Carbon-Aware Design of DNN Accelerators: Bridging Performance and Sustainability

Aikaterini Maria Panteleaki and Iraklis Anagnostopoulos
School of Electrical, Computer and Biomedical Engineering, Southern Illinois University, Carbondale, U.S.A.
{aikaterinimaria.panteleaki,iraklis.anagno}@siu.edu

***Abstract*—The rise of Machine Learning (ML) and the wide use of Deep Neural Networks (DNNs) have led to the development of specialized DNN accelerators, aimed at improving the computing capabilities of ML systems. While these accelerators have been beneficial in enhancing computational efficiency, their design presents significant challenges due to the complexity of hardware configurations. Additionally, it is critical to consider and address the environmental implications and challenges. Unfortunately, these aspects have often been neglected during the design of ML systems. The carbon footprint, both operational and embodied, of these accelerators is a growing concern, with the latter becoming increasingly significant. This paper presents a framework for designing DNN accelerators with an emphasis on embodied carbon footprint. Using a genetic algorithm, we address the balance between performance and sustainability, focusing on reducing the embodied carbon footprint. Experimental results on different types of DNNs show that our method, by exploiting the properties of on-chip logic and memory, can generate DNN accelerators with considerably less embodied carbon and with a negligible performance overhead.**

***Index Terms*—Deep Neural Networks, Embodied Carbon Footprint, DNN Accelerators, Sustainable Computing**

I. Introduction

The growth of Machine Learning (ML) in the last years has been remarkable, making Deep Neural Networks (DNNs) a key component in modern computing systems. This has led to an increased demand for specialized DNN accelerators, which are computing architectures tailored to efficiently process the complex operations of DNNs. These accelerators are important for enhancing the speed and efficiency of ML systems, enabling them to handle more complex tasks faster than conventional computing components.

Designing hardware accelerators, within specific area constraints, presents a significant challenge, particularly due to the vast design space for hardware configurations and mappings [1]. This complexity is based on deciding the number of Processing Elements (PEs), as well as the configurations of local and global memories, which can greatly differ and significantly affect the accelerator's performance. The task of mapping DNNs onto these hardware accelerators introduces another dimension of complexity, significantly increasing the search space. This complexity is further increased due to the inter-dependencies between hardware configurations and DNN mapping strategies, where choices in one area can severely influence outcomes in the other. Previous works have shown that the possible hardware configurations are in the order of billions [2]. These factors highlight the critical need for developing sophisticated automation methods capable of co-exploration of hardware configurations and DNN mappings, avoiding also manual tuning.

Although the field of ML is evolving rapidly, it is critical to consider and address the environmental implications and challenges. Unfortunately, these aspects have often been neglected during the design of hardware accelerators [3]. In particular, operating and designing hardware accelerators carries a substantial environmental operational and embodied carbon footprint. The term operational refers to the carbon footprint associated with the ongoing operation and maintenance of the ML systems, including energy consumption and cooling requirements, while embodied refers to the carbon footprint associated with the entire life cycle of the devices, including their design, manufacturing, and disposal. Although, previous works focus on the impact of operational carbon footprint [4], [5], recent studies showed that the embodied carbon footprint of systems is becoming a dominating factor for ML's overall environmental impact [6] and optimizations at that level are still unexplored. In particular, previous studies have demonstrated that the use of hardware accelerators can substantially reduce the operational carbon footprint and energy consumption of DNN training [7]. However, these accelerators require more system resources, leading to larger embodied carbon footprints [8].

Therefore, designing sustainability-based DNN accelerators should move beyond traditional optimization methods. Specifically, the embodied carbon footprint can be reduced by scaling down energy-efficient hardware accelerators and lowering footprint circuit design. However, this approach introduces a performance and sustainability-oriented dilemma. On one hand, minimizing the size and increasing the efficiency of hardware components can lead to lower energy consumption during operation, contributing to a reduced operational and embodied carbon footprint. On the other hand, targeting high performance requires sophisticated and often resource-intensive hardware designs, which can increase the embodied carbon footprint through the use of more complex manufacturing processes, and increased resource utilization. This delicate balance between enhancing performance and reducing embodied carbon poses significant challenges.

In this paper, we present a framework for carbon-aware design of DNN accelerators. In particular, we present a genetic algorithm-oriented method to design hardware accelerators, under a specific area budget, considering hardware architecture, mapping of DNNs, and embodied carbon. The innovations of our work are manyfold: (1) Our framework simultaneously considers sustainability alongside traditional first-order metrics like performance and power efficiency for optimization, ensuring a holistic approach to accelerator design. (2) We integrate embodied carbon modeling directly into the design process of DNN accelerators, utilizing sustainability-oriented metrics as key decision-making tools. (3) Our approach reduces the embodied carbon footprint of the DNN accelerator with negligible performance impact.

