$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$: Adaptive Attention-Driven Dynamic Convolution for Local Feature Adaptation

Tianyu Zhang[a,1], Fan Wan[d,1], Xingyu Miao[a], Jingjing Deng[b,*], Xianghua Xie[c], Yang Long[a]

[a] *Computer Science Department, Durham University, The Palatine Centre, University, Stockton Rd, Durham, DH1 3LE, County Durham, United Kingdom*
[b] *Computer Science Department, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol, BS8 1UB, Bristol, United Kingdom*
[c] *Computer Science Department, Swansea University, Singleton Park, Swansea, SA2 8PP, Swansea, United Kingdom*
[d] *Central Research Institute, Tongfang Knowledge Network Digital Technology Co., Ltd., Beijing China*

Abstract

Dynamic convolution is an advanced deep-learning strategy that enables neural networks to adjust their convolutional kernels dynamically in response to varying input data. This adaptability enhances the network's efficiency in processing diverse features. However, traditional dynamic convolution techniques often overlook the critical role of local features in image classification, resulting in suboptimal performance in capturing fine details and textures necessary for accurate image analysis. To address this, our research introduces Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$), an innovative adaptive adjustment mechanism that focuses on local image features, significantly improving the network's ability to capture fine details and overall performance. Moreover, our paper proposes a novel dynamic convolution that enhances the network's feature learning ability by combining the input feature map with multiple convolution kernels to generate the attention weights. Additionally, we develop a streamlined version of our model, named $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$, which significantly increases operational efficiency and reduces computational costs. Experimental evaluations on the ImageNet, CIFAR-100 and COCO datasets demonstrate substantial performance enhancements, underscoring the efficacy and applicability of our approach.

*Corresponding author.
[1]These authors contributed equally to this work.

Keywords: Attention, Dynamic convolution, Local features.

1. Introduction

In recent years, deep learning has made substantial advancements, particularly in image processing [1, 2] and computer vision [3]. Among various methods, convolutional neural networks (CNNs) [4] have become fundamental due to their ability to process spatial hierarchies in images effectively. CNNs have revolutionized tasks such as image classification [5, 6], object detection [7, 8], and semantic segmentation [9, 10] by leveraging their hierarchical structure to learn increasingly complex features from raw pixel data. Traditionally, CNNs employ fixed weights during inference, which may not optimally handle diverse or dynamic input scenarios. This rigidity means that the same convolution kernel is applied to all input images regardless of changes in content. This limits the model's ability to adapt to subtle differences between various images, especially in real-world applications where data can be highly heterogeneous.

Dynamic convolution models have been introduced to address this limitation, offering unique advantages in enhancing CNN performance. These models leverage attention mechanisms [11, 12] to selectively focus on the more information-rich parts of the input by dynamically adjusting their convolution kernels based on the input, thereby improving the efficiency and accuracy of feature extraction. By dynamically adjusting their convolution kernels based on the input, they achieve desired results with minimal additional computational cost and enhance the model's representational capabilities. For instance, CondConv [13] employs conditionally parameterized convolutions to generate convolutional weights tailored to each input adaptively. Similarly, DynamicConv [14] integrates dynamic convolution with attention mechanisms to adjust kernel weights based on input features, improving flexibility and performance. Additionally, ODConv [15] explores the use of four-dimensional attention in dynamic convolution, further pushing the boundaries of performance.

Despite significant advancements, traditional dynamic convolution models [14, 16, 17] typically use a uniform approach for convolution kernel adaptation. This approach primarily focuses on the global characteristics of the input image, often struggling to

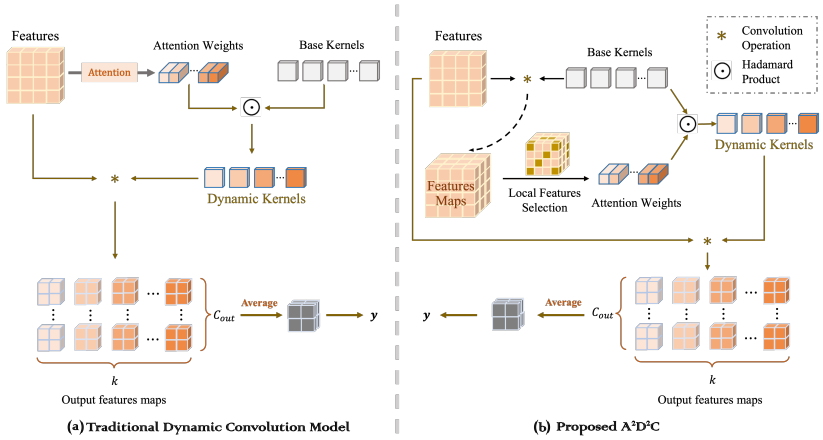

Figure 1: The figure compares the traditional dynamic convolution framework with our proposed Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2C$) framework.

leverage the rich local features inherent in images effectively. Local features are critical in many image-processing tasks, such as image recognition and classification, because they represent fine details and textures. These features provide resources for the model to learn subtle image differences. For example, details like texture, edges, and small shape variations in objects are conveyed through local features. If a network neglects to capture these details effectively, its performance on complex images will suffer.

Moreover, existing dynamic convolution techniques [13, 14, 15] struggle to inherently integrate the dynamics of the convolution kernel with the attention mechanism. In traditional dynamic convolution models, the attention mechanism is generated directly from the input image, resulting in attention weights not tightly integrated with multiple kernels. This disconnect limits the feature learning ability when processing complex features, particularly when dealing with diverse image data.

To address these challenges, we propose an Adaptive Attention-Driven Dynamic Convolution framework ($\mathcal{A}^2\mathcal{D}^2C$), as shown in Fig. 1. Our framework enhances the model by focusing on both global and local image features and integrating attention weights with the input feature map and multiple convolution kernels. To effectively capture subtle information from local features, we introduce a multi-point random sampling method within each mini-batch (MRS) that enables efficient local feature

extraction. This method involves randomly sampling multiple batches of points from the feature maps obtained through convolution and then averaging these points to capture local features. Furthermore, we enhance the network's feature learning ability by combining the input feature map with multiple convolution kernels to generate the attention weights. Specifically, we extract feature maps from the input image using multiple convolution kernels, which are then integrated to produce attention weights. By converting the feature maps into corresponding weights, our method allows finer adjustment of each kernel's response to different input features. This enables the model to capture both global and local characteristics of the input image, thereby improving its capability to handle complex and diverse image data.

While the proposed $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ effectively addresses several limitations of traditional dynamic convolution models, we identified certain complexities and redundancies within the model structure during our exploration. Specifically, the process of generating attention weights and dynamically adjusting convolution kernels introduces significant computational overhead. This includes the additional steps required to integrate and form new multiple convolution kernels, which increases memory usage and processing time. To address these inefficiencies, we develop a streamlined version of our model, named $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$, which simplifies our model by directly combining the calculated feature maps with attention weights, thereby reducing redundant computations and streamlining the architecture. By employing this approach, we significantly lower computational costs and complexity, thereby enhancing overall efficiency.

In summary, our contributions are as follows:

1. **Enhanced Local Feature Extraction:** We propose a novel framework that enhances the network's ability to capture details and improve overall performance by using random points extracted from feature maps.
2. **Adaptive Attention-Driven Enhancement:** We introduce a new attention mechanism that integrates attention with multiple base convolution kernels, significantly boosting the feature learning capability of the network.
3. **Optimized Model Architecture:** We present a method to streamline our model structure, reducing redundant computations and improving computation efficiency.

