Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung
Nr. 108

Simon Maria Friedrich

On Interactions of Deep Neural Network Acceleration and Memory Subsystem

VOGT

Dresden 2025

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Bibliographic Information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2025

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „On Interactions of Deep Neural Network Acceleration and Memory Subsystem“ von Simon Maria Friedrich überein.

Gesetzt vom Autor

ISBN 978-3-95947-085-8

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

On Interactions of Deep Neural Network Acceleration and Memory Subsystem

Dipl.-Ing.

Simon Maria Friedrich

der Fakultät Elektrotechnik und Informationstechnik der
Technischen Universität Dresden

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. Christian G. Mayr
Gutachter:	Prof. Dr.-Ing. Dr. h.c. Gerhard P. Fettweis
	Prof. Dr. Oliver Bringmann
Mitglied der Kommission:	Prof. Dr.-Ing. Rafael F. Schaefer
Tag der Einreichung:	31.03.2025
Tag der Verteidigung:	23.09.2025

Simon Maria Friedrich
On Interactions of Deep Neural Network Acceleration and Memory Subsystem
Dissertation, 23.09.2025

Technische Universität Dresden
Lehrstuhl für Mobile Nachrichtensysteme
Institut für Nachrichtentechnik
Fakultät Elektrotechnik und Informationstechnik
01062 Dresden, Germany

Abstract

Artificial intelligence has rapidly advanced over the past decade. Following breakthroughs in image classification, neural networks have been applied to a wide range of tasks, expanding their impact across various domains. The increasing adoption of Deep Neural Networks (DNNs) is driven by their improved accuracy and growing capabilities. However, this surge in model development has only been possible through significant advancements in hardware systems. Dedicated accelerators are now widely used for training and executing neural networks, extending the design space beyond traditional Central Processing Unit (CPU) and Graphics Processing Unit (GPU) clusters. These accelerators are gaining popularity due to their specialized architectures, which are optimized for neural network workloads. This evolution enables the training and execution of larger models on increasingly extensive datasets. Despite these advancements, dedicated DNN accelerators, like all compute cores, remain constrained by the *memory wall* issue. This challenge arises as computing performance continues to outpace the growth of memory bandwidth and interconnect speeds, making the memory subsystem a critical performance bottleneck. Several strategies exist to mitigate the *memory wall*. This thesis extends these approaches by analyzing the interactions between DNN acceleration and the memory subsystem. Firstly, it introduces novel contributions that leverage DNNs themselves to enhance the memory interconnect efficiency. Additionally, it presents a dedicated hardware architecture designed to enable memory-efficient Dilated Convolution (DCONV) processing.

In embedded systems, high-performance cores typically rely on fast and predictable on-chip memory. To improve conflict handling and reduce execution time, offline arbitration can be combined with memory access prediction, a technique known as Access Interval Prediction (AIP). This thesis introduces neural network-based AIP units to enhance prediction accuracy. By further leveraging model compression techniques, the system's compute cost can be reduced while maintaining performance improvements in configurations with multiple masters and shared memory.

However, memory sharing is generally not feasible for compute cores specifically designed for DNN execution. Since these systems also face limited effective memory bandwidth, we introduce a novel memory mapping and address generation scheme.

This approach eliminates redundant operations for DCONV, resulting in a net performance increase even in memory-bound systems with constrained bandwidth. Additionally, by implementing data-reuse register stages within the compute core, energy efficiency can be further improved. Our results demonstrate that the negative impact of the *memory wall* can be mitigated by aligning the compute and memory systems for specific operations such as DCONV. This optimization enhances the deployment of DCONV operations in embedded systems, making applications such as semantic segmentation feasible on mobile devices.