II. Related Work

Modern research towards the estimation of embodied carbon emissions has been derived from data in Life Cycle Assessment (LCA) reports [9], [10]. Although the investigation of such analyses is very important, LCA summaries cannot be considered in early-stage design space exploration, as they provide coarse-grained information, that usually corresponds to older semiconductor technologies [11]. Other models [12], [13] may correlate embodied carbon footprint with only one parameter like the die area or the manufacturing cost. However, as it has been proven, the CO_2 emissions depend on multiple factors, like the fab characteristics and the technology of the transistors [14], so a more detailed method should be deployed.

Although there have been various studies [15], [16] that investigate the energy efficiency and the optimal utilization of hardware resources in DNN accelerators, it is not sufficient concerning the sustainability. In reality, such methods may even increase the manufacturing footprint, due to the additional circuit control complexity [17]. The need for sustainability has led to novel optimization metrics, used during the accelerator design phase. Apart from performance, power and area, the device carbon footprint is also taken into account. There have been several approaches [11], [6], [18] that combine both embodied and operational CO_2 emissions throughout the life of the accelerator. Nevertheless, operational and embodied carbon are estimated on different scales and therefore cannot be practically compared [19]. For this reason, in our optimization strategy, we consider only the embodied carbon footprint, which is responsible for most of the environmental impact of edge devices [6].

Regarding the DNN accelerator Design Space Exploration (DSE), previous methodologies tend to use optimization methods that examine both the hardware configuration and the mapping strategy, given a specific workload. Many studies [20], [2], [21], [22] divide the workload in layers and perform an optimization strategy examining each layer separately, ignoring any inter-layer modification. Each final design point has to satisfy some resource constraints, such as power consumption and area, which are very crucial in edge devices. Certain tools [21], [22] may find an optimal solution using a two-loop searching algorithm, by first selecting a hardware configuration and then reaching the most efficient mapping strategy for the predefined architecture. The process recurs in a feedback closed loop, according to the optimization procedure. However, this method results in a very large sample space and, in order to get a solution in a reasonable time, the authors have to substantially restrict the possible design points. Other works [20], [2] choose to flatten the hardware dataflow search space into one loop and thereby achieve faster results and better sample efficiency. Among these DSE solutions, Digamma framework [2] implements a domain-aware genetic algorithm to find an optimal solution under specific resource constraints, for a given neural network. The estimation of latency, energy and hardware requirements is employed by the cost model MAESTRO [23]. The differentiators of our work lies in the formulation of the embodied carbon for DNN accelerators, the integration of this metric into the design process, and the exploration of the vast design space.

III. Background and Motivation

The need for DNN accelerators arises from the increasing complexity and computational demands of modern neural networks. These accelerators are designed to efficiently execute the massive number of computations required by DNNs, enabling faster and more energy-efficient inference and training processes. In this work, we utilize the Eyeriss accelerator [24] as the architectural template for our exploration. The Eyeriss accelerator has a unique internal architecture that enables efficient processing of convolutional neural networks. At its core, Eyeriss consists of a large array of Processing Elements (PEs) organized in a mesh-like structure. Each PE is responsible for executing a specific portion of the neural network computation. The PEs are connected through a network-on-chip that facilitates data communication and synchronization between them. Additionally, Eyeriss incorporates a dedicated memory hierarchy that includes local memories associated with each PE, as well as a shared buffer for storing intermediate results. This hierarchical memory organization minimizes data movement and maximizes data reuse, leading to improved energy efficiency and performance.

As aforementioned, the embodied carbon footprint refers to the footprint associated with the entire life cycle of the devices, including their design, manufacturing, and disposal. The goal of this work is to associate the embodied carbon footprint with the architectural characteristics of the accelerator considering a standard manufacturing process. Particularly, considering a DNN accelerator that follows the Eyeriss architectural template, the embodied carbon emissions originate predominantly from the fabrication of PEs, the SRAM memories (local and global) and the chip packaging. However, the calculation of such a quantity is a multidimensional process that cannot be simplified into a linear model based solely on chip area or financial cost [11], [18]. Thus, to derive the CO_2 emissions, we follow the computation of the overall embodied carbon of the accelerator using ACT [11] as a baseline. Since ACT follows a more coarse-grain approach, we enriched it with more on-chip information using the model in [18]. Specifically, we considered the internal architecture of the accelerators, in terms of PEs and memories, the footprint of raw material extraction, the node technology in use, the wafer yield, as well as the final die packaging. For a single DNN accelerator chip, the embodied carbon footprint is calculated as:

$$C_{embodied} = C_{logic} + C_{memory} + C_{packaging} \quad (1)$$

where the embodied carbon of each component is given by the formula [11]:

$$C^i_{embodied} = (CI_{fab} \times EPA + MPA + GPA) \times \frac{A}{Y} \quad (2)$$