2. Related Works

2.1. Backbone for visual perception

Extracting local features or patches from images is fundamental to many computer vision tasks, such as object recognition, texture analysis, and scene understanding. Classic feature detection and description techniques like Scale Invariant Feature Transform (SIFT) [18] and Speeded-up Robust Features (SURF) [19] were pivotal in early developments. These methods identify and describe local features invariant to scaling and rotation changes. With the emergence of deep learning, convolutional neural networks (CNNs) [20] have become widely used for local feature extraction. Region-based CNN (R-CNN) and its variants (Fast R-CNN, Faster R-CNN, and Mask R-CNN) [21, 22, 23] have revolutionized object detection and instance segmentation by effectively extracting and processing regions of interest. Recently, attention mechanisms have been integrated into deep learning models to improve the specificity and relevance of extracted features. Models such as the Transformer Network [24] have significantly improved focus on relevant image patches, enhancing tasks like image classification and segmentation.

Recent ConvNet backbones improve local detail modeling while preserving convolutional efficiency, e.g., large-kernel/context-mixing designs (OverLoCK)[25] and efficient deformable convolutions (DCNv4) [26]. Our work leverages these advances to develop a dynamic convolution method that enhances the extraction and utilization of local image patches by integrating attention mechanisms and adaptive strategies. This approach achieves more detailed local feature extraction, balancing the need for regional information with the overall image context.

2.2. Dynamic Convolution Neural Networks

Building on the concept of local feature extraction, dynamic convolution neural networks have emerged to further enhance the adaptability and efficiency of CNNs. The concept of dynamic convolution was first proposed by Yang et al. in their 2019 work on CondConv [13]. Unlike static convolution, which applies a uniform kernel to all data, dynamic convolution employs different kernels for each image, conditioned on

the input. Chen et al. [14] introduced an attention mechanism into the kernel, dynamically integrating multiple convolutional kernels based on layer inputs, significantly enhancing the network's expressive power without increasing depth or width. Li et al. [15] further advanced this by proposing multi-dimensional integration, including kernel attention, output channel attention, input channel attention, and spatial attention. To reduce overhead, DCD [27] performs channel fusion in a low-dimensional space, improving efficiency with fewer parameters.

Recent studies further improve dynamic convolution via more expressive routing or frequency-aware filtering (e.g., OverLoCK [25]; Frequency Dynamic Convolution [28]). In contrast, we compute routing weights from random local evidence (MRS) and fuse k base kernels so inference still performs a single convolution, improving local adaptivity without increasing convolutional complexity.

2.3. Model Architecture Optimization

As dynamic convolution techniques advance, the need for model optimization becomes increasingly important, especially for deploying these models in real-world applications. Optimizing deep learning models is crucial for reducing computational demands and resource requirements, especially for deployment on resource-constrained devices and real-time applications. As introduced by Han et al. [29], network pruning involves removing redundant connections in a neural network, followed by quantization and compression techniques to reduce model size. Pruning can be performed either statically before training or dynamically during training. Quantization, demonstrated by Jacob et al. [30], reduces the precision of weights and activations, allowing models to be represented using lower-bit integers, thus decreasing model size and computational complexity with minimal accuracy loss. Knowledge distillation, proposed by Hinton et al. [31], involves training a smaller "student" model to replicate the behavior of a larger "teacher" model, achieving comparable performance with fewer parameters. Efficient network architectures like MobileNet [32] and EfficientNet [33] use depthwise separable convolutions and compound scaling to maintain high performance with fewer parameters and operations.

In comparison with these techniques, our work focuses on reducing redundant com-

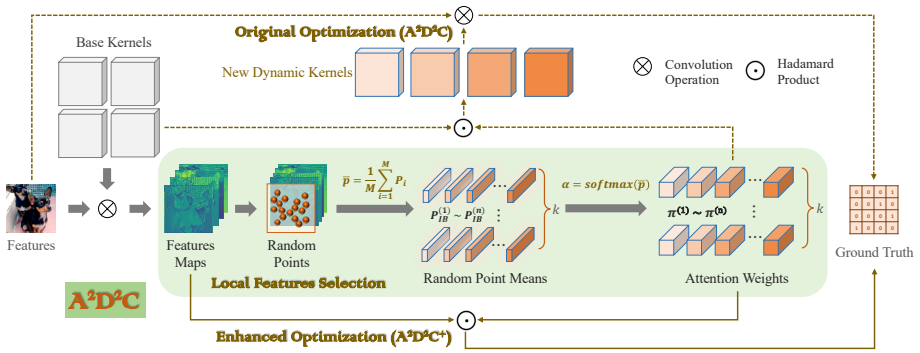

Figure 2: The main mechanism of our Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$) framework. The model extracts feature maps from the input, selects random local features, and generates adaptive attention weights. These weights modify base kernels to create dynamic kernels, enhancing feature learning. Both $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ and optimized $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ steps are shown, emphasizing improved feature extraction through adaptive kernel generation and local feature integration.

putations and streamlining the architecture of dynamic convolutional networks. This optimization minimizes computational costs and complexity while maintaining high performance, ensuring the models are lightweight and effective for real-time applications and resource-constrained devices.

3. Methodology

Our approach is motivated by the limitations of traditional dynamic convolution techniques. We first address these challenges and then review existing methods, laying the groundwork for our novel solution. Following this, we introduce a method for local feature extraction, which is crucial for improving image representation and processing. We then introduce Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$), which combines input feature maps and convolution kernels with multiple attention weights to enhance the adaptability and accuracy of the convolution process. Finally, we describe a streamlined model structure $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ to reduce computational redundancy and complexity. Fig. 2 illustrates the complete procedure for our approach, including the extraction of local features and the proposed contributions to dynamic convolution.

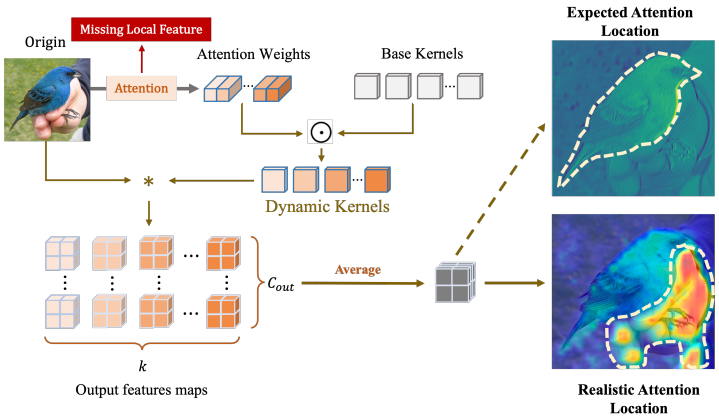

Figure 3: Illustration of the challenges in traditional dynamic convolution.

3.1. Motivation

Traditional dynamic convolution methods often convert the entire feature map directly into attention weights. While this approach helps capture the global characteristics of an image, it frequently neglects important local features that are crucial for accurate classification, particularly in complex images. For instance, as illustrated in Fig. 3, when processing an image of a hand holding a small bird, the primary classification focus should be on the bird. However, the attention mechanism in traditional methods unexpectedly prioritizes the hand, as these models are often guided by the dominant, more salient features in the image, which, in this case, are the hand's distinct shape and high contrast with the background. This prioritization occurs because traditional attention mechanisms tend to focus on prominent global patterns rather than subtle, context-dependent cues that are critical for distinguishing the bird from the hand. Consequently, this results in suboptimal performance in identifying the bird. In this paper, we address this issue by proposing a method that incorporates adaptive attention driven by both global and local feature information, enabling the model to better capture the essential details for accurate bird identification.

