A Memory-Efficient Framework for Deformable Transformer with Neural Architecture Search

Wendong Mao[a], Mingfan Zhao[a], Jianfeng Guan[a], Qiwei Dong[b], Zhongfeng Wang[a]
[a]School of Integrated Circuits, Sun Yat-Sen University, Shenzhen, China
[b]School of Electronic Science and Engineering, Nanjing University, Nanjing, China
Email: maowd@mail.sysu.edu.cn, zhaomf7@mail2.sysu.edu.cn, guanjf@mail2.sysu.edu.cn
qiweidong@smail.nju.edu.cn, wangzf83@mail.sysu.edu.cn

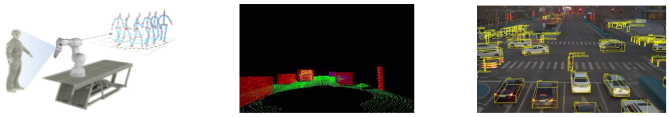
***Abstract*—Deformable Attention Transformers (DAT) have shown remarkable performance in computer vision tasks by adaptively focusing on informative image regions. However, their data-dependent sampling mechanism introduces irregular memory access patterns, posing significant challenges for efficient hardware deployment. Existing acceleration methods either incur high hardware overhead or compromise model accuracy. To address these issues, this paper proposes a hardware-friendly optimization framework for DAT. First, a neural architecture search (NAS)-based method with a new slicing strategy is proposed to automatically divide the input feature into uniform patches during the inference process, avoiding memory conflicts without modifying model architecture. The method explores the optimal slice configuration by jointly optimizing hardware cost and inference accuracy. Secondly, an FPGA-based verification system is designed to test the performance of this framework on edge-side hardware. Algorithm experiments on the ImageNet-1K dataset demonstrate that our hardware-friendly framework can maintain have only 0.2% accuracy drop compared to the baseline DAT. Hardware experiments on Xilinx FPGA show the proposed method reduces DRAM access times to 18% compared with existing DAT acceleration methods.**

***Index Terms*—Deformable Attention, Transformer, NAS, Acceleration.**

I. Introduction

Nowadays, Transformer [1] has shown outstanding performance in natural language processing (NLP). As the potential of Transformers became evident, researchers extended their application to computer vision (CV), leading to the development of the Vision Transformer (ViT) architecture [2]. ViT has achieved impressive results across various CV tasks, such as object detection [3], image classification [4], and image segmentation [5]. Based on ViT's self-attention mechanism, numerous subsequent works [6]–[8] have been proposed to further enhance the performance and efficiency of transformers on visual tasks. However, Xia et al. [9] pointed out that when too many keys correspond to a single query in visual Transformers, it can lead to high computational cost, slow convergence, and an increased risk of overfitting. This has inspired the emergence of new attention mechanisms that enable the key/value set for a given query to be both flexible and adaptive.

The development of Deformable Attention Transformer (DAT) [9] effectively solves the above issue. In its deformable attention mechanism, the sampling positions of key-value pairs are data-dependent rather than fixed. Consequently, the self-attention module can more effectively focus on important image regions that contain informative features. This flexible attention mechanism facilitates DAT's application across diverse domains such as robotics and autonomous driving, as illustrated in Fig. 1.



(a) Human Motion Prediction (b) Obstacle Detection (c) 3D Object Detection

Fig. 1. Practical applications of deformable attention.

However, deploying the DAT on edge platforms, such as intelligent vehicles and mobile devices, remains challenging due to the data-dependent nature of its deformable attention, which dynamically samples keys and values from feature maps. This data dependency induces random and conflicting memory accesses. Therefore, traditional Transformer accelerators like SpAtten [10] and ELSA [11] are not well-suited for accelerating deformable Transformers because of their unique architectural characteristics. DEFA [12] is a pioneering work to accelerate multi-scale deformable attention, which introduces pruning-assisted sampling to reduce the memory footprint of feature map sampling. Nevertheless, DEFA suffers from considerable hardware overhead. To address this limitation, this paper proposes a hardware-friendly and resource-efficient acceleration method for deformable attention.

In this paper, we propose a slicing-based acceleration strategy for the deformable attention mechanism, along with an optimal slice size search algorithm. The slicing strategy enables efficient computation of deformable attention on hardware platforms with limited resources. Furthermore, we introduce a neural architecture search (NAS)-based algorithm to determine the optimal slicing configuration.