The carbon intensity of the fabrication facility's electrical grid is denoted as CI_{fab}, while EPA represents the energy used by the fab per unit area of the die. MPA indicates the carbon footprint of materials procured for manufacturing per unit area, and GPA refers to the direct emissions from gases used in the fabrication process. A stands for the area of the die, and Y symbolizes the yield of the fabrication process.

Previous research has demonstrated that designing DNN accelerators solely with performance metrics in mind significantly impacts the embodied carbon of these systems [11]. Recognizing this, we

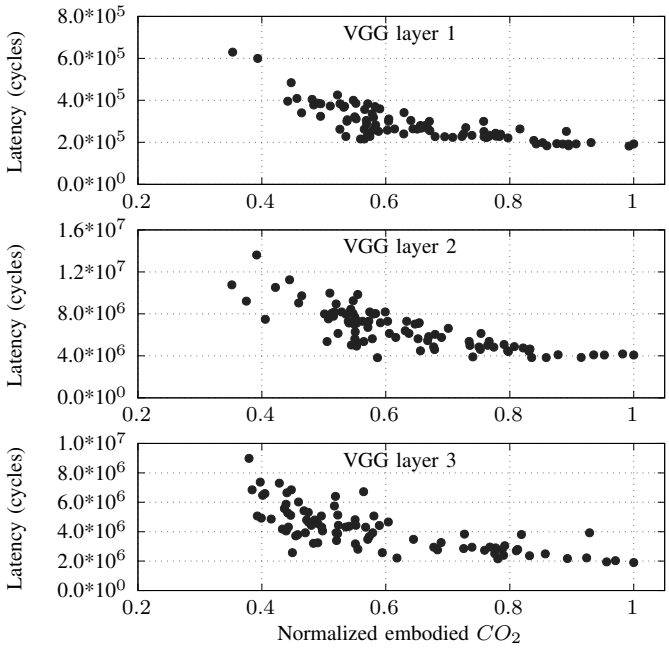

Fig. 1: Pareto front between cycle count (latency) and normalized embodied CO_2 for the first three layers of the VGG network.

present a motivational example that shows the importance of including embodied carbon in the design criteria for DNN accelerators. *Our primary objective is to demonstrate that it is possible to design DNN accelerators with considerably lower embodied carbon without compromising on performance*. In our motivational example, we utilize MAESTRO [23] to model and evaluate DNN accelerator designs and mappings under the same area budget (the exact experimental set up will be presented in detail in Section V). Specifically, we focus on the first three layers of the VGG16 network. Understanding that different layers may require distinct architectural optimizations, we used MAESTRO to generate *mutliple random* DNN architectures for each of the three layers, all within an area constraint of $0.2mm^2$. This approach aligns with previous works showing that the optimal design varies significantly from layer to layer [2]. Figure 1 shows the Pareto front between cycle count (latency) and normalized embodied CO_2 for the three first three layers of VGG16.

Based on Figure 1, we observe that accelerator designs that achieve lower latency tend to have higher embodied carbon, primarily due to the need for more PEs. This correlation suggests that designs optimized solely for speed may inadvertently lead to an increased embodied carbon footprint. Interestingly, our example also shows that for all layers examined, there are viable design solutions that achieve substantially lower embodied carbon while still maintaining low latency. This is particularly important as it pinpoints that focusing exclusively on latency can lead to over-design. Often, such strict optimization for minimal latency may not always be necessary, and a more balanced approach could yield equally effective, yet more sustainable solutions. Additionally, all the setups that we showed were randomly generated. This is because the whole design space is vast. For example, in a static hardware architecture a single layer of a DNN can be mapped in $O(10^{24})$ different ways [2]. This leads to an important question: *How can we effectively explore this expansive space to identify designs that optimally balance performance with reduced embodied carbon?* Efficient exploration strategies are essential for systematically identifying the most sustainable and efficient designs, while still achieving low latency. Finally, another interesting aspect is that the first layers of the VGG network are more compute-intensive compared to later layers. This characteristic affects the design strategies that can be applied, as the early layers may require more robust hardware configurations, potentially impacting both performance and embodied carbon.