3.2. Enhanced Local Feature Extraction

It is crucial to capture the specific local information of images for tasks such as object detection, image segmentation, and image classification. This capability helps

the model to identify the nuanced differences between various images, thereby improving accuracy and robustness in these tasks. Existing static convolution techniques [34] apply the same convolution kernel across all input images, limiting their effectiveness in capturing diverse and intricate features of complex images. To address this issue, previous works [14, 15] have introduced dynamic convolution methods, such as DynamicConv proposed by Chen et al. [14]. These methods adjust kernel parameters based on the input image. The dynamic convolution can be expressed as:

$$y = (\alpha^x_{w_1} \odot \mathcal{W}_1 + \ldots + \alpha^x_{w_i} \odot \mathcal{W}_i + \ldots + \alpha^x_{w_n} \odot \mathcal{W}_n) * x, \tag{1}$$

where $\mathcal{W}_i$ represents the weight of the i-th convolutional kernel, and $\alpha^x_{w_i}$ is the corresponding attention value conditioned on x.

As shown in Equation (1), traditional dynamic convolution models adopt the attention mechanism $\alpha^x_{w_n}$ generated directly from the input image x, which enhances the model's adaptability to different features within the image, allowing for a more flexible and context-aware convolution operation. One of the most naïve methods of extracting local features is to select all points globally and operate average pooling to obtain the subtle information. However, conventional dynamic convolution methods for extracting local features tend to dilute important local variations by averaging out critical details, leading to a less precise image representation. These limitations can significantly impact the performance of image analysis and recognition systems, which rely on capturing fine-grained information.

To this end, we propose an approach that emphasizes a detailed analysis of feature maps to uncover these subtle details. The process begins with generating a feature map $\mathcal{F}$ from the input image $\mathcal{I}$ using a standard convolution process. This process is expressed as:

$$\mathcal{F} = \text{Conv}(\mathcal{I}, \mathcal{W}), \tag{2}$$

where $\mathcal{W}$ denotes the convolutional kernels. This initial feature map comprehensively represents the input, facilitating detailed subsequent analysis.

Considering the computational efficiency and the enhanced capability to capture and adapt to diverse local image features, we randomly select M points from this fea-

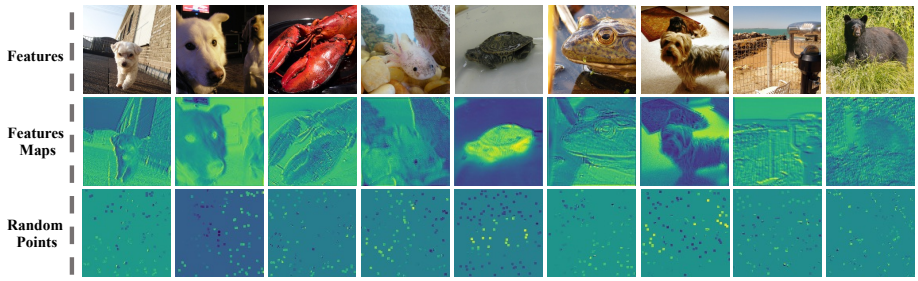

Figure 4: Illustration of the process for extracting local features using random points. The top row shows the original input images, the middle row displays the corresponding feature maps generated by the convolutional neural network, and the bottom row highlights the randomly sampled points used for capturing local features.

ture map $\mathcal{F}$, as illustrated in Fig. 4 for further calculation of weights. The discussion of selecting the optimal value of M can be seen in Table 9 and part 5.2 of the ablation study. This selection ensures that our analysis captures diverse features beyond the most conspicuous ones. Formally, multi-point random sampling (MRS) is defined as:

$$p_i \sim \mathcal{U}(\mathcal{F}), \tag{3}$$

where $\mathcal{U}(\mathcal{F})$ represents a uniform distribution over the spatial dimensions of the feature map $\mathcal{F}$, and p_i denotes the set of randomly selected points from this distribution.

Next, we compute the average of these selected points $\bar{p}$, and recognize it as the overall local feature characteristics:

$$\bar{p} = \frac{1}{M} \sum_{i=1}^{M} p_i, \tag{4}$$

where p_i denotes each selected point from the feature map, and M is the number of selected points. Let π denote the mixture weights over the k base kernels.

We then apply the softmax function to the average of the selected points to obtain the weights π:

$$\pi = \text{softmax}(\bar{p}), \tag{5}$$

where $\bar{p}$ represents the mean response over M sampled locations (MRS), serving as a compact local descriptor.

In summary, our approach to local feature extraction addresses the limitations of traditional methods by capturing diverse and intricate local features through random

sampling. This enhances the model's ability to process fine-grained information, leading to improved accuracy and robustness in image analysis tasks. A detailed discussion of these experiments is provided in Section 5.5 of the Ablation Study.

3.3. Adaptive Attention-Driven Dynamic Convolution

Traditional convolution often overlooks the intricate local details within images, leading to inadequate capture and processing of local image features. This oversight results in suboptimal utilization of the rich information embedded in the input image, particularly when dealing with complex and diverse data. Consequently, the convolution operations are not effectively integrated with multiple kernels, reducing the ability of the dynamic convolution to efficiently capture and use local features.

To address these limitations, we propose $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$. To capture local feature information of the images, we combine the generation of weights with the input feature map and multiple convolution kernels. Specifically, our approach begins by convolving the input image with k base kernels to produce a feature map. This step generates a diverse feature representation, allowing the model to capture various aspects of the image. This enhances the analysis of local features while preserving the overall image context. The average of the randomly selected points and the output of the Softmax function are used to generate a set of weights for each kernel. Finally, π_1 to π_k are created. Here π_k is the scalar coefficient for the k-th base kernel. Creating N such weight sets allows for a range of combinations and adaptability, closely reflecting the local features of the image. These weight sets are then combined with their corresponding kernels to generate new kernels, defined as:

$$\mathcal{K}_d = \pi_1^{(n)} \odot \mathcal{K}_{b1} + \pi_2^{(n)} \odot \mathcal{K}_{b2} + \ldots + \pi_k^{(n)} \odot \mathcal{K}_{bk}, \tag{6}$$

where $\mathcal{K}_d$ represents the newly weighted kernel, and $\mathcal{K}_{b1}$, $\mathcal{K}_{b2}$, and $\mathcal{K}_{bk}$ denote the set of base kernels, where $\mathcal{K}_{bk}$ is the k-th base kernel. Additionally, $\pi_1^{(n)}$, $\pi_2^{(n)}$, and $\pi_k^{(n)}$ are the weights assigned to the base kernels.

Finally, the dynamically adjusted kernels $\mathcal{K}_d$ are applied to the original feature map $\mathcal{F}$ to produce the final output:

$$\mathcal{Y} = \mathrm{Conv}(\mathcal{F}, \mathcal{K}_d), \tag{7}$$

where $\mathcal{Y}$ represents the output feature map that integrates the local feature details and the attention-adjusted convolution.