The main contributions of this work are as follows:

- We propose a training-free slicing method, which does not change training process and only divides the input image into local patches during inference with the pre-established strategy. It decoupled the serial dependencies among different input tiles, reducing memory requirements without changing model architecture.
- We develop a memory-aware NAS algorithm to construct a continuous search space encompassing various slicing

strategies. It can determine the optimal slicing configuration, achieving the optimal balance between accuracy and hardware overhead.
- We deploy the proposed framework on the Xilinx FPGA platform, and experimental results demonstrate its multiple advantages in terms of hardware overhead and algorithm accuracy.

II. A Training-Free Slicing Method for Deformable Attention

Conventional self-attention modules acquire key and value information uniformly and sequentially across the image, enabling relatively straightforward application of acceleration techniques like parallel computation. In contrast, deformable attention mechanisms assign each reference point a learned random offset to determine its sampling position. This leads to completely random access patterns to the input image, resulting in the following challenges:

1) Memory access conflicts: Multiple reference points may be offset to nearby image locations and access identical sampling point information, causing simultaneous access to the same memory location.

2) Large memory overhead: Random access to the input image makes it necessary to store the entire input image in a large buffer for computing the sampled features. Meanwhile, the computation must be performed on the full image simultaneously, preventing dividing the image into independent regions for parallel processing, which results in significant memory and hardware resource overhead.

3) Serial processing dependency: Disordered memory access impedes parallel computing strategies and requires serial computations, which leads to low computational efficiency.

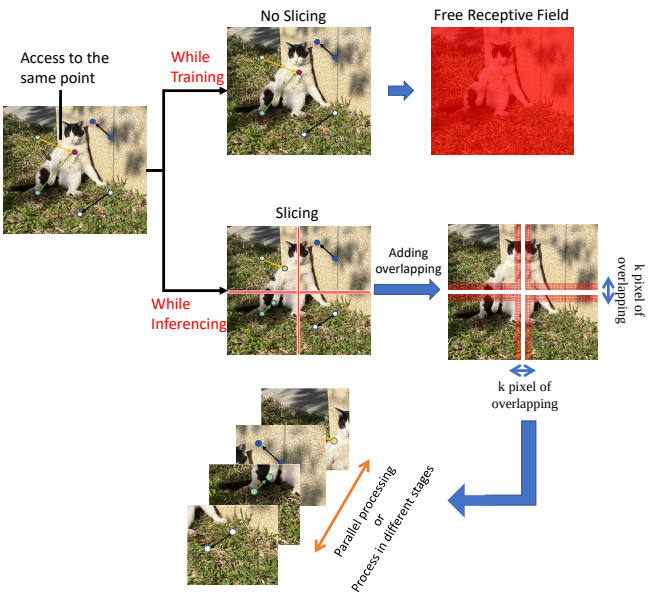


Fig. 2. Illustration of the proposed slicing strategy.

To address the aforementioned issues, we propose a traing-free slicing strategy, as illustrated in Fig. 2. Specifically, during inference, the input image is divided into several independent patches, while training process remains unchanged. Reference points in a given patch are constrained so that they cannot shift outside their own patch, and no attention is computed across patches. After slicing, each patch can be processed individually in separate stages, which greatly reduces hardware resource consumption. As the patch size decreases, the required hardware resources drop exponentially. This is particularly significant for deploying deformable transformers on resource-constrained platforms such as FPGAs and mobile devices. Moreover, when hardware resources permit, computations on different patches can be parallelized, further improving computational efficiency.

However, if a reference point in the original image is shifted beyond its patch after slicing, the constrained offset range may lead to a decrease in model accuracy. To mitigate this issue, we introduce an overlapping region of width k at the patch boundaries, as shown in Fig. 2. This overlapping region contains edge information from adjacent patches, which helps preserve model accuracy by alleviating the impact of restricted reference point movement.

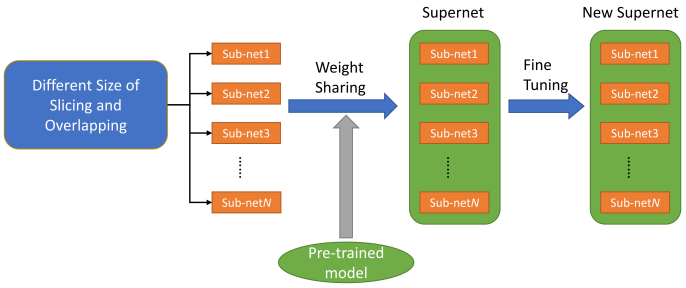
III. Neural Architecture Search for Optimal Slicing Strategy

For the aforementioned slicing strategy, both the size of the sliced image patches and the size of the overlapping regions are uncertain, constituting different slicing schemes. Smaller patch sizes are more hardware-friendly for deployment, but they also result in greater loss of model accuracy. It is difficult to manually balance these two aspects and find the globally optimal solution. Therefore, this paper proposes a NAS-based method to search for the optimal slicing strategy. As shown in Fig. 3, the procedure consists of three steps: supernet construction, fine-tuning, and optimal strategy search.