Kurzfassung

Die Entwicklung der künstlichen Intelligenz hat sich im letzten Jahrzehnt rasant beschleunigt. Nach herausragenden Fortschritten in der Klassifizierung von Bildern wurden neuronale Netzwerke auf eine Vielzahl von Aufgaben angewendet. Folglich kommen sie in unterschiedlichen Bereichen nun zum Einsatz. Aufgrund von Verbesserungen in der Genauigkeit und wachsenden Fähigkeiten werden vor allem zunehmend tiefe neuronale Netze (engl. *Deep Neural Networks*, DNNs) eingesetzt. Diese Entwicklung an Modellen wäre jedoch ohne bedeutende Fortschritte in den zugehörigen Hardwaresystemen nicht möglich gewesen. Neben traditionellen zentralen Recheneinheiten und Grafikprozessoren werden heute weitgehend dedizierte Beschleuniger für das Training und die Ausführung von neuronalen Netzwerken eingesetzt. Vor allem aufgrund ihrer spezialisierten Architekturen, die für die Verarbeitung von neuronalen Netzwerken optimiert sind, gewinnen diese Beschleuniger an Popularität. Diese Entwicklung ermöglicht es, größere Modelle auf zunehmend umfangreicheren Datensätzen zu trainieren and auszuführen. Trotz dieser Fortschritte wird die Leistungsfähigkeit dedizierter DNN-Beschleuniger, wie alle anderen Rechensysteme, durch das Problem der sogenannten *Speicherwand* (engl. *memory wall*) eingeschränkt. Die Herausforderung hierbei ist, dass die Rechenleistung schneller als die Bandbreite des Speichers und der Geschwindigkeit der Speicherinterkonnektivität wächst, wodurch das Speichersystem zu einem entscheidenden Engpass des Gesamtsystems wird. Es gibt verschiedene Strategien, um die *memory wall* zu überwinden. Durch die Analyse der Interaktionen zwischen DNN-Beschleunigung und dem Speichersystem werden diese Strategien innerhalb dieser Arbeit erweitert. Zunächst wird eine Methode ausgearbeitet, die DNNs selbst nutzen, um die Effizienz der Speicherinterkonnektivität zu verbessern. Zusätzlich wird eine dedizierte Hardwarearchitektur vorgestellt, die ein speichereffizientes Ausführen von gedehnten Faltungen (engl. *Dilated Convolution*, DCONV) ermöglicht.

In eingebetteten Systemen setzen Hochleistungs-Rechenkerne in der Regel auf schnelle und vorhersehbare Speichermodule, welche auf dem Chip integriert sind. Um Konflikte besser zu lösen und die Ausführungszeit zu reduzieren, kann eine offline Arbitrierung mit Speicherzugriffsvorhersage kombiniert werden, eine Technik, die als Zugriffsintervallvorhersage (engl. *Access Interval Prediction*, AIP) bekannt ist. Zur Verbesserung der Vorhersagegenauigkeit stellt diese Arbeit AIP-Einheiten auf

Basis neuronaler Netzwerke vor. Durch die Nutzung von Techniken zur Komprimierung von DNNs können die Rechenkosten des Systems gesenkt werden, während weiterhin die Verbesserung der Leistungsfähigkeit in Systemenkonfigurationen mit mehreren Mastern und gemeinsamem Speicher aufrechterhalten wird.

Das Teilen eines gemeinsamen Speichers ist jedoch in der Regel nicht praktikabel für Rechenkerne, die speziell für die Ausführung von DNNs entwickelt wurden. Da diese Systeme ebenfalls mit begrenzter effektiver Speicherbandbreite konfrontiert sind, wird ein neuartiges Speicherabbildungs- und Adressgenerierungsschema vorgestellt. Dieser Ansatz eliminiert redundante Operationen während der Berechung von DCONVs und führt zu einer effektiven Steigerung der Leistungsfähigkeit, selbst in Systemen, bei denen die Speicherbandbreite der limitierende Faktor ist. Zur weiteren Steigerung der Energieeffizienz können mit zusätzlichen Registerstufen Daten zwischen den Rechenmodulen wiederverwendet werden. Die Ergebnisse dieser Arbeit zeigen, dass die negativen Auswirkungen der *memory wall* durch die gegenseitige Abstimmung des Rechen- und Speichersystems für bestimmte Operationen wie DCONVs abgeschwächt werden können. Diese Optimierung verbessert den Einsatz von DCONV-Operationen in eingebetteten Systemen und ermöglicht Anwendungen wie die semantische Segmentierung auf mobilen Geräten.