Local sampling and kernel fusion. For each base kernel $i \in \{1, \ldots, k\}$, we independently sample M locations $\{(u_{i,j}, v_{i,j})\}_{j=1}^{M} \sim \mathcal{U}(\{1, \ldots, H\} \times \{1, \ldots, W\})$ and form a per-kernel descriptor $\bar{p}_i = \frac{1}{M} \sum_{j=1}^{M} \mathcal{F}(:, u_{i,j}, v_{i,j}) \in \mathbb{R}^C$. A shared scalar router maps each descriptor to a logit $z_i = w^\top \bar{p}_i + b \in \mathbb{R}$, then stacking $\{z_i\}$ gives $z \in \mathbb{R}^k$ and $\boldsymbol{\pi} = \mathrm{softmax}(z)$ (over k). Given k base kernels $\{\mathcal{K}_{b,i}\}_{i=1}^{k}$ that share the same shape, stride, padding and dilation as the replaced static kernel (e.g., 3×3), we fuse them into a single effective kernel and apply one convolution:

$$\mathcal{K}_d = \sum_{i=1}^{k} \pi_i \odot \mathcal{K}_{b,i}, \qquad \mathcal{Y} = \mathrm{Conv}(\mathcal{F}, \mathcal{K}_d).$$

Backbone integration and base kernels. On ResNet-18/50 we replace the first 3×3 convolution in each residual block of stages 2–4; on MobileNetV2 we replace the 3×3 *depthwise* convolution in each inverted residual block (the 1×1 pointwise convolutions remain static). Unless otherwise noted we use k=4 and M=100.

Rationale and efficiency. We use multi-point random sampling (MRS) to summarize local evidence: rather than averaging all spatial locations as in global average pooling, we average features over M sampled locations (optionally response-weighted). This compact subset provides a reliable descriptor of local cues while emphasizing salient regions, thereby improving sensitivity to subtle patterns. The descriptor is mapped by a linear router and softmax to mixture weights $\boldsymbol{\pi}$, which linearly combine k base kernels into an effective kernel $\mathcal{K}_d = \sum_{i=1}^{k} \pi_i \odot \mathcal{K}_{b,i}$. Hence the model adapts to local content with a *single* convolution. The MRS estimator is unbiased and its error decays as $O(1/\sqrt{M})$, so a modest M (e.g., 100) suffices. Because the router is linear followed by a softmax, small descriptor perturbations induce proportionally small changes in $\boldsymbol{\pi}$. Unlike standard dynamic convolution that evaluates k experts and sums k outputs, we fuse the k base kernels once and perform one convolution; the routing overhead is negligible compared with the convolution. Compared with patch-wise attention (typically $O(N^2)$ in tokens), our scheme retains the $O(N)$ cost of one convolution while preserving content adaptivity, which explains the accuracy–efficiency of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$.

By leveraging multiple sets of weights, our dynamic convolution strategy provides an adaptive and detailed feature extraction method (outlined in Algorithm 1).

To conclude, our Adaptive Attention-Driven Dynamic Convolution method integrates attention mechanisms with convolutional kernels, enabling a more flexible and context-aware convolution process. This approach enhances the model's adaptability and accuracy in capturing and utilizing local features across diverse image data.

Algorithm 1 Training Procedure for adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$)

Require: Input image $\mathcal{I}$, base kernels $\{\mathcal{K}_{b,i}\}_{i=1}^{k}$, number of sampled points M, number of base kernels k, epochs T.

Ensure: The trained parameters for the dynamically adjusted kernels $\mathcal{K}_d$.

Initialize $\mathcal{K}_b$, set epoch $t = 1$.
while $t \leq T$ **do**
    Generate feature map $\mathcal{F}_\rangle$ from input image $\mathcal{I}$ using initial convolutional kernels:
        $\mathcal{F} = \text{Conv}(\mathcal{I}, \mathcal{K}_b)$
    for each $i = 1$ to k **do**
        Randomly select M points from $\mathcal{F}$:
            $\{p_j\}_{j=1}^{M} \sim \mathcal{U}(\mathcal{F})$
        Compute the average of the selected points:
            $\bar{p}_i = \frac{1}{M} \sum_{j=1}^{M} p_j$
        Apply softmax to the average to obtain attention weights:
            $\pi_i = \text{softmax}(\bar{p}_i)$
    end for
    Adjust the base kernels $\mathcal{K}_b$ using the attention weights π_i:
        $\mathcal{K}_d = \sum_{i=1}^{k} \pi_i \odot \mathcal{K}_{b,i}$
    Apply the dynamically adjusted kernels $\mathcal{K}_d$ to the feature map $\mathcal{F}$:
        $\mathcal{Y} = \text{Conv}(\mathcal{F}, \mathcal{K}_d)$
    Update $\mathcal{K}_b$ and other parameters using backpropagation and optimization techniques.
    $t := t + 1$;
end while

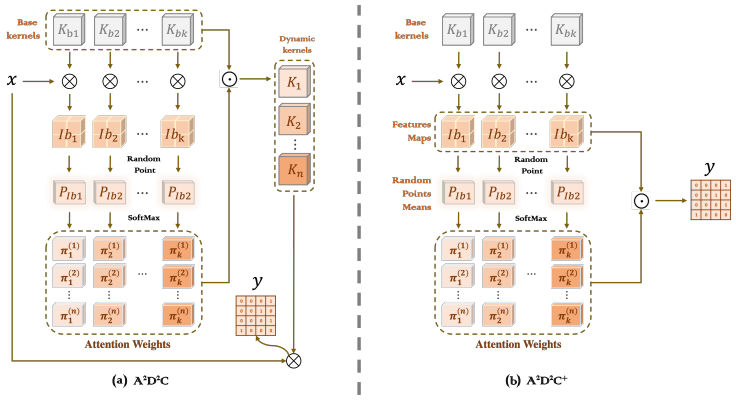

Figure 5: (a) Architecture of Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$). (b) Architecture of Adaptive Attention-Driven Dynamic Convolution Plus ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$).

3.4. $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$: Optimized Model Architecture

Although the proposed approach effectively addresses several limitations of traditional dynamic convolution models, complexities and redundancies are still significant issues. We identified certain complexities and redundancies within the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ model structure during our exploration. Specifically, our original optimization selects random points, and processes them with mean and Softmax functions to generate weights. Each group's weights are multiplied by corresponding feature maps $\mathcal{K}_{b1}$ to $\mathcal{K}_{bk}$ to create the final kernels y_1 to y_n. This step introduces complexity and redundancy, particularly because the previous convolution calculations involve feature maps that have already been processed, leading to unnecessary computational overhead.

To address these challenges, we further propose a simplified computational method, $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$, as illustrated in Fig. 5(b), we refine our method by directly multiplying the convolution results with their respective kernel weight values and then averaging these

weighted results. The detailed simplification process is expressed as follows:

$$
\begin{aligned}
y &= \left(\pi_1^{(n)} \odot \mathcal{K}_{b1} + \pi_2^{(n)} \odot \mathcal{K}_{b2} + \ldots + \pi_k^{(n)} \odot \mathcal{K}_{bk}\right) * x \\
&= \left(\sum_{i=1}^{k} \pi_i^{(n)} \odot \mathcal{K}_{bi}\right) * x \\
&= \sum_{i=1}^{k} (\mathcal{K}_{bi} * x) \odot \pi_i^{(n)} \\
&= x * \mathcal{K}_{b1} \odot \pi_1^{(n)} + x * \mathcal{K}_{b2} \odot \pi_2^{(n)} + \ldots + x * \mathcal{K}_{bk} \odot \pi_k^{(n)},
\end{aligned} \tag{8}
$$

where $\pi_1^{(n)}$, $\pi_2^{(n)}$, and $\pi_k^{(n)}$ represent the weights assigned to the base kernels, while $\mathcal{K}_{b1}$, $\mathcal{K}_{b2}$, and $\mathcal{K}_{bk}$ denote the base kernels, detailed can be shown in Algorithm 2.