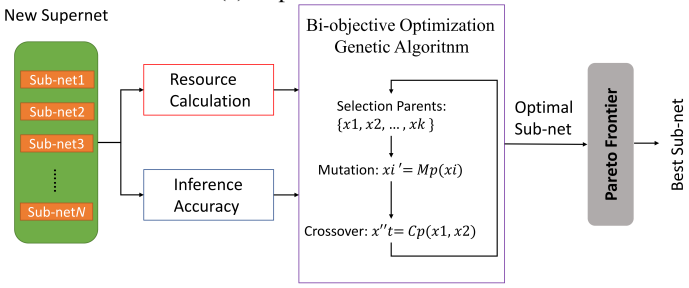
A. Supernet Construction and Fine-Tuning

To search for the optimal slicing strategy, we first construct a continuous search space. This search space contains all slicing schemes with different slicing sizes and overlapping sizes, and each slicing scheme corresponds to a sub-network (sub-net). These sub-nets collectively form a supernet. Specifically, the input image is divided into slices of size $H_S \times W_S$, where H_S and W_S can take Num_H and Num_W possible values, respectively. This creates a continuous search space containing $Num_H \times Num_W$ sub-nets. Furthermore, we introduce three sizes of overlapping: 0 pixel, 1 pixel, and 2 pixels. With the introduction of overlapping, the search space expands to include $3 \times Num_H \times Num_W$ sub-nets. Independently training each sub-net will incur significant costs. Therefore, in this work, all sub-nets share the same weight parameters. In other words, the slicing operation is only involved during inference and not during training.

Since all sub-nets share the same weights, it is only necessary to train a single supernet. After the supernet is trained, a fine-tuning process is required. The standard teacher-student



(a) Supernet Construction



(b) Optimal Strategy Search

Fig. 3. A NAS-based method to search for the optimal slicing strategy.

paradigm typically uses *KL*-divergence to measure the discrepancy between the teacher and student networks. However, for a supernet, *KL*-divergence may fail to cover one or more modes of the teacher model, leading to severe penalties on the student model. Therefore, inspired by AlphaNet [13], this work adopts α-divergence for fine-tuning the supernet:

$$D_{\alpha+,\alpha-}(p\|q) = \max\left\{D_{\alpha+}(p\|q), D_{\alpha-}(p\|q)\right\},$$
$$\text{where } D_\alpha(p\|q) = \frac{1}{\alpha(\alpha-1)}\sum_{i=1}^{m} q_i\left[\left(\frac{p_i}{q_i}\right)^{\alpha} - 1\right]. \quad (1)$$

Finally, the fine-tuning process of the supernet adopts the adaptive-KD loss within the α-divergence constraint, which is formulated as:

$$\mathcal{L}_{\mathrm{KD}}(\theta, A) = \mathbb{E}_{x\sim D}\left[D_{\alpha+,\alpha-}\left(p(x;\theta,\alpha_b)\,\|\,q(x;\theta,A)\right)\right], \quad \text{where } A \in \{\alpha_s, \alpha_r\}. \quad (2)$$

B. Optimal Strategy Search

After the construction and training of the supernet are completed, we search for the optimal sub-net within it. Two key metrics are first defined for the search:

1) Resource Consumption: The computational formula for resource consumption R is defined as follows:

$$R = \text{BitWidth} \times (W_S + O_W) \times (H_S + O_H) + \beta, \quad (3)$$

where W_S and H_S are the slice width and height, respectively, and O represents the size of the overlapping region. The parameter β is a constant indicating the resources consumed by other components of the hardware system.

2) Accuracy: The accuracy metric is defined as the inference accuracy of the sub-net.

Since the search for the optimal strategy involves balancing these two metrics, it can be formulated as a bi-objective optimization problem. As shown in Algorithm 1, our search employs a sampling-based genetic algorithm, where the search process constitutes a multi-objective genetic algorithm optimization with two objectives corresponding to the aforementioned search metrics. Specifically, each iteration of the search algorithm consists of four steps: selection, updating the Pareto frontier, mutation, and crossover. Mutation randomly modifies the selected slicing strategy, while crossover generates new slicing strategies by recombining two existing ones.