Acknowledgement

An dieser Stelle möchte ich all den Leuten danken, die mir innerhalb der letzten vier Jahre meiner Promotion zur Seite standen. Ohne deren Unterstützung wäre meine Doktorarbeit in dieser Form nicht möglich gewesen.

Besonders hervorheben möchte ich meinen Doktorvater Gerhard Fettweis und meinen Gruppenleiter Emil Matus. Beide gaben mir über die Jahre hinweg nicht nur inhaltliche Anregungen und Ideen, sondern sorgten auch dafür, dass ich die Einreichung der Doktorarbeit nicht aus den Augen verloren habe.

Wenn ich an die Zeit am Lehrstuhl zurückdenke, muss ich natürlich auch Sylvia, Claudia, Nike, Rüdiger und Raffael für ihre Unterstützung bei all den administrativen Angelegenheiten und das Bahnen durch die teilweise bürokratischen Strukturen der Universität danken. So kam ich doch beinahe jeden Tag aufs neue wieder motiviert ins Büro, was auf jeden Fall auch dadurch katalysiert wurde, dass mir einige meiner Kollegen über all die Zeit freundschaftlich ans Herz gewachsen sind. Besonders mein Kollege Robert hatte während jeder Phase meiner Promotion immer ein offenes Ohr für mich und stand mir stets zur Seite.

Auch möchte ich all meinen Freunden danken, die mich in den Bereichen fernab der Promotion begleitet und mir geholfen haben den benötigten Ausgleich in meinem Leben zu finden. Besonders gilt dies für Franz, Michael und Daniel.

Der größte Dank gilt aber meiner Familie. Sie war es, die nicht nur über die Jahre hinweg bei meiner Promotion mitgefiebert und sich über jedes akzeptiertes Paper von Herzen gefreut hat. Sondern sie stand auch in den anspruchvollen Abschnitten meiner Doktorarbeit immer an meiner Seite und hat mich hierbei bedingungslos unterstützt.

Der allergrößte Respekt gebührt hierbei meiner Partnerin Irena. Dank ihrer Wertschätzung und aufopferungsvollen Unterstützung wusste ich auch in den stressigsten Phasen, dass ich mich immer auf sie verlassen und alles schaffen kann. Seitdem wir uns kennengelernt haben kam eine solch unglaubliche und positive Dynamik in mein Leben, die mich nicht nur beim Schreiben meiner Doktorarbeit beflügelt hat und welche ich nie mehr missen möchte.

Dresden, März 2025 Simon Maria Friedrich

Contents

1 Introduction

The development of Artificial Intelligence (AI) has significantly accelerated over the past decade. Since the notable achievements of Convolutional Neural Networks (CNNs) in image classification in 2012 [KSH12] and transformer networks in 2017 [Vas+17], Deep Neural Networks (DNNs) have seen widespread adoption across various industrial sectors. Their applications span various fields, including manufacturing, finance, and medical science [Abi+18; Sar21]. A growing number of scientific publications highlights the surge in AI development. In the medical field alone, AI-related publications doubled between 2015 and 2022 [Kar+23]. The tasks are similar across various industries and can be categorized into prediction, pattern recognition, and classification. Image processing, which incorporates classification and object detection, is essential in many sectors, such as healthcare and automotive. In autonomous driving, the role of DNNs has expanded to support more complex tasks, such as semantic segmentation [Yur+20]. Other control applications in automation and robotics also benefit from DNNs, as demonstrated in fields like agriculture [dOeS23].

The increasing use of DNNs is driven by enhanced neural network accuracy and their growing capabilities. For example, image-based detection of skin lesion cancer by DNNs is accurate by $98\,\%$ [Sal+23]. These enhancements are mainly based on new types of neural network models, larger model sizes, and increasing data sets for the model training [Sch+20]. However, this rise in model development was only possible with significantly improved hardware systems. Dedicated accelerators are utilized to train and execute these networks, broadening the design space beyond traditional Central Processing Unit (CPU) and Graphics Processing Unit (GPU) clusters. The accelerators are gaining popularity because their designs are tailored explicitly for neural network tasks [Reu+22]. This evolution allows for the training of larger models on more extensive datasets.

