Latency, energy consumption, cost	MOHaM framework for multi-objective hardware-mapping co-optimization	Exploring dynamic workload allocation, optimizing for emerging DL algorithms
[18]	TEFLON: A Design Space Exploration Framework for Hardware Accelerators	Design space exploration, accelerator architectures	TEFLON framework for exploring accelerator designs with customizable datapath and memory hierarchy	Enhancing design exploration capabilities, incorporating new architectural innovations
[21]	Deep Learning Accelerators: A Comprehensive Survey	Deep learning accelerator architectures, performance, energy efficiency	Survey of various deep learning accelerator architectures and their characteristics	Investigating hardware–software co-design, exploring heterogeneous computing platforms
[23]	Efficient Processing of Deep Learning Models: A Tutorial and Survey	Deep learning model compression, quantization, hardware-friendly optimization	Tutorial and survey on various techniques for efficient processing of deep learning models	Exploring federated learning approaches, optimizing for edge and IoT devices
[27]	Hardware Architectures for Deep Learning: A Survey	Hardware architectures for deep learning, accelerators, memory systems	Comprehensive survey on hardware architectures for deep learning, including accelerators and memory systems	Investigating neuromorphic computing, exploring advanced memory technologies

6. Conclusions

In conclusion, this comprehensive review has explored the advancements in processing-in-memory (PIM) techniques for deep learning applications. The limitations of monolithic chip designs in deep learning, such as area, yield, and on-chip interconnection costs, have been addressed, and chiplet-based architectures have emerged as a promising solution. These architectures offer improved scalability, modularity, and flexibility, allowing for better yield and reduced costs. Furthermore, chiplet-based designs enable efficient utilization of resources by distributing the computational workload across multiple chiplets, resulting in enhanced performance and energy efficiency.

Processing-in-memory (PIM) techniques have gained significant attention as they aim to overcome the memory bottleneck in deep learning. By integrating processing units directly into the memory subsystem, PIM architectures minimize data movement, reduce latency, and improve energy efficiency. This review has highlighted the potential of PIM architectures in leveraging emerging memory technologies like resistive random-access memory (ReRAM) to achieve high-performance and energy-efficient acceleration of deep learning tasks.

The importance of dataflow-awareness and communication optimization in the design of PIM-enabled manycore platforms has been emphasized. Different machine learning workloads require tailored dataflow-awareness to minimize latency and improve energy efficiency. Additionally, the challenges associated with on-chip interconnection networks,

thermal constraints, and scalable communication in chiplet-based architectures have been discussed.

A heterogeneous PIM system for energy-efficient neural network training has been presented, combining fixed-function arithmetic units and programmable cores on a 3D die-stacked memory. This approach provides a unified programming model and runtime system for efficient task offloading and scheduling. The significance of programming models that accommodate both fixed-function logics and programmable cores has been highlighted.

This review has also explored thermally efficient dataflow-aware monolithic 3D (M3D) NoC architectures for accelerating CNN inferencing. By integrating processing-in-memory cores using ReRAM technology and designing efficient network-on-chip (NoC) architectures, data movement can be reduced. The advantages of TEFLON (thermally efficient dataflow-aware 3D NoC) over performance-optimized space-filling curve (SFC)-based counterparts in terms of energy efficiency, inference accuracy, and thermal resilience have been highlighted.

Overall, the advancements in processing-in-memory techniques have the potential to revolutionize deep learning hardware. These approaches offer scalability, flexibility, improved performance, and energy efficiency, addressing the challenges faced by traditional monolithic chip designs. By leveraging the benefits of processing-in-memory techniques, researchers and engineers can pave the way for enhanced deep learning capabilities and contribute to the development of efficient and powerful AI hardware.

Author Contributions: A.A. provided conceptualization, R.K. and A.A. contributed to the methodology, F.M. provided funding for the research. The writing, review, and editing of the paper were primarily undertaken by R.K. The validation process involved the contributions of R.K., A.A. and F.M. Together, the collective efforts ensured the successful completion of this comprehensive survey paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSERC.

Conflicts of Interest: The authors declare no funding interest.