Algorithm 1 Bi-Objective Evolutionary Search for Neural Sub-nets

Initialize:
  Search space of sub-net architectures $\mathcal{X}$.
  Objective functions: $F_1(x) = \text{Acc}(x)$, $F_2(x) = \text{Resource}(x)$.
  Resource bounds: $R_{\min}$ (min resource), $R_{\max}$ (max resource).
for $t = 1$ to T **do**
  Selection:
    Randomly sample k sub-nets $\{x_1, \ldots, x_k\} \subset \mathcal{X}$.
    Evaluate objectives:

$$F_1(x_i) = \text{Acc}(x_i), \quad F_2(x_i) = \text{Resource}(x_i) \quad \forall x_i$$

  Update Pareto Front $\mathcal{P}$:
  for each sub-net x_i **do**
    if $F_2(x_i) \in [R_{\min}, R_{\max}]$ **then**
      Add x_i to $\mathcal{P}$ if non-dominated.
    end if
  end for
  Mutation:
    Randomly pick $x_i \in \mathcal{P}$, perturb its parameters:

$$x_i' = x_i + \epsilon, \quad \epsilon \sim \mathcal{R}andom$$

    Clip mutated W', H' to valid ranges.
  Crossover:
    Select $x_a, x_b \in \mathcal{P}$, perform crossover:

$$x_i'' = C_p(x_a, x_b),$$

    where the function C_p performs crossover on the H_S (height) and W_S (width) values of two sub-net slices with probability P, generating new sub-net parameters.
end for
Output: Pareto-optimal sub-nets $\mathcal{P}$.

IV. Experiment Results

A. Experimental Setup

All algorithmic experiments are conducted on NVIDIA A100 GPUs, with models and training procedures implemented using PyTorch. The experimental task was image classification using

the ImageNet-1K dataset [14],and the pre-trained model employed in our experiments is DAT [9]. During both the training and inference phases, the input image dimensions (W and H) are consistently set to 224×224 pixels. For our algorithmic experiments, we utilized the DAT-based model for both training and inference.

B. Fine-Tuning Performance of the Supernet

After the supernet is constructed, it undergoes five epochs of fine-tuning. The fine-tuning results are presented in Table I. Epoch 0 refers to the original accuracy of DAT without any fine-tuning. Through this process, our model achieves a 0.2% improvement in model accuracy.

TABLE I
TOP-1 ACCURACY DURING FINE-TUNING AFTER SLICING

Epoch of fine-tuning	Accuracy
0	84.6%
1	84.6%
2	84.7%
3	84.7%
4	84.7%

To investigate the impact of our hardware-friendly optimizations on model accuracy, we conduct an ablation study as detailed as shown in Table II. The results demonstrate that our optimizations result in only a 0.2% accuracy drop, while this marginal degradation could be effectively compensated through the aforementioned supernet fine-tuning procedure.

TABLE II
ABLATION EXPERIMENT RESULTS OF THE ALGORITHM

Method			Top-1
Slicing Strategy	Overlap Slicing	Fine-Tuning	Accuracy
✗	✗	✗	84.9%
✓	✗	✗	84.6%
✓	✓	✗	84.6%
✓	✓	✓	84.7%

C. Algorithmic Result of Optimal Slicing Strategy

After finishing the supernet fine-tuning, the proposed NAS-based method identifies the optimal slicing strategy for the given input image and model architecture. In our study, the input image size of the deformable attention layer is 56×56, and the slicing size is limited to less than or equal to 28. In addition, the slicing size should be greater than the 7×7 window size. Therefore, the values of H_S and W_S range between 8 and 28, and the best slicing values searched by our method are: $W_s = 14$ and $H_s = 28$.

We compared the top-1 accuracy of other works that also performed classification tasks on the ImageNet-1K dataset. Our method achieved an accuracy of 84.7%, outperforming similar approaches as shown in Table III.

TABLE III
TOP-1 ACCURACY COMPARISON WITH OTHER WORKS

Method	Dataset	Accuracy
MambaVision-B [15]	ImageNet-1K	84.1%
GroupMamba [16]	ImageNet-1K	83.9%
SpectFormer [17]	ImageNet-1K	80.21%
Our Work	ImageNet-1K	84.7%

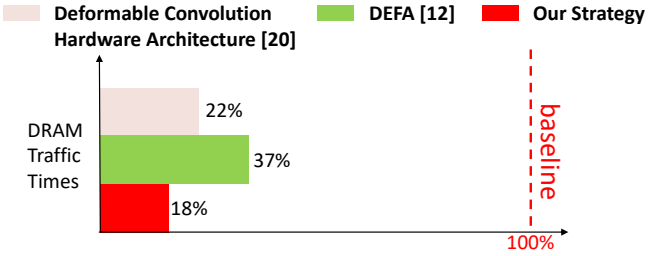


Fig. 4. Normalized DRAM traffic times comparisons.