Apart from the model training, which has to be conducted only once, mainly on a cluster computer, the repeated execution of the pre-trained models, called inference, remains challenging. Due to the wide range of heterogeneous applications, a suitable hardware system has to be selected for the model execution based on the requirements of the DNN task. Fig. 1.1 shows a typical design flow to accelerate

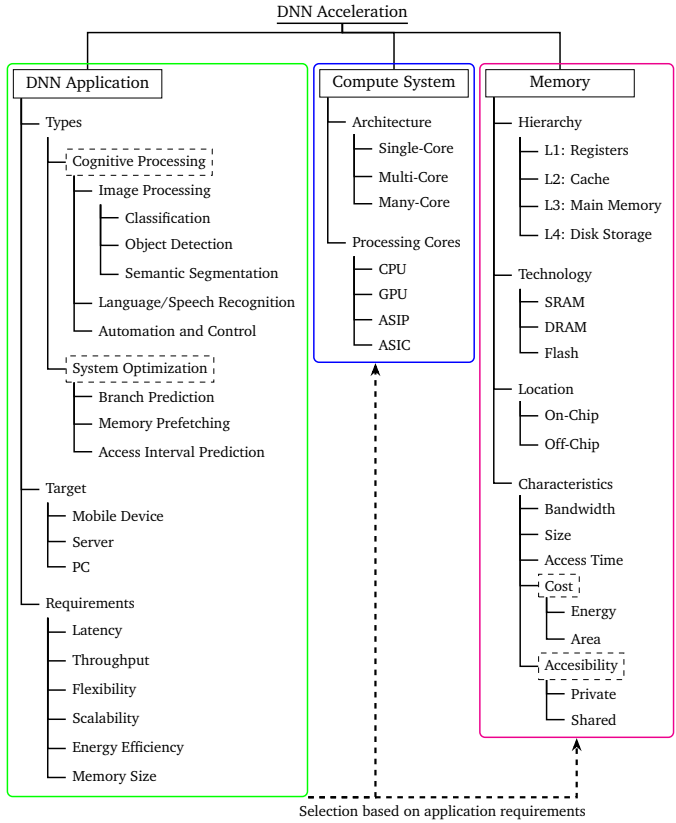

Fig. 1.1.: Overview of the design flow for DNN hardware execution after training.

a DNN application on a hardware system. The starting point for accelerating such an application is a pre-trained DNN. In this thesis, we distinguish between cognitive applications accelerated by a computing system and applications focused on system optimization. The previously mentioned examples fall into the cognitive category, whereas the second category leverages DNNs to optimize the computing system itself. One example is neural network-aided memory prefetching [Shi+21]. Additionally, the requirements and target of each application can differ. Consequently, these parameters determine the choice of the hardware system, which consists of computing and memory components.

Regarding the compute system, the processing core is selected based on several trade-offs. Higher performance generally reduces flexibility but lowers power consumption [Blu+02]. For instance, applications that require very low latency and high energy efficiency may benefit from a dedicated Application-Specific Integrated Circuit (ASIC). If necessary, multiple processing cores can be integrated into a multi-core system. The selected memory system can be divided into different levels[i] mostly fabricated in different technologies [HP17]. Commonly, the memory hierarchy of a custom ASIC is further categorized by its location, on- or off-chip. Multiple memories from different levels with various characteristics are typically combined within the memory system to match the application's requirements.

Within the hardware, both systems, compute and memory, interact. To handle the interactions, the so-called memory subsystem serves as the interface between processing cores and physical memory. However, these interactions require improvements due to a phenomenon called *memory wall*. First introduced in [WM95], the increase in computing performance has been outpacing the bandwidth growth of memory and its interconnect. This trend is evident in any compute system and especially dominant in hardware designed to accelerate DNNs [Gho+24]. Fig. 1.2 illustrates performance scaling of server-grade AI hardware over more than 20 years. The compute performance triples every two years, while off-chip memory and interconnect bandwidth increase by factors of only 1.6 and 1.4, respectively — only half of the increase in the compute performance. Consequently, the available memory bandwidth is limited, and the disparity between processing speeds and memory

[i]Higher levels are located closer to the compute core but are labeled with a lower number.