Based on the previous analysis, first-order metrics, such as latency, provide valuable insights into the performance of a DNN accelerator architecture. However, they are not sufficient for a comprehensive evaluation that takes into account the environmental impact. To integrate and quantify the carbon footprint associated with the accelerator, *it is necessary to introduce new metrics*. In this work, we utilize the carbon delay product (CDP) as a metric that combines the performance aspects with the embodied carbon footprint. The CDP metric considers the total delay incurred by the accelerator during the execution of a neural network, taking into account both the computational time and the associated embodied carbon footprint. By incorporating the CDP metric, we can evaluate the architectural characteristics of the accelerator in terms of its impact on both performance and embodied carbon footprint.

IV. Methodology

Figure 2 presents an overview of our proposed methodology. Our objective is to design the hardware architecture of a DNN accelerator (for each layer of a DNN) and determine the corresponding mapping to optimize the Carbon Delay Product (CDP). As previously mentioned,

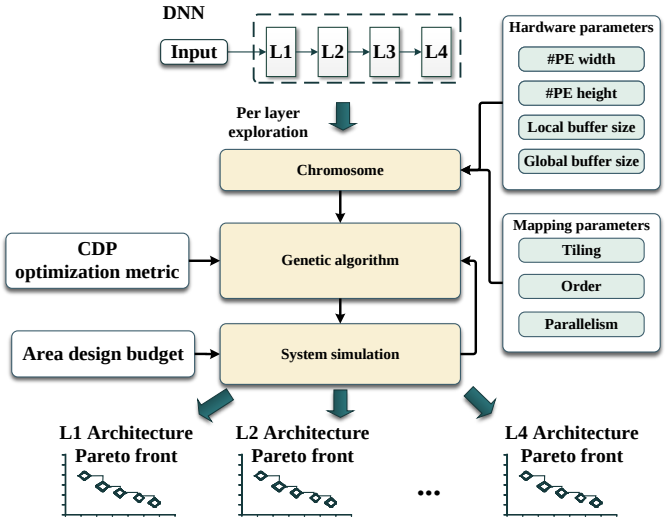

Fig. 2: Overview of the proposed methodology

the CDP is a metric that integrates performance measures with the embodied carbon footprint, offering a holistic assessment. In our methodology, we explore various hardware characteristics, including the width and height of the DNN accelerator, defined by the number of Processing Elements (PEs), the capacity of the local buffer for each PE, and the size of the shared global buffer. Additionally, the mapping characteristics considered are tiling, order, and level of parallelism of execution. Given the vast design space of potential solutions, we employ a genetic algorithm to navigate this complexity efficiently. Our approach aims to generate a Pareto front that represents optimal trade-offs between performance and embodied carbon, under the constraint of a predefined area budget for the final designs.

The first step in our methodology is the design of the chromosome, later used in the genetic algorithm for optimizing DNN accelerator designs. In genetic algorithms, a chromosome is essentially a structured representation of a solution's variables. In this work, the chromosome encodes comprehensive information about both hardware characteristics and mapping strategies of the DNN accelerator. Regarding the hardware aspects, the chromosome includes information such as the number of Processing Elements (PEs), organized in an XY mesh to support specific interconnections and data flow strategies. It also contains the capacity of the local buffer for each PE and the size of the shared global buffer. These elements are vital as both PEs and memory systems are the major sources of a chip's embodied carbon footprint. Moreover, their arrangement and cooperation impact execution runtime significantly, as the logic and memory elements present complex inter-dependencies that need to be investigated in depth, for an optimal selection. We mainly focus on DNN models that are either explicitly Convolutional Neural Networks (CNN), or their operations can be simplified into a set of convolutions, even if they are not traditional CNNs. This approach allows us to represent every layer operations with a multi-dimensional for-loop, that is flexible in terms of loop ordering, sectioning and parallelization. Accordingly, the mapping part of our chromosome follows the optimizations described in [2], a comprehensive framework for hardware mapping. This includes tiling, which defines how tensors are sliced, stored, and fetched within the memory hierarchy, effectively managing data locality and access patterns. It also includes the compute order, specifying the sequence in which computational operations are executed, and the level of parallelism, which determines how computations are distributed across

the PEs. These mapping characteristics are critical as they directly affect the execution speed and efficiency of the DNN, ensuring that the hardware's capabilities are fully utilized. By integrating both hardware characteristics and mapping strategies in a single chromosome, the algorithm achieves design space co-exploration, providing both sample efficacy and convergence speed. Moreover, the efficient selection of the sample design points leads to identifying solutions that optimize the CDP while addressing the trade-offs between performance and embodied carbon footprint.