As shown in Equation (8), we utilize the weights and feature maps calculated by multiple base convolution kernels to combine them separately, instead of recalculating new convolution kernels and new convolutions, thus simplifying the operation steps. This streamlined approach yields results identical to those of more complex methods while significantly reducing the computational burden.

Algorithm 2 Training Procedure for Adaptive Attention-Driven Dynamic Convolution Plus ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$)

Require: Input image $\mathcal{I}$, initial base kernels $\mathcal{K}_b$, number of random points M, number of weight groups k, number of epochs T.

Ensure: The trained parameters for the dynamically adjusted kernels $\mathcal{K}_d$.

Initialize $\mathcal{K}_b$, set epoch $t = 1$.
while $t \leq T$ **do**
    Generate feature map $\mathcal{F}$ from input image $\mathcal{I}$ using initial convolutional kernels:
        $\mathcal{F} = \text{Conv}(\mathcal{I}, \mathcal{K}_b)$
    for each $i = 1$ to k **do**
        Randomly select M points from $\mathcal{F}$:
            $\{p_i\}_{i=1}^{M} \sim \mathcal{U}(\mathcal{F})$
        Compute the average of the selected points:
            $\bar{p}_i = \frac{1}{M} \sum_{j=1}^{M} p_j$
        Apply softmax to the average to obtain attention weights:
            $\pi_i = \text{softmax}(\bar{p}_i)$
    end for
        $\mathcal{Y} = \sum_{i=1}^{k} (\pi_i \odot \mathcal{F})$
    Update $\mathcal{K}_b$ and other parameters using backpropagation and optimization techniques.
    $t := t + 1$;
end while

This streamlined method directly addresses the identified inefficiencies by reducing redundant computations and improving overall computational efficiency. By simplifying the dynamic convolution process, we maintain the model's performance while lowering operational costs and processing time, making it more suitable for real-time applications and deployment on resource-constrained devices.

In summary, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ optimization significantly reduces computational redundancy and complexity while consistently maintaining high performance. This streamlined and efficient architecture is particularly well-suited for real-time applications and deployment on resource-constrained devices, ensuring both operational efficiency and practical effectiveness across various scenarios.

4. Experiments and Results

4.1. Experiments setup

To evaluate the efficacy of the proposed Adaptive Attention-Driven Dynamic Convolution ($\mathcal{A}^2\mathcal{D}^2\mathcal{C}$), we conducted experiments on the CIFAR-100 [35], ImageNet [34] and COCO [36] datasets. For image classification, we use CIFAR-100 and ImageNet. The CIFAR-100 dataset includes 50,000 training images and 10,000 testing images, while the ILSVRC2012 comprises 1,281,167 training images across 1,000 categories and 50,000 validation images. The ImageNet presents a significantly more challenging benchmark than CIFAR-100 due to its larger dataset size, higher image resolution, greater number of categories, and increased image diversity. These characteristics necessitate more sophisticated model evaluation, providing a rigorous test for assessing model performance. For preprocessing, images were initially resized to 256x256 pixels, followed by a random crop to 224x224 pixels and random horizontal flipping. These steps align with standard augmentation practices for ImageNet training [34], ensuring consistency and fairness in comparison with existing methods. For object detection, we use the COCO dataset. The Common Objects in Context (COCO) dataset is widely recognized for its rich annotations, including object segmentation masks, bounding boxes, and keypoint detection, making it a comprehensive framework for evaluating object detection and segmentation algorithms. For our experiments, we utilized the 2017 version, which comprises 118,287 training images and 5,000 validation images, with annotations covering 80 object categories and approximately 1.5 million labeled instances. These annotations include segmentation masks for all labeled object instances, providing a robust benchmark for evaluating our proposed method.

Training recipe and baseline protocol. Unless otherwise noted, all classification models are trained from scratch for 100 epochs with SGD (momentum 0.9, weight decay 1×10^{-4}), global batch size 256, initial learning rate 0.0625, and a step schedule decaying by 0.1 at epochs 30/60/90. We use random resized crop to 224×224 and horizontal flip; no label smoothing, Mixup/CutMix, or EMA unless explicitly stated. The same recipe, augmentations, and training budget are applied to all convolutional baselines (DynamicConv, ODConv, DCD, etc.). ViT-Small and Swin-Tiny are trained

without pretraining (w/o P) under the same 100-epoch budget. For MobileNetV2 on ImageNet, we follow the common 150-epoch recipe (all methods under identical settings in this case). For fairness, we additionally report *DCD (params-matched†)*, where the network width is adjusted to match $\mathcal{A}^2\mathcal{D}^2C^+$'s parameter count within $\pm 1\%$.

4.2. Evaluation Metrics

For image classification, our primary evaluation metrics are top-1 and top-5 accuracy, measured on the ImageNet and CIFAR-100 validation set, which are standard benchmarks for assessing model performance. Additionally, we report the number of parameters and floating-point operations (FLOPs) to assess model efficiency.

For object detection, we used the widely adopted MMDetection toolkit [37], with pre-trained ResNet-50 models serving as the detector's backbones. Performance was assessed using the standard COCO metrics: Mean Average Precision (mAP) at different Intersections over Union (IoU) thresholds, specifically mAP@[0.5:0.05:0.95], which averages mAP calculated at IoU thresholds from 0.5 to 0.95 in steps of 0.05. Additional metrics included mAP@0.5 and mAP@0.75, as well as category-specific AP evaluations to determine model effectiveness across various object types.

4.3. Implementation Details

Backbones. For image classification we use ResNet18, ResNet50 and MobileNetV2 ($\times$0.5/$\times$0.75/$\times$1.0) as backbones; all are trained *from scratch*. For object detection we adopt Mask R-CNN with a ResNet-50–FPN backbone whose weights are ImageNet-pretrained.

Training setup (classification). SGD with momentum 0.9 and weight decay 1×10^{-4}, *global* batch size 256, 100 epochs, initial LR 0.0625 with step decay ($\times$0.1 at epochs 30/60/90), RandomResizedCrop to 224$\times$224 and horizontal flip. No label smoothing/Mixup/CutMix/EMA unless explicitly stated. All convolutional baselines (DynamicConv, ODConv, DCD, etc.) use the *same* recipe and budget. For MobileNetV2 on ImageNet we use a 150-epoch recipe under identical settings for all methods. Experiments run on NVIDIA A100 GPUs with Distributed Data Parallel (DDP).

$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ **hyperparameters.** Unless otherwise noted, each dynamic layer uses k=4 base kernels and M=100 sampled locations. The router temperature is initialized to 30.0 and linearly annealed during the first 10 epochs. MAdds/FLOPs are measured at 224×224 input; parameter counts include the classifier head.

Detection setup (COCO). We integrate $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ into the ResNet-50 backbone at three stages (kept consistent across variants). Training follows MMDetection defaults with a *1× schedule* (12 epochs) and the same optimizer (SGD, momentum 0.9, weight decay 1×10^{-4}). Images are resized with the shorter side = 800 and the longer side $\leq$ 1333; standard horizontal flip is applied. All detector variants (baseline and ours) share the same schedule, augmentations, and evaluation protocol.