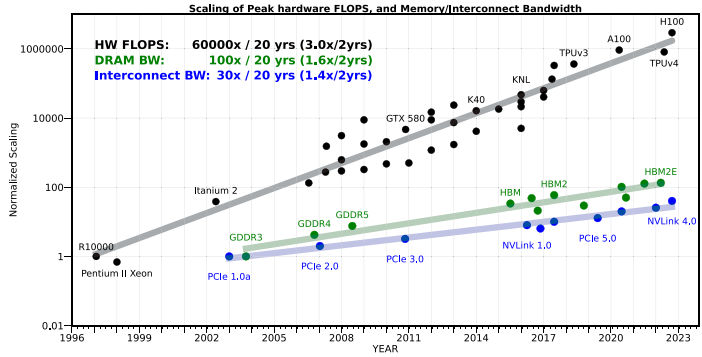

Fig. 1.2.: The history of performance scaling compared to bandwidth scaling of off-chip memory and interconnect [Gho+24].

access speeds creates a performance bottleneck in computing systems. Hence, the additional time for data access negatively impacts the overall system performance. As analyzed throughout this thesis, the *memory wall* issue also extends to on-chip memory, for example, in shared memory systems.

Several approaches are possible to overcome this issue. They can be divided into two sections. First, enhance the technology of the physical memory and its interconnect. Second, improve the memory subsystem itself to better balance the interactions between the compute and memory systems. This approach comprises enhancements to memory contention handling as well as workload-dependent advancements in the alignment of compute and memory.

1.1 Contributions to State-of-the-Art and Related Work

As mentioned at the beginning, there is a wide range of DNN applications with different requirements that affect the memory interactions. Therefore, we concentrate our analysis on two types of DNN applications, listed in Fig. 1.1. Access Interval Prediction (AIP) and image processing are chosen to address both system optimization and cognitive applications. As neural networks have not been applied to AIP yet, we focus on training these models first and provide optimizations toward a model execution in hardware. Whereas for image processing, we target only the hardware for network inference as a wide range of neural models has already been trained. Moreover, this thesis explores the DNN inference on a single hardware system. In the future, our findings could be applied to individual hardware units within distributed inference systems designed for applications with growing workloads. However, such systems must also address additional challenges, such as merging intermediate results and managing load balancing [PB24]. Finally, we focus on memory interactions within on-chip memory, representing the highest interaction level between the compute core and memory. These interactions either face a *memory wall* due to limited effective bandwidth, as seen in interconnects in shared memory systems, or mitigate the *memory wall* of off-chip memory by avoiding external memory transfers. Future work could explore enhancements to the lower memory levels within the memory subsystem to improve further the contributions made.

The contributions of this thesis are highlighted in Fig. 1.3. It shows an overview of approaches within the literature that are applied to counteract the *memory wall* [WM95] and reduce the gap between the performance of the memory and computing system.