In the context of designing DNN accelerators under specific constraints, our problem formulation requires finding the optimal configuration of Processing Elements (PEs), local buffers, global buffers, and efficient mapping for each layer of a DNN. The goal is to optimize the Carbon Delay Product (CDP), balancing performance and embodied carbon footprint, all within a predetermined chip area budget.

A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This algorithm represents potential solutions as chromosomes, which evolve over generations to find the most optimal solution to a problem. In our case, as mentioned above, each chromosome encodes different configurations of hardware and mappings that collectively determine the accelerator's performance and carbon footprint. The fitness of each chromosome is evaluated using the CDP as a reward function, which integrates latency with embodied carbon footprint considerations. If a solution exceeds the chip area budget, it is penalized, effectively receiving a fitness score of negative infinity. This ensures that non-viable solutions are quickly discarded from the population pool. The algorithm iteratively refines the population through the processes of selection, crossover, mutation, and aging. Crossover is a process where segments of two parent chromosomes are combined to produce offspring, potentially inheriting the strengths of both parents. Mutation, on the other hand, introduces random alterations to a chromosome. This helps maintain diversity within the gene pool and prevents the algorithm from becoming stuck in local optima. Aging is another mechanism used in GAs to prevent the stagnation of the population. Older solutions may be phased out over time, allowing newer solutions, potentially with better adaptability to the problem constraints, to dominate the population. This ensures that the population does not converge prematurely and continues to explore new areas of the solution space. Over time, as less fit solutions are discarded and more promising solutions are promoted, the GA converges towards a solution that optimally balances the CDP while adhering to the area constraint. Through these mechanisms, the genetic algorithm effectively searches through a vast and complex design space, gradually evolving and converging toward an optimal solution that meets the specific needs of DNN accelerator design within the constraints provided.

V. Evaluation

In this section, we evaluate the effectiveness of our proposed framework through a detailed analysis of various DNNs. We have enhanced the functionality of the ACT framework [11] by improving its support for on-chip components specifically designed for DNN accelerators. Subsequently, we integrated our enhanced model into MAESTRO, a tool used for modeling and evaluating the performance of different dataflows in DNN architectures. This integration allows us to assess the impact of our method across multiple DNNs

We evaluated five distinct DNNs from various domains to investigate their performance and the associated embodied carbon footprint of each solution. These included VGG16 and Alexnet for computer vision, BERT and ALBERT for language processing, and T5 for text processing. For each layer within these DNNs, we used our

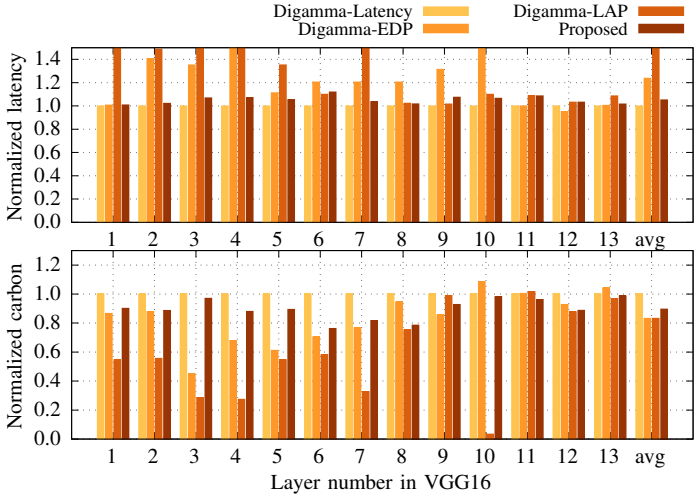

Fig. 3: Comparative analysis of normalized latency and embodied carbon for DNN accelerators designed for VGG16 using different optimization methods.

methodology to identify the architecture that provided the best Carbon Delay Product (CDP) reward. We then assessed the performance of these optimized architectures in terms of execution latency (measured in cycles) and the associated embodied carbon. For comparison, we used Digamma [2], a hardware-mapping co-optimization framework, examining how each layer performed when optimized for different criteria: 1) latency; 2) energy delay product (EDP); and 3) latency area product (LAP). It is important to note that an area constraint of $0.2mm^2$ was set across all these optimization metrics. This number is often used for edge-based accelerators [25] and allowed us to fairly assess the trade-offs and efficiencies of the different optimization strategies. Finally, the design optimized for latency was selected as the baseline for all the following experiments.