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
ResNet-18 (static)	11.69M	1.81G	66.50	88.38
ViT-Small	22.12M	4.60G	66.54	88.49
DCNv4 [26]	12.22M	1.86G	69.42	89.53
InternImage [38]	25.39M	4.50G	69.52	89.55
CondConv	81.35M	1.89G	69.80	88.90
DynamicConv	45.47M	1.86G	70.40	89.79
DCD	14.70M	1.84G	71.41	91.68
ODConv (4×)	44.90M	1.92G	72.05	91.78
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	43.93M	1.93G	74.05±0.22	92.72±0.15
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	43.93M	1.93G	**74.49**±0.20	**92.93**±0.14

Table 1: CIFAR-100 validation with ResNet-18 (100 epochs, r = 0.1). Means±std are over 5 seeds for our methods; baselines are single runs. *Statistical testing:* Two-sided paired Student-t tests (df=4) against the strongest baseline in this table (ODConv 4×). $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$: 95% CI [74.00, 74.10], $p < 0.01$; $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$: 95% CI [74.24, 74.74], $p < 0.01$.

4.4. Image Classification

For the CIFAR-100 dataset, we use the ResNet-18 architecture as the backbone, Table 1 details the performance improvements achieved by the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ framework. Specifically, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4x) with 4 convolutional kernels achieves a top-1 accuracy

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
ResNet-50 (static)	25.58M	3.86G	68.10	89.29
ViT-Small	22.12M	4.60G	66.54	88.39
DCNv4	27.32M	4.65G	69.42	89.53
InternImage	25.39M	4.50G	69.52	89.55
CondConv	81.35M	3.98G	70.88	90.50
DynamicConv	45.47M	3.97G	71.45	91.29
DCD	14.70M	3.94G	72.85	92.65
ODConv (4×)	90.67M	4.08G	72.85	92.03
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	89.70M	4.08G	74.37 ± 0.20	92.73±0.16
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	89.70M	4.10G	**75.09 ± 0.22**	**93.03**±0.15

Table 2: Comparison of results on the CIFAR-100 validation set with ResNet-50 backbone trained for 100 epochs. The regularization parameter is set to $r = 0.1$. The best results are highlighted in bold. Means±std are over 5 seeds for our methods; baselines are single runs. *Statistical testing.* Two-sided paired Student-t tests (df=4) against the strongest baseline (ODConv 4×). $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$: 95% CI [74.12, 74.62], $p < 0.01$; $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$: 95% CI [74.82, 75.36], $p < 0.01$.

of 74.05% and a top-5 accuracy of 92.72%, with improvements of 7.55% and 4.34%. Furthermore, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4x) with 4 convolutional kernels achieves a top-1 accuracy of 74.49% and a top-5 accuracy of 92.93%, with a remarkable 7.99% increase in top-1 accuracy and a 4.55% improvement in top-5 accuracy compared to the traditional baseline. These results are consistent with those obtained in the ImageNet experiments, confirming the substantial accuracy gains on the CIFAR-100 dataset. Furthermore, using the ResNet-50 architecture as shown in Table 2, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4x) with 4 convolutional kernels achieves a top-1 accuracy of 74.37% and a top-5 accuracy of 92.73%, with improvements of 6.27% and 3.44%. Respectively, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4x) with 4 convolutional kernels achieves a top-1 accuracy of 75.09% and a top-5 accuracy of 93.03%, with improvements of 6.99% and 3.74%. These results significantly underscore the ability of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ to refine image classification accuracy. Collectively, these insights solidify the crucial role of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ in improving the discriminative efficiency of convolutional networks, underscoring its effectiveness in tackling advanced image recogni-

tion tasks. As shown in Table 3, our proposed $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ framework demonstrates significant improvements in classification accuracy across all configurations of MobileNetV2 on the CIFAR-100 dataset. For MobileNetV2 (x0.5) backbone, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 70.24% and a top-5 accuracy of 92.51%, which represents improvements of 0.43% and 1.99%, respectively, over the baseline configuration. For MobileNetV2 (x0.75) backbone, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 72.13% and a top-5 accuracy of 93.21%, which represents improvements of 1.65% and 0.80%. For MobileNetV2 (x1.0) backbone, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 72.86% and a top-5 accuracy of 93.37%, which represents improvements of 1.21% and 3.15%.

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
MobileNetV2 (0.5×)	2.00M	0.097G	69.81	90.52
DynamicConv	4.57M	0.103G	70.05	91.37
ODConv	4.44M	0.106G	70.21	91.95
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	3.32M	0.104G	70.18±0.18	92.38±0.12
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	3.32M	0.105G	**70.24**±0.17	**92.51**±0.12
MobileNetV2 (0.75×)	2.64M	0.209G	70.48	92.41
DynamicConv	7.95M	0.222G	71.75	92.53
ODConv	7.50M	0.229G	72.07	92.51
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	5.08M	0.224G	72.05±0.20	93.07±0.14
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	5.08M	0.226G	**72.13**±0.19	**93.21**±0.13
MobileNetV2 (1.0×)	3.50M	0.300G	71.65	90.22
DynamicConv	12.40M	0.319G	71.94	91.83
ODConv	11.51M	0.329G	72.85	92.83
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	10.21M	0.323G	72.78±0.19	93.26±0.13
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	10.21M	0.325G	**72.86**±0.18	**93.37**±0.14

Table 3: Comparison of results on the CIFAR-100 validation set with MobileNetV2 backbones trained for 100 epochs. The regularization parameter is set to $r = 0.1$. The best results are highlighted in bold. Means±std are over 5 seeds for our methods; baselines are single runs.

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
ResNet-18 (static)	11.7M	1.81G	69.57	89.24
ViT-Small (w/o P)	22.1M	4.60G	71.60	90.10
CondConv	89.9M	1.89G	71.99	90.27
DynamicConv	45.5M	1.86G	72.76	90.79
DCD	14.7M	1.84G	72.33	90.65
DCD (params-matched†)	44.9M	1.93G	72.40	91.10
ODConv (4×)	44.9M	1.92G	73.25	91.07
Swin-Tiny (w/o P)	28.3M	4.50G	73.30	91.20
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	44.9M	1.93G	73.33±0.16	91.15±0.12
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	44.9M	1.93G	**73.47**±0.18	**92.72**±0.10

Table 4: ImageNet validation with ResNet-18 backbone (100 epochs, r=0.0625). Means±std are over 5 seeds for our methods; baselines are single runs. Params include the classifier head. MAdds are measured at 224×224 and approximate FLOPs. *w/o P* means trained from scratch without pretraining. † Re-implemented DCD with width adjusted to match $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$'s parameter count (±1%); same training budget.

For the ImageNet dataset, on the ResNet-18 architecture, as shown in Table 4, our proposed model demonstrates a significant improvement in classification accuracy on the ImageNet validation dataset compared to general convolution. Specifically, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 73.47% and a top-5 accuracy of 92.72%, representing a 3.90% increase in top-1 accuracy and a 3.48% increase in top-5 accuracy compared to the baseline configuration.

On the ResNet-50 architecture, as illustrated in Table 5, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 78.86% and a top-5 accuracy of 93.67%, with improvements of 3.56% and 1.47%, respectively, over the baseline configuration using four base kernels. Similar to the results obtained with ResNet-18, adapting our model to ResNet-50 further substantiates the substantial accuracy gains on the ImageNet dataset.