References

1. Liu, J.; Zhao, H.; Ogleari, M.A.; Li, D.; Zhao, J. Processing-in-Memory for Energy-Efficient Neural Network Training: A Heterogeneous Approach. In Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-51), Fukuoka, Japan, 20–24 October 2018; pp. 655–668. [CrossRef]
2. Sharma, H.; Narang, G.; Doppa, J.R.; Ogras, U.; Pande, P.P. Dataflow-Aware PIM-Enabled Manycore Architecture for Deep Learning Workloads. *arXiv* **2024**, arXiv:2403.19073. Available online: https://arxiv.org/abs/2403.19073 (accessed on 24 May 2024).
3. Narang, G.; Ogbogu, C.; Doppa, J.; Pande, P. TEFLON: Thermally Efficient Dataflow-Aware 3D NoC for Accelerating CNN Inferencing on Manycore PIM Architectures. *ACM Trans. Embed. Comput. Syst.* **2024**, *just accepted*. [CrossRef]
4. Joardar, B.K.; Choi, W.; Kim, R.G.; Doppa, J.R.; Pande, P.P.; Marculescu, D.; Marculescu, R. 3D NoC-Enabled Heterogeneous Manycore Architectures for Accelerating CNN Training: Performance and Thermal Trade-Offs. In Proceedings of the Eleventh IEEE/ACM International Symposium on Networks-on-Chip, Seoul, Republic of Korea, 19 October 2017; pp. 1–8.
5. Giannoula, C.; Yang, P.; Vega, I.F.; Yang, J.; Li, Y.X.; Luna, J.G.; Sadrosadati, M.; Mutlu, O.; Pekhimenko, G. Accelerating Graph Neural Networks on Real Processing-In-Memory Systems. *arXiv* **2024**, arXiv:2402.16731.
6. Oliveira, G.F.; Gómez-Luna, J.; Ghose, S.; Boroumand, A.; Mutlu, O. Accelerating Neural Network Inference with Processing-in-DRAM: From the Edge to the Cloud. *IEEE Micro* **2022**, *42*, 25–38. [CrossRef]
7. Gómez-Luna, J.; El Hajj, I.; Fernandez, I.; Giannoula, C.; Oliveira, G.F.; Mutlu, O. Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware. In Proceedings of the 2021 12th International Green and Sustainable Computing Conference (IGSC), Pullman, WA, USA, 18 October 2021; pp. 1–7.
8. Ogbogu, C.; Joardar, B.K.; Chakrabarty, K.; Doppa, J.; Pande, P.P. Data Pruning-enabled High Performance and Reliable Graph Neural Network Training on ReRAM-based Processing-in-Memory Accelerators. *ACM Trans. Des. Autom. Electron. Syst.* **2024**, *just accepted*. [CrossRef]
9. Dhingra, P.; Ogbogu, C.; Joardar, B.K.; Doppa, J.R.; Kalyanaraman, A.; Pande, P.P. FARe: Fault-Aware GNN Training on Re-RAM-based PIM Accelerators. *arXiv* **2024**, arXiv:2401.10522.