In our experimental analysis of VGG16, presented in Figure 3, we show the normalized latency and normalized embodied carbon footprint under four different metrics. The results for each layer are presented individually, while the groups of bars at the end shows the averages across all layers. As expected, the method focused on optimizing latency achieves the highest performance. However, our approach demonstrates a small performance overhead of only about 5% on average. Despite this slight increase in latency, the CDP method significantly reduces the embodied carbon footprint by 11% compared to the baseline. The other two methods, EDP and LAP, also reduced even more the embodied carbon footprint, with reductions up to 17% on average compared to the baseline. Nonetheless, these benefits came at a considerable cost to latency performance. Notably, the LAP method increased latency by more than 40%, highlighting a substantial trade-off between embodied carbon footprint and performance. This analysis underscores the complexity of balancing performance with sustainability in DNN accelerator design, particularly when adapting to various optimization priorities.

Figure 4 shows the comparison of the DNN accelerators's designs under the four different optimization metrics for Alexnet. Similarly, our method that focuses on CDP, achieves an average reduction of 18% regarding the embodied carbon footprint, with a performance overhead of only 6%. Interestingly, the LAP method achieved even lower embodied carbon footprint, but with a considerable performance overhead of 31%.

In our analysis of the experimental results for ALBERT, presented

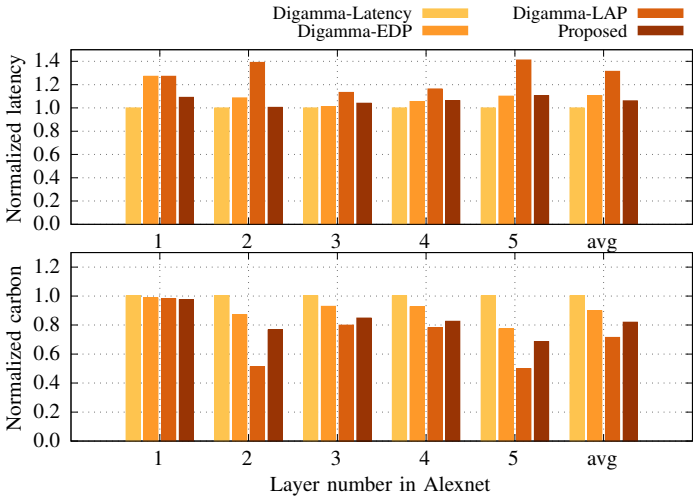

Fig. 4: Comparative analysis of normalized latency and embodied carbon for DNN accelerators designed for Alexnet using different optimization methods.

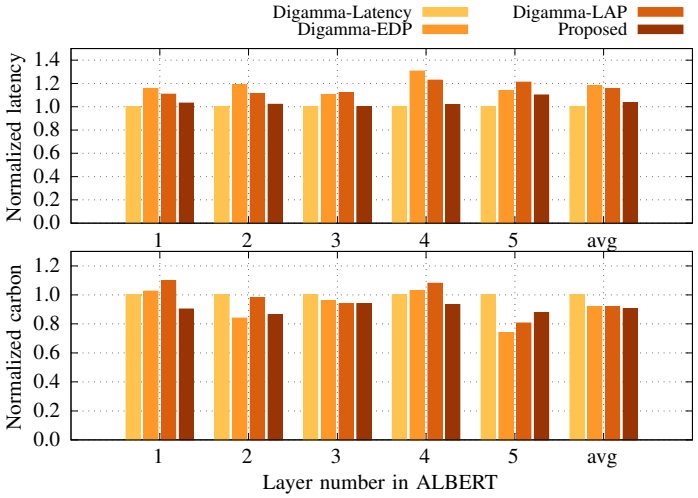

Fig. 5: Comparative analysis of normalized latency and embodied carbon for DNN accelerators designed for ALBERT using different optimization methods.

in Figure 5, we examined the normalized latency and normalized embodied carbon footprint across the various optimization metrics and methods for each layer of the DNN. Again, as expected, the optimization method focused solely on minimizing latency had the highest performance (lowest latency). Conversely, our CDP-based method introduced a minimal performance overhead, averaging only about 4%. Additionally, it also reduced the embodied carbon footprint by 8% compared to the baseline. The other two methods, EDP and LAP, while also reducing the embodied carbon footprint relative to the baseline, had a more substantial impact on latency, with an average increase of up to 15%. This happened due to their inability to simultaneously account for the effects that logic and memory configurations have on both performance and embodied carbon. Overall, the results confirm that it is feasible to design DNN accelerators for ALBERT that significantly lower the embodied carbon footprint with only a negligible compromise in performance, showcasing the effectiveness of our CDP approach.