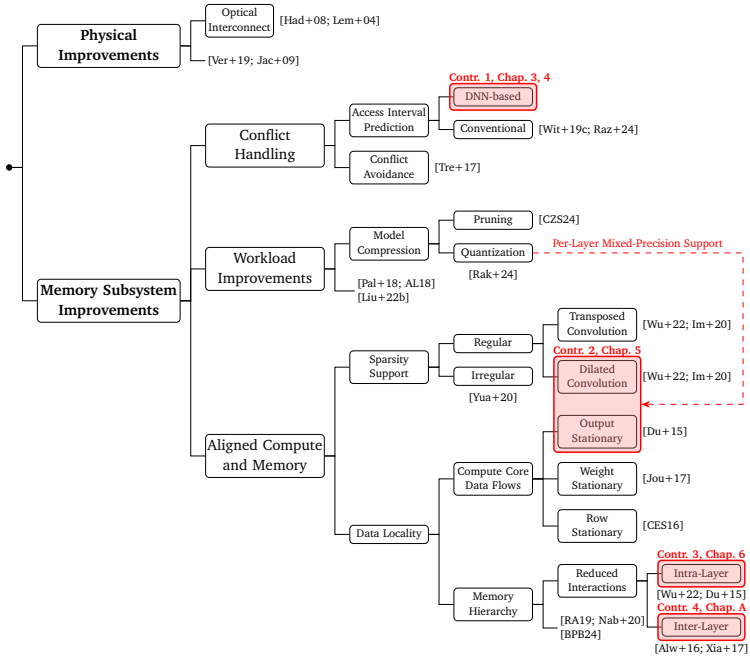

Fig. 1.3.: Overview of approaches and related research to counteract the *memory wall*. The four contributions of this thesis are put into context using this diagram.

Physical improvements contain several methods to improve the bandwidth of the memory and interconnect and to reduce their access times. This helps to increase the existing slope of the performance line in Fig. 1.2 for upcoming memory systems. In the field of interconnects, an increase in bandwidth can be achieved by replacing electrical with optical interconnects [Had+08; Lem+04]. For the memory cell itself, In-Memory Computing (IMC) can be applied. As stated by its name, IMC executes the computations directly within the memory, resulting in bandwidth and latency benefits [Ver+19]. Furthermore, three-dimensional memory stacking decreases the access latency of caches for high-speed compute systems [Jac+09]. However, many of these technological improvements are not yet within large-scale production and typically require costly and special production techniques. Therefore, this thesis explicitly does not deal with technologies and concentrates on the upcoming approaches to improve systems with conventional memories.

Memory subsystem improvements aim at enhancing the interactions between the compute and memory system. They can be divided into three subcategories.

Conflict handling deals with shared memory systems. As on-chip memory requires a considerable amount of chip area, memory sharing among multiple masters is a typical approach, especially for embedded systems. However, this technique can result in access conflicts, which decrease the already limiting access speed to the memory system. Within embedded systems, typically Tightly Coupled Memory (TCM) is implemented due to its deterministic and fast access times. According to previous work [Wit+19c], conventional conflict avoidance principles such as [Tre+17] are not applicable due to timing constraints of TCM or are limited for dynamic scheduled systems. Therefore, the conflicts further degrade compute performance by reducing effective memory bandwidth. This is because data access latency increases not only from unavoidable memory contention but also from the additional delays required for conflict resolution. For example, to maintain the area-related advantages of memory sharing, conventional online conflict resolution methods for shared TCM, like [Rah+11], resolve the conflict in a single cycle but degrade the system performance in terms of maximum clock frequency. Therefore, [Wit+19c] suggested AIP for an offline conflict resolution to predict the occurrences of memory accesses. This method enables fast memory arbitration and results in fast conflict resolution by pre-allocation, which minimizes performance degradation compared to online methods. As our **first contribution**, we introduce DNN-based predictors for executing AIP, a novel approach not previously explored. By analyzing the dataset, we demonstrate the feasibility of using neural networks for this task. Our trained models achieve lower miss rates across various programs from the MiBench benchmark suite than conventional designs such as the TAgged GEometric history length (TAGE) predictor [Raz+24]. Further, we expand the design space by cascading multiple neural network models, applying model pruning and quantization, and conducting a predictor latency analysis. As a result, system designers can define an acceptable miss rate and select the model with the lowest compute cost that meets the desired system execution time. Therefore, the *memory wall* issue, which affects explicitly systems with online arbitration [Rah+11], can be alleviated by increasing effective memory bandwidth through DNN-aided AIP and offline arbitration.