10. Lee, S.; Kang, S.H.; Lee, J.; Kim, H.; Lee, E.; Seo, S.; Yoon, H.; Lee, S.; Lim, K.; Shin, H.; et al. Hardware Architecture and Software Stack for PIM Based on Commercial DRAM Technology: Industrial Product. In Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 14–18 June 2021; pp. 43–56.
11. Joardar, B.K.; Arka, A.I.; Doppa, J.R.; Pande, P.P.; Li, H.; Chakrabarty, K. Heterogeneous Manycore Architectures Enabled by Processing-in-Memory for Deep Learning: From CNNs to GNNs (ICCAD Special Session Paper). In Proceedings of the 2021 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Munich, Germany, 1 November 2021; pp. 1–7.
12. Zheng, Q.; Wang, Z.; Feng, Z.; Yan, B.; Cai, Y.; Huang, R.; Chen, Y.; Yang, C.L.; Li, H.H. Lattice: An ADC/DAC-less ReRAM-Based Processing-in-Memory Architecture for Accelerating Deep Convolutional Neural Networks. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20 July 2020; pp. 1–6.
13. Zhao, X.; Chen, S.; Kang, Y. Load Balanced PIM-Based Graph Processing. *ACM Trans. Des. Autom. Electron. Syst.* **2024**, *just accepted*. [CrossRef]
14. Sharma, H.; Mandal, S.K.; Doppa, J.R.; Ogras, U.Y.; Pande, P.P. SWAP: A Server-Scale Communication-Aware Chiplet-Based Manycore PIM Accelerator. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4145–4156. [CrossRef]
15. Jiang, H.; Huang, S.; Peng, X.; Yu, S. MINT: Mixed-Precision RRAM-Based In-Memory Training Architecture. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12 October 2020; pp. 1–5.
16. Das, A.; Russo, E.; Palesi, M. Multi-Objective Hardware-Mapping Co-Optimisation for Multi-DNN Workloads on Chiplet-Based Accelerators. *IEEE Trans. Comput.* **2024**, *73*, 1883–1898. [CrossRef]
17. Hyun, B.; Kim, T.; Lee, D.; Rhu, M. Pathfinding Future PIM Architectures by Demystifying a Commercial PIM Technology. In Proceedings of the 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Edinburgh, UK, 2 March 2024; pp. 263–279.
18. Lopes, A.; Castro, D.; Romano, P. PIM-STM: Software Transactional Memory for Processing-In-Memory Systems. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, La Jolla, CA, USA, 27 April 2024; Volume 2, pp. 897–911.
19. Bavikadi, S.; Sutradhar, P.R.; Ganguly, A.; Dinakarrao, S.M.P. Reconfigurable Processing-in-Memory Architecture for Data Intensive Applications. In Proceedings of the 2024 37th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID), Kolkata, India, 6–10 January 2024; pp. 222–227.
20. An, Y.; Tang, Y.; Yi, S.; Peng, L.; Pan, X.; Sun, G.; Luo, Z.; Li, Q.; Zhang, J. StreamPIM: Streaming Matrix Computation in Racetrack Memory. In Proceedings of the 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Edinburgh, UK, 2–6 March 2024; pp. 297–311.
21. Gogineni, K.; Dayapule, S.S.; Gómez-Luna, J.; Gogineni, K.; Wei, P.; Lan, T.; Sadrosadati, M.; Mutlu, O.; Venkataramani, G. SwiftRL: Towards Efficient Reinforcement Learning on Real Processing-In-Memory Systems. *arXiv* **2024**, arXiv:2405.03967.
22. Yang, Z.; Ji, S.; Chen, X.; Zhuang, J.; Zhang, W.; Jani, D.; Zhou, P. Challenges and Opportunities to Enable Large-Scale Computing via Heterogeneous Chiplets. In Proceedings of the 2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC), Incheon, Republic of Korea, 22–25 January 2024; pp. 765–770.
23. Wang, C. Social Media Platform-Oriented Topic Mining and Information Security Analysis by Big Data and Deep Convolutional Neural Network. *Technol. Forecast. Soc. Chang.* **2024**, *199*, 123070. [CrossRef]
24. Miranda-García, A.; Rego, A.Z.; Pastor-López, I.; Sanz, B.; Tellaeche, A.; Gaviria, J.; Bringas, P.G. Deep Learning Applications on Cybersecurity: A Practical Approach. *Neurocomputing* **2024**, *563*, 126904. [CrossRef]
25. Çavuşoğlu, Ü.; Akgun, D.; Hizal, S. A Novel Cyber Security Model Using Deep Transfer Learning. *Arab. J. Sci. Eng.* **2024**, *49*, 3623–3632. [CrossRef]
26. Kasarapu, S.; Bavikadi, S.; Dinakarrao, S.M. Empowering Malware Detection Efficiency within Processing-in-Memory Architecture. *arXiv* **2024**, arXiv:2404.08818.
27. Kanellopoulos, K.; Bostanci, F.; Olgun, A.; Yaglikci, A.G.; Yuksel, I.E.; Ghiasi, N.M.; Bingol, Z.; Sadrosadati, M.; Mutlu, O. Amplifying Main Memory-Based Timing Covert and Side Channels using Processing-in-Memory Operations. *arXiv* **2024**, arXiv:2404.11284.
28. Asad, A.; Kaur, R.; Mohammadi, F. A Survey on Memory Subsystems for Deep Neural Network Accelerators. *Future Internet* **2022**, *14*, 146. [CrossRef]
29. Kaur, R.; Mohammadi, F. Power Estimation and Comparison of Heterogeneous CPU-GPU Processors. In Proceedings of the 2023 IEEE 25th Electronics Packaging Technology Conference (EPTC), Singapore, 5–8 December 2023; pp. 948–951.
30. Kaur, R.; Mohammadi, F. Comparative Analysis of Power Efficiency in Heterogeneous CPU-GPU Processors. In Proceedings of the 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 24–27 July 2023; pp. 756–758.
31. Kaur, R.; Saluja, N. Comparative Analysis of 1-bit Memory Cell in CMOS and QCA Technology. In Proceedings of the 2018 International Flexible Electronics Technology Conference (IFETC), Ottawa, ON, Canada, 7–9 August 2018; pp. 1–3. [CrossRef]
32. Safayenikoo, P.; Asad, A.; Fathy, M.; Mohammadi, F. An Energy Efficient Non-Uniform Last Level Cache Architecture in 3D Chip-Multiprocessors. In Proceedings of the 2017 18th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 14–15 March 2017; pp. 373–378. [CrossRef]