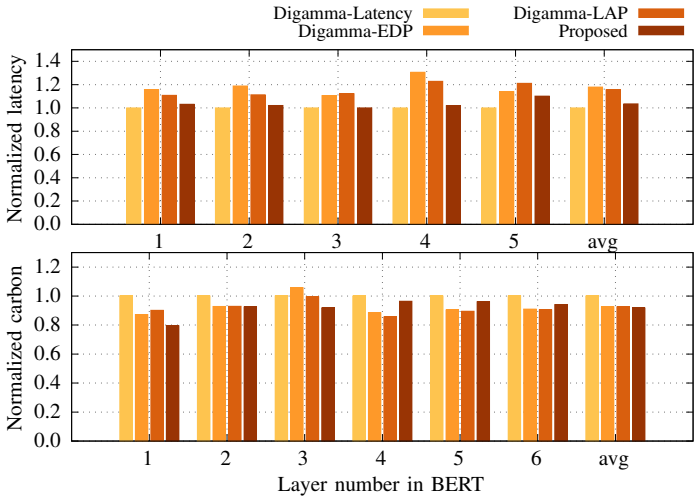

Fig. 6: Comparative analysis of normalized latency and embodied carbon for DNN accelerators designed for BERT using different optimization methods.

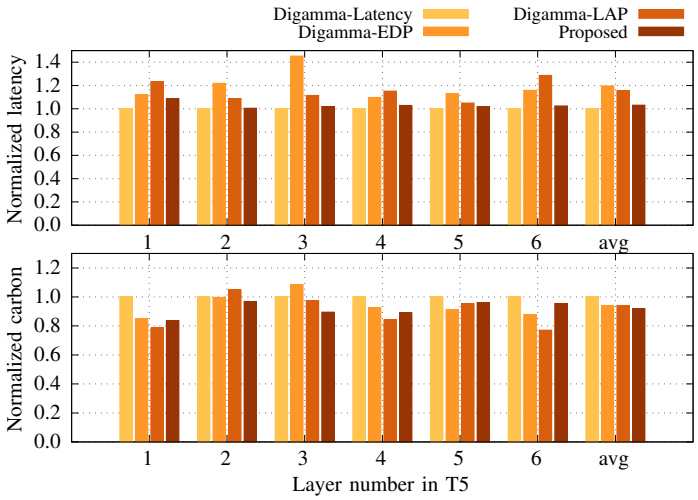

Fig. 7: Comparative analysis of normalized latency and embodied carbon for DNN accelerators designed for T5 using different optimization methods.

Figure 6 and Figure 7 shows the results for BERT and T5 accordingly. The experimental results exhibited patterns consistent with those observed in our previous experiments. In these cases, the baseline configuration achieved the lowest cycle, indicating the highest performance in terms of speed. However, our method achieved the most advantageous balance between performance and the embodied carbon footprint. Although the LAP method did achieve the lowest embodied carbon footprint among the strategies tested, it came with a considerable performance overhead. This significant increase in cycle count under the LAP optimization illustrates the trade-offs inherent in prioritizing environmental metrics over operational speed. Such results underscore the effectiveness of our CDP method in providing a more holistic approach to DNN accelerator design, optimizing both environmental impact and computational efficiency.

VI. Conclusion

The rapid advancement of machine learning and the increasing use of DNNs require the development of specialized accelerators designed with both performance and environmental considerations in mind. In this paper, we emphasize the importance of incorporating carbon-aware principles into the design of these accelerators, which extends beyond traditional performance metrics. By using a genetic algorithm that evaluates hardware architecture, DNN mappings, and sustainability within a specific area constraint, our approach tackles the complex design challenges posed by numerous potential hardware configurations. The outcomes demonstrate the feasibility of reducing the embodied carbon footprint with negligible performance overhead, offering an alternative approach to more sustainable ML operations.

Acknowledgments

This work has been supported by grant NSF CCF 2324854.

References

[1] Q. Huang, M. Kang, G. Dinh, T. Norell, A. Kalaiah, J. Demmel, J. Wawrzynek, and Y. S. Shao, "Cosa: Scheduling by constrained optimization for spatial accelerators," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2021, pp. 554–566.