To provide a comprehensive comparison, we also evaluate Transformer-based models such as ViT-Small and Swin-Tiny. These models typically rely on extensive pretraining on large-scale datasets to achieve competitive performance. In our experiments, we assess them under non-pretrained conditions. Notably, our method outper-

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
ResNet50 (static)	25.6M	3.86G	75.30	92.20
ViT-Small (w/o P)	22.1M	4.60G	71.60	90.10
Swin-Tiny (w/o P)	28.3M	4.50G	73.30	91.20
CondConv	189.9M	3.98G	76.70	93.12
DynamicConv	100.9M	3.97G	76.82	93.16
DCD	29.8M	3.94G	76.92	93.46
DCD (params-matched†)	89.7M	4.08G	77.10	93.35
DCNv4 (w/o P)	27.3M	4.65G	77.65	93.48
ODConv (4×)	90.7M	4.08G	78.32	93.56
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	89.7M	4.08G	78.56±0.22	93.59±0.12
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	89.7M	4.10G	**78.86**±0.21	**93.67**±0.13

Table 5: ImageNet validation with ResNet-50 backbone (100 epochs, r=0.0625). Means±std are over 5 seeds for our methods; baselines are single runs. Params include the classifier head. MAdds are measured at 224×224 and approximate FLOPs. *w/o P* means trained from scratch without pretraining. † Re-implemented DCD with width adjusted to match $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$'s parameter count (±1%); same training budget.

forms the non-pretrained versions of ViT-Small and Swin-Tiny, which underscores the strength of our convolutional approach in scenarios where pretraining is not feasible. However, we acknowledge that when large-scale pretraining is applied, Transformer-based architectures can achieve even higher accuracy.

As shown in Table 6, on the MobileNetV2 (x0.5) backbone, the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model with 4 convolutional kernels achieves a top-1 accuracy of 70.25% and a top-5 accuracy of 89.20%, with improvements of 5.95% and 3.92%, respectively, over the baseline configuration using four base kernels. Adapting our model to MobileNetV2 (x0.5) further confirms significant accuracy gains on the ImageNet dataset. These results demonstrate the effectiveness of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ in improving image classification accuracy and underscore its ability to enhance the discriminative capacity of convolutional networks for advanced image recognition tasks.

All these results indicate that the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ framework effectively enhances the performance of different backbones on the ImageNet and CIFAR-100 dataset, providing

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
MobileNetV2 (0.5×)	2.00M	97.1M	64.30	85.21
CondConv	13.61M	110.0M	67.24	87.51
DynamicConv	4.57M	103.2M	69.05	88.37
DCD	3.06M	105.6M	69.32	88.44
ODConv (4×)	4.44M	106.4M	70.01	89.01
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4x)	4.05M	105.2M	70.22± 0.18	**89.20± 0.12**
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4x)	4.05M	105.2M	**70.25± 0.17**	**89.20± 0.11**

Table 6: Results comparison on the ImageNet validation set with the MobileNetV2 backbones trained for 150 epochs. We set r = 0.0625. The best results are bolded. Means±std are over 5 seeds for our methods; baselines are single runs.

a notable increase in classification accuracy. The improvements are consistent across different width multipliers, showcasing the robustness and scalability of the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ approach.

4.5. Comparison with pretrained transformers

As shown in Table 7, we compare $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ and $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ against pretrained ViT-Small and Swin-Tiny. All models are fine-tuned under the same budget on a fixed 25% ImageNet-1K subset (class-balanced sampling). CNN backbones (static and ours) are initialized from ImageNet-1K pretrained weights and trained with SGD; ViT/Swin use AdamW with standard hyperparameters. We report MAdds and Top-1 on the ImageNet-1K val set; robustness is assessed on ImageNet-V2 without test-time augmentation.

To further demonstrate the effectiveness of our $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ model, we visualize Grad-CAM++ results for different baselines and our models using ResNet-18, as shown in Figure 6. To ensure fairness, we use identical preprocessing, target class, colormap, and normalization. Across diverse scenes, bird-in-hand, turtle in a bowl, and indoor pets, the baseline frequently fires on contextual cues (e.g., the hand, specular highlights, background person), whereas $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ shifts the response toward object-centric regions (bird head and claws, turtle carapace patterns, dog muzzle and eyes) with

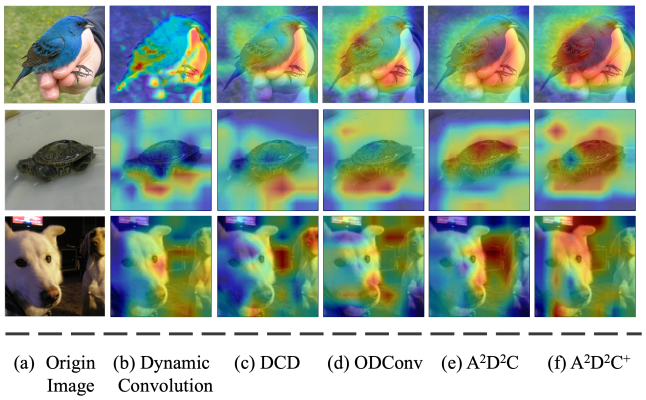

Figure 6: Grad-CAM++ visualization results for baselines and our models on ImageNet. (a) Original Images, (b) Dynamic Convolution, (c) DCD, (d) ODConv, (e) $\mathcal{A}^2\mathcal{D}^2C$ and (f) $\mathcal{A}^2\mathcal{D}^2C^+$.

Model (pretrained)	Params (M)	MAdds (G)	IN-1K 25% Top-1 (%)	IN-V2 Top-1 (%)
ViT-Small (pt)	22.1	4.60	73.1	71.0
Swin-Tiny (pt)	28.3	4.50	74.2	72.1
ResNet-50 (pt, static)	25.6	3.86	73.8	71.8
$\mathcal{A}^2\mathcal{D}^2C$ (pt, 4×)	89.7	4.08	75.0	72.7
$\mathcal{A}^2\mathcal{D}^2C^+$ (pt, 4×)	89.7	4.10	**75.4**	**73.0**

Table 7: Pretrained comparison under a 25% ImageNet-1K fine-tuning budget and robustness on ImageNet-V2.

tighter, less diffused heatmaps. This consistent relocation of attention supports our claim that $\mathcal{A}^2\mathcal{D}^2C^+$ better preserves local discriminative features while suppressing spurious background evidence.

4.6. Object Detection

In Table 8, we present the performance of various models on the MS-COCO 2017 validation set using Mask R-CNN. The results demonstrate that our proposed method, $\mathcal{A}^2\mathcal{D}^2C^+$ (4×), achieves superior performance across multiple evaluation metrics. Specifically, $\mathcal{A}^2\mathcal{D}^2C^+$ (4×) achieves an Average Precision (AP) of 41.2%, AP_{50} of 62.4%, AP_{75} of 44.3%, AP_S of 24.9%, AP_M of 44.5%, and AP_L of 53.1%, outperforming all existing methods.