33. Asad, A.; AL-Obaidy, F.; Mohammadi, F. Efficient Power Consumption using Hybrid Emerging Memory Technology for 3D CMPs. In Proceedings of the 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), San Jose, Costa Rica, 25–28 February 2020; pp. 1–4. [CrossRef]
34. Asad, A.; Kaur, R.; Mohammadi, F. Noise Suppression Using Gated Recurrent Units and Nearest Neighbor Filtering. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 368–372. [CrossRef]
35. Shin, H.; Kang, M.; Kim, L. A thermal-aware optimization framework for ReRAM-based deep neural network acceleration. In Proceedings of the ICCAD '20: IEEE/ACM International Conference on Computer-Aided Design, Virtual Event, USA, 2–5 November 2020.
36. Mutlu, O.; Ghose, S.; Gómez-Luna, J.; Ausavarungnirun, R. A modern primer on processing in memory. In *Emerging Computing: From Devices to Systems: Looking beyond Moore and Von Neumann*; Springer Nature Singapore: Singapore, 2022; pp. 171–243.
37. Yu, C.; Liu, S.; Khan, S. Multipim: A detailed and configurable multi-stack processing-in-memory simulator. *IEEE Comput. Archit. Lett.* **2021**, *20*, 54–57. [CrossRef]
38. Mosanu, S.; Sakib, M.N.; Tracy, T.; Cukurtas, E.; Ahmed, A.; Ivanov, P.; Khan, S.; Skadron, K.; Stan, M. Pimulator: A fast and flexible processing-in-memory emulation platform. In Proceedings of the 2022 Design Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 14–23 March 2022; pp. 1473–1478.
39. Roesch, J.; Lyubomirsky, S.; Weber, L.; Pollock, J.; Kirisame, M.; Chen, T.; Tatlock, Z. Relay: A new IR for machine learning frameworks. In Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, Philadelphia, PA, USA, 18 June 2018; pp. 58–68.
40. Kim, C.H.; Lee, W.J.; Paik, Y.; Kwon, K.; Kim, S.Y.; Park, I.; Kim, S.W. Silent-PIM: Realizing the processing-in-memory computing with standard memory requests. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 251–262. [CrossRef]
41. Jin, H.; Liu, C.; Liu, H.; Luo, R.; Xu, J.; Mao, F.; Liao, X. ReHy: A ReRAM-Based Digital/Analog Hybrid PIM Architecture for Accelerating CNN Training. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 2872–2884. [CrossRef]
42. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends Signal Process* **2014**, *7*, 3–4. [CrossRef]
43. Haj-Ali, A.; Ben-Hur, R.; Wald, N.; Ronen, R.; Kvatinsky, S. Not in name alone: A memristive memory processing unit for real in-memory processing. *IEEE Micro* **2018**, *38*, 13–21. [CrossRef]
44. Ben-Hur, R.; Ronen, R.; Haj-Ali, A.; Bhattacharjee, D.; Eliahu, A.; Peled, N.; Kvatinsky, S. SIMPLER MAGIC: Synthesis and mapping of in-memory logic executed in a single row to improve throughput. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 2434–2447. [CrossRef]
45. Mittal, S. A survey of ReRAM-based architectures for processing-in-memory and neural networks. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 75–114. [CrossRef]
46. Kim, D.; Na, T.; Yalamanchili, S.; Mukhopadhyay, S. DeepTrain: A Programmable Embedded Platform for Training Deep Neural Networks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *37*, 2360–2370. [CrossRef]
47. Boroumand, A.; Ghose, S.; Oliveira, G.F.; Mutlu, O. Polynesia: Enabling Effective Hybrid Transactional/Analytical Databases with Specialized Hardware/Software Co-Design. *arXiv* **2021**, arXiv:2103.00798.
48. Gu, P.; Xie, X.; Ding, Y.; Chen, G.; Zhang, W.; Niu, D.; Xie, Y. iPIM: Programmable In-Memory Image Processing Accelerator using Near-Bank Architecture. In Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 30 May–3 June 2020.
49. Huang, Y.; Zheng, L.; Yao, P.; Zhao, J.; Liao, X.; Jin, H.; Xue, J. A Heterogeneous PIM Hardware-Software Co-Design for Energy-Efficient Graph Processing. In Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 18–22 May 2020.
50. Liu, W. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]