[2] S.-C. Kao, M. Pellauer, A. Parashar, and T. Krishna, "Digamma: Domain-aware genetic algorithm for hw-mapping co-optimization for dnn accelerators," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 232–237.

[3] H. Amrouch, G. Zervakis, S. Salamin, H. Kattan, I. Anagnostopoulos, and J. Henkel, "Npu thermal management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3842–3855, 2020.

[4] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[5] J. Koomey and E. Masanet, "Does not compute: Avoiding pitfalls assessing the internet's energy and carbon impacts," *Joule*, vol. 5, no. 7, pp. 1625–1628, 2021.

[6] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing carbon: The elusive environmental footprint of computing," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 854–867.

[7] D. Patterson *et al.*, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[8] C.-J. Wu *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[9] S. Tannu and P. J. Nair, "The dirty secret of ssds: Embodied carbon," *ACM SIGENERGY Energy Informatics Review*, vol. 3, no. 3, pp. 4–9, 2023.

[10] S. Prakash, M. Stewart, C. Banbury, M. Mazumder, P. Warden, B. Plancher, and V. J. Reddi, "Is tinyml sustainable?" *Communications of the ACM*, vol. 66, no. 11, pp. 68–77, 2023.

[11] U. Gupta, M. Elgamal, G. Hills, G.-Y. Wei, H.-H. S. Lee, D. Brooks, and C.-J. Wu, "Act: Designing sustainable computer systems with an architectural carbon modeling tool," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 784–799.

[12] L. Eeckhout, "A first-order model to assess computer architecture sustainability," *IEEE Computer Architecture Letters*, vol. 21, no. 2, pp. 137–140, 2022.

[13] M. Elgamal, D. Carmean, E. Ansari, O. Zed, R. Peri, S. Manne, U. Gupta, G.-Y. Wei, D. Brooks, G. Hills *et al.*, "Design space exploration and optimization for carbon-efficient extended reality systems," *arXiv preprint arXiv:2305.01831*, 2023.

[14] Z. Zhang, R. Wang, C. Chen, Q. Huang, Y. Wang, C. Hu, D. Wu, J. Wang, and R. Huang, "New-generation design-technology co-optimization (dtco): Machine-learning assisted modeling framework," in *2019 Silicon Nanoelectronics Workshop (SNW)*. IEEE, 2019, pp. 1–2.

[15] O. Spantidi, G. Zervakis, I. Anagnostopoulos, and J. Henkel, "Energy-efficient dnn inference on approximate accelerators through formal property exploration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3838–3849, 2022.

[16] O. Spantidi, G. Zervakis, S. Alsalamin, I. Roman-Ballesteros, J. Henkel, H. Amrouch, and I. Anagnostopoulos, "Targeting dnn inference via efficient utilization of heterogeneous precision dnn accelerators," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 1, pp. 112–125, 2022.

[17] Z.-G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy-efficient neural network inference accelerators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4670–4683, 2020.

[18] Y. Zhao, C. Wan *et al.*, "3d-carbon: An analytical carbon modeling tool\\for 3d and 2.5 d integrated circuits," *arXiv preprint arXiv:2307.08060*, 2023.

[19] N. Bashir, D. Irwin, and P. Shenoy, "On the promise and pitfalls of optimizing embodied carbon," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems*, 2023, pp. 1–6.

[20] C. Hong, Q. Huang, G. Dinh, M. Subedar, and Y. S. Shao, "Dosa: Differentiable model-based one-loop search for dnn accelerators," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, 2023, pp. 209–224.

[21] Y. Lin, M. Yang, and S. Han, "Naas: Neural accelerator architecture search," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1051–1056.

[22] Q. Huang, C. Hong, J. Wawrzynek, M. Subedar, and Y. S. Shao, "Learning a continuous and reconstructible latent space for hardware accelerator design," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2022, pp. 277–287.

[23] H. Kwon, P. Chatarasi, V. Sarkar, T. Krishna, M. Pellauer, and A. Parashar, "Maestro: A data-centric approach to understand reuse, performance, and hardware cost of dnn mappings," *IEEE micro*, vol. 40, no. 3, pp. 20–29, 2020.

[24] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," *ACM SIGARCH computer architecture news*, vol. 44, no. 3, pp. 367–379, 2016.

[25] Y. S. Shao, J. Clemons, R. Venkatesan, B. Zimmer, M. Fojtik, N. Jiang, B. Keller, A. Klinefelter, N. Pinckney, P. Raina *et al.*, "Simba: Scaling deep-learning inference with multi-chip-module-based architecture," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 14–27.