However, memory sharing and AIP are only partly applicable to DNN compute cores. On the one hand, custom-designed DNN accelerators typically show regular memory access patterns for their workloads. E.g., the cores in [Wu+22; CES16] show deterministic and highly regular patterns when executing CNNs, a major workload for DNN-based image processing. Hence, simple predictors like counters or dedicated memory control units that leverage regular access patterns as described

in [BPB24], are sufficient and can be employed for AIP. On the other hand, memory sharing is only possible for cores with a workload-dependent memory utilization below $100\,\%$. While this holds for CPUs, many DNN accelerators, Digital Signal Processors (DSPs), and GPUs do not meet this condition. The DNN core in [Jou+17] accesses its large $4\,\mathrm{MiB}$ memory every cycle. Therefore, we have to improve other parts of the memory subsystem for DNN cores beyond conflict handling to counteract the impact of the *memory wall*.

Workload improvements form an approach to reduce the amount and size of memory interactions for DNN cores. The size of DNN models has been significantly enlarged within the last few years. Not only the number of layers within a CNN has become larger, but also new categories of large neural network models such as Transformers have been introduced [Sze+15; Sch+20]. Therefore, several approaches have been developed to decrease the model size for these networks containing a large amount of model parameters. They range from weight encoding [Pal+18] to more advanced methods like low-rank approximation of CNNs and increase of sparsity in the attention layers of Transformers [Liu+22b]. However, even the classical model compression techniques show promising results for DNNs [Jan+24]. The quantization can be reduced to $8\,\mathrm{bit}$ and below without a significant degradation of the model accuracy [Rak+24]. The same applies to pruning, which omits unimportant weights that have a low impact [CZS24; LM23] on the model's output.

System alignment is the last subcategory and deals with improvements helping to align better the performance of the compute and memory system. The compute core has to support the advanced model compression techniques to take advantage of the reduced memory interactions. The already mentioned pruning is one compression method that typically results in irregular sparsity. However, the unpredictable positions of zero weights are difficult to handle for DNN accelerators, and dedicated costly hardware modules are necessary. E.g., the indexing module in [Yua+20] accounts for $18\,\%$ of the total power budget of the Processing Element (PE) array. In contrast, regular sparsity, introduced by advanced convolutions such as Dilated Convolution (DCONV) and Transposed Convolution (TCONV), is easier to process. Without efficiently processing these DCONVs, the corresponding layers are dominated by unnecessary data transfers and computations on zero values. Moreover, convolutions with different dilation rates are becoming more important in image processing [Yur+20; Nog+19]. Several accelerators with efficient handling of DCONVs have been introduced in the literature to target DNNs designed for flexible image processing [Im+20; Wu+22]. Both designs integrate an additional level of on-chip memory to avoid processing zero elements. The approach in [Im+20] employs shift registers in each PE, while [Wu+22] uses a register stage to reorder values loaded

from memory. However, their approaches are not universal and have limitations in the supported dilation rates. For larger dilation rates, they face inefficiencies such as extra zero transfers [Im+20] and additional latency [Wu+22]. The misalignment between the memory system and the compute core causes a resurgence of the *memory wall* issue in the form of pending memory transfers. As a result, the memory system again becomes a bottleneck, leading to increased computation time.

Additionally, the data locality and reuse within the compute core itself can be increased to ease the load on the memory subsystem. Therefore, several data flows have been introduced to keep loaded data stationary in the compute core. They can be clustered in output [Du+15], weight [Jou+17], and row stationary [CES16]. However, the performance, e.g., in terms of required memory bandwidth, main memory accesses, or energy, of each of them varies depending on the parameters and dimensions of the executed DNNs [Gui+19]. Hence, for each data flow, it is important to have a flexible interconnect to provide the required operands. However, efficient processing of the aforementioned DCONVs is a difficulty for all data flows. Although handling DCONVs with fewer restrictions compared to the weight stationary approach in [Im+20], the dilation rates of the output stationary design in [Wu+22] are still limited. Therefore, our **second contribution** is an improved output stationary DNN accelerator for handling DCONVs without restrictions. Apart from limited dilation rates, the previously introduced DCONV cores are only designed for fixed precision, causing the memory subsystem to process unnecessarily large memory transfers. To take advantage of the workload improvements of quantized models even for DCONVs, we propose a new memory mapping and a corresponding address generation scheme. This regular, flexible scheme supports any set of convolution parameters, including varying dilation rates. By leveraging our newly developed load unit with strided data access, we eliminate unnecessary zeros in DCONV processing. Additionally, our design efficiently supports operands with layer-wise varying precision for any convolution type, including DCONV. Consequently, we further reduce the memory transfer sizes compared to existing designs.