D. Hardware Overhead Analysis

To verify the advantages of our framework in terms of hardware deployment, we implement the optimized DAT on the Xinlin FPGA platform using Vivado 2019.1 with verilog HDL, the hardware architecture is designed refer to [18]. For a fair comparison, we normalize the DRAM access times for the deformable attention/convolutional layer, following the approach proposed in [19], and normalize times for the methods of DEFA [12] and [20]. As shown in Fig. 4, our method decreases DRAM access times to 18% compared with baseline (layer by layer processing the sampling layer and the attention layer), significantly reducing bandwidth resources and power consumption. DEFA utilizes operator fusion and feature map reuse techniques to improve hardware efficiency. In addition, it develops multi-scale grid-sampling scheme(MSGS), reducing DRAM access times to nearly 37%. However, MSGS adopts the frequency-weighted pruning method to Optimize memory access, which added hardware overhead for masks and control logic. Our method proposes a hardware-friendly framework without modifying algorithm architecture, significantly reducing the hardware costs in terms of memory access and logical resources.

V. CONCLUSION

This paper presents a comprehensive framework for accelerating deformable attention mechanisms. By introducing a training-free slicing strategy, we effectively address the irregular memory access challenges due to data-dependent sampling, enabling parallel processing and reduced hardware resource consumption. Using a memory-aware NAS algorithm, our method automatically identifies optimal slice configurations that balance hardware efficiency and model performance. The experimental results demonstrate the dual advantages of our method in terms of algorithm accuracy and hardware efficiency.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, and S. a. Gelly, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.

[4] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & amp; distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 347–10 357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html

[5] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 108–126.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[7] S. Mehta and M. Rastegari, "MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] B. Graham *et al.*, "LeViT: a vision Transformer in convnet's clothing for faster inference," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 259–12 269.

[9] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4784–4793.

[10] H. Wang, Z. Zhang, and S. Han, "Spatten: Efficient sparse attention architecture with cascade token and head pruning," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2021, pp. 97–110.

[11] T. J. Ham, Y. Lee, S. H. Seo, S. Kim, H. Choi, S. J. Jung, and J. W. Lee, "Elsa: Hardware-software co-design for efficient, lightweight self-attention mechanism in neural networks," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, 2021, pp. 692–705.

[12] Y. Xu, D. Lyu, Z. Li, Z. Wang, Y. Chen, G. Wang, Z. Wang, H. Li, and G. He, "Defa: Efficient deformable attention acceleration via pruning-assisted grid-sampling and multi-scale parallel processing," *arXiv preprint arXiv:2403.10913*, 2024.

[13] D. Wang, C. Gong, M. Li, Q. Liu, and V. Chandra, "Alphanet: Improved training of supernets with alpha-divergence," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 760–10 771. [Online]. Available: https://proceedings.mlr.press/v139/wang21i.html

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "The imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: https://link.springer.com/article/10.1007/s11263-015-0816-y

[15] A. Hatamizadeh and J. Kautz, "Mambavision: A hybrid mamba-transformer vision backbone," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 25 261–25 270.

[16] A. Shaker, S. T. Wasim, S. Khan, J. Gall, and F. S. Khan, "Groupmamba: Efficient group-based visual state space model," in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, June 2025, pp. 14 912–14 922.

[17] B. N. Patro, V. P. Namboodiri, and V. S. Agneeswaran, "Spectformer: Frequency and attention is what you need in a vision transformer," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 9543–9554.

[18] Q. Zeng, Y. Wang, Z. Wang, and W. Mao, "An automated hardware design framework for various dnns based on chatgpt," in *2024 IEEE 37th International System-on-Chip Conference (SOCC)*, 2024, pp. 1–6.

[19] F. Tu, S. Yin, P. Ouyang, S. Tang, L. Liu, and S. Wei, "Deep convolutional neural network architecture with reconfigurable computation patterns," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 8, pp. 2220–2233, 2017.

[20] Y. Yu, J. Luo, W. Mao, and Z. Wang, "A memory-efficient hardware architecture for deformable convolutional networks," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 2021, pp. 140–145.