The data locality can be further increased by reducing the transfers within the memory hierarchy. When processing a dedicated layer, the data locality of a compute core can be enlarged by applying additional memory stages. A sorting buffer [Wu+22] or First-In First-Outs (FIFOs) between the PEs [Du+15] are implemented for their output stationary designs. Our **third contribution** is the integration of two established approaches within a DNN accelerator for the first time: an additional FIFO register stage, initially used for accelerators targeting only standard convolutions [Du+15], and load balancing for DCONVs [CC20a]. This combination is possible using our address generation scheme with strided memory access, which can also be applied to

writing data. As a result, our design reduces the number of memory accesses to the on-chip global memory for DCONVs, compared to the approach in [Wu+22]. Other designs apply topologies like meshes [RA19] to communicate between PEs with local memory. However, not having a direct path increases the latency between distant PEs, which slows down the interconnect and worsens the impact of the *memory wall* [Nab+20]. Another aspect is the dimensions of the memory hierarchy, which affects the number of on- and off-chip memory transfers. A memory control unit utilizing multiple levels of on-chip memory was introduced in [BPB24] to exploit regular memory access patterns during DNN execution. While it has demonstrated potential for optimizing weight memory, future work is needed to extend its application to the memory storing the input data of the DNN layers. Moreover, by fusing multiple layers of on-chip memory, off-chip transfers for intermediate data can be avoided and replaced by on-chip transfers [Alw+16]. We introduce a framework to determine the required number of fused-layers to reach local optima for required on-chip memory sizes and data transfers. Its principle is based on a line buffer approach [Xia+17]. However, the original method is constrained to a specific temporal execution order and is only applicable to accelerators with limited compute parallelism. For **our fourth contribution**, we extend this approach to larger accelerators with multiple execution orders and align it with our instruction set architecture.

Please refer to the related work sections in the following chapters for more detailed insights, as this section offers only a brief overview of each research area.

1.2 Outline

The rest of the thesis is structured as follows.

Chapter 2 defines and describes the memory subsystem and its interactions. Based on this, we introduce the system concept used throughout the thesis. Additionally, the chapter covers the fundamentals of neural networks and analyzes the selected software and hardware DNN applications, identifying their distinct requirements.

Chapter 3 uses this knowledge to introduce DNN-based AIP for conflict handling. This represents the first contribution of this thesis. Through analysis of interval distribution and access patterns, we demonstrate that this task can be modeled as a multi-class time-series forecasting problem with offline training. As a result, we develop a training framework for AIP to train and evaluate state-of-the-art DNNs, comparing them to conventional predictors.

Chapter 4 further enhances this contribution by reducing the computational resources required for neural network-aided AIP. We combine multiple small predictors to reduce the number of calculations per cycle while maintaining the desired execution time in a shared memory system with multiple PEs.

Chapter 5 describes the development of a new, regular, and universal instruction set for DNN accelerators, focusing on CNN image processing. This second contribution of the thesis proposes a design that efficiently supports per-layer mixed-precision operands and DCONVs to avoid unnecessary data transfers. Furthermore, we evaluate the benefits of our approach and integrate it into a comprehensive Systems on Chip (SoC) for autonomous driving applications.

Chapter 6 and Appendix A introduce two strategies to reduce low-level memory interactions. By applying minor adjustments to our instruction set, we can improve data reusability for DCONVs and reduce on-chip memory accesses, forming the third contribution of this thesis. Additionally, we extend the line buffer approach for layer fusion to an additional dimension and analyze the number of fused-layers needed to decrease the size of the on-chip memory in Appendix A, representing the fourth contribution.

Finally, Chapter 7 summarizes the research and provides an outlook on future work.