These results highlight the effectiveness of the $\mathcal{A}^2\mathcal{D}^2C^+$ framework in enhancing

Backbone Models	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_S(%)	AP_M(%)	AP_L(%)	Params	MAdds
ResNet50	38.0	58.6	41.5	21.6	41.5	49.2	46.45M (23.51M)	260.14G (76.50G)
CondConv (8×)	38.8	59.3	42.3	22.5	42.5	50.3	136.4M (113.46M)	260.15G (76.51G)
DynamicConv (4×)	39.2	60.3	42.5	23.0	42.9	51.4	121.77M (98.83M)	260.30G (76.66G)
DCD	38.8	59.8	42.2	23.1	42.7	49.8	50.73M (27.79M)	260.27G (76.63G)
Swin-tiny	39.3	60.7	42	23.1	42.9	52.5	47.80M (25.23M)	264.32G (77.10G)
ODConv (1×)	39.9	61.2	43.5	23.6	43.8	52.3	49.53M (26.59M)	260.25G (76.61G)
ODConv (4×)	40.1	61.5	43.6	24.0	43.6	52.3	111.56M (88.62M)	260.49G (76.85G)
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ (4×)	40.9	62.0	44.0	24.6	44.2	52.8	110.24M (87.95M)	260.18G (76.54G)
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (4×)	**41.2**	**62.4**	**44.3**	**24.9**	**44.5**	**53.1**	110.24M (87.95M)	260.16G (76.52G)

Table 8: Results comparison on the MS-COCO 2017 validation set on Mask R-CNN. Regarding parameters or MAdds, the number in the bracket is for the pre-trained backbone models excluding the last fully connected layer, while the other number is for the whole object detector. The best results are bolded.

feature representation and improving object detection accuracy. Notably, our method achieves substantial improvements across objects of different scales, particularly for small and medium-sized objects (AP_S and AP_M). This demonstrates the robustness of our approach in capturing fine-grained details and local features, leading to superior object localization and recognition capabilities.

5. Ablation Study

We conducted several ablation experiments on the ImageNet and CIFAR-100 datasets to evaluate the performance of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ and $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$.

5.1. The impact of the selection of M points.

In this ablation study, we evaluate the impact of varying the number of random points on the accuracy of our model. We experiment with four configurations where the number of random points M is set to 50, 100, 150, and 200. As observed in Table 9, increasing the number of random points generally leads to a higher accuracy. Specifically, when $M = 50$, the model achieves an accuracy of 73.65%. Increasing M to 150 and 200 results in accuracies of 73.39% and 74.08% respectively. The best performance is seen when $M = 100$, with an accuracy of 74.49%. Further analysis suggests that selecting $M = 100$ as the optimal value may be due to the model's ability

Random Points	Params	MAdds	Top-1 (%)	Top-5 (%)
M=50	44.93M	1.93G	73.65	92.37
M=100	44.93M	1.93G	74.49	92.93
M=150	44.93M	1.93G	73.39	92.32
M=200	44.93M	1.93G	74.08	92.53

Table 9: Results comparison of the implementation of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ at the initial of different selection of M points number of on the Cifar100 with the ResNet18 backbones trained for 100 epochs.

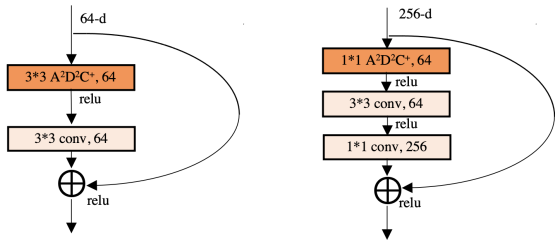

Figure 7: Results comparison of the implementation of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ at the initial of ResNet blocks on the ImageNet validation set with the ResNet18 & ResNet50 backbones.

to effectively capture data features while avoiding potential overfitting associated with too many random points. Therefore, this configuration enhances feature representation without compromising the model's generalization and stability.

5.2. The impact of different block positions of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$.

As shown in Fig. 7, we use $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ in the initial of each block in ResNet18 and ResNet50 to replace ordinary convolution. We found that when the number of kernels is unified to 8, the replaced ResNet18 has a Top-1 accuracy of 72.20% and a Top-5 accuracy of 90.65%, while the original ResNet18 only has a 69.57% and a Top-5 accuracy of 89.24% (see Table 10). Also in ResNet50, the replaced ResNet50 has a Top-1 accuracy of 76.71% and a Top-5 accuracy of 93.72%, while the original ResNet50 only has a 75.30% and a Top-5 accuracy of 92.20%. These results suggest that replacing initial convolutional layers with $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ allows the network to capture and enhance local features early, establishing a stronger foundation for subsequent layers and improving overall performance. The consistent gains across both ResNet18 and ResNet50 high-

Model	Params	MAdds	Top-1 (%)	Top-5 (%)
ResNet18	11.69M	1.814G	69.57	89.24
+$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (8x, Initial)	44.78M	1.920G	72.20(↑2.63)	90.65(↑1.41)
ResNet50 [39]	25.56M	3.858G	75.30	92.20
+$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (8x, Initial)	55.95 M	3.900G	76.71(↑1.41)	93.72(↑0.52)

Table 10: Results comparison of the implementation of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ at the initial of each block on the ImageNet validation set with the ResNet18 and ResNet50 backbones trained for 100 epochs. We set r = 0.0625.

Kernel	Params	Top-1 (%)	Top-5 (%)	Time Cost (s)
K=4	44.93M	74.05	92.72	140.56
K=8	88.87M	74.08	92.78	187.43
K=12	132.82M	74.24	92.91	237.50
K=16	176.76M	74.32	92.92	293.78

Table 11: Comparison Result of Different Kernel Number of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ on CIFAR-100

light the robustness and scalability of the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ framework. Strategically placing $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ at the initial stages of ResNet blocks significantly enhances classification accuracy, offering valuable guidance for integrating dynamic convolution techniques in deep learning architectures.

5.3. The impact of Convolution Kernel Number.

We conduct experiments on the classification Top-1 accuracy with different numbers of convolution kernels, as shown in Table 11. We find that when the kernel number is 16, the Top-1 accuracy can reach a maximum of 74.32%, and the Top-5 accuracy is 92.92%. The analysis suggests that while increasing the number of convolution kernels can enhance the network's performance by allowing it to capture more detailed features, it also significantly raises the computational burden. Therefore, a balance must be struck between accuracy and computational efficiency based on the specific application requirements. This insight is crucial for optimizing the implementation of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ in real-world scenarios where computational resources may be limited.

5.4. *Effect of Local Feature Extraction*

We evaluate the contribution of local sampling (LS) under compute- and parameter-matched controls to disentangle its effect from model size. All variants use ResNet-18 at 224×224 with the same training budget and augmentations; we report mean±std over 5 seeds (p-values from paired t-tests across seeds). In addition to the baseline and the original *w/o LS* and *fixed sampling* (FS) variants, we introduce: (i) FLOPs-matched w/o LS, where a lightweight 1×1 projection adjusts MAdds to match the LS model within 0.5%; (ii) Params-matched w/o LS, with the same parameter count as the LS model; and (iii) Random-LS and Uniform-LS, which keep compute identical to LS but remove informative sampling (same M and router).

Model	**Params (M)**	**MAdds (G)**	**Top-1 (%)**	**Δ**	***p*-value**	**Top-5 (%)**
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (w/o LS)	43.21	1.88	71.49±0.21	−0.53	0.002	90.72
FLOPs-matched w/o LS	43.74	**1.93**	72.02±0.20	–	–	91.80
Params-matched w/o LS	**44.93**	1.91	71.84±0.21	−0.18	0.210	91.65
Random-LS (same compute as LS)	44.93	**1.93**	71.81±0.19	−0.21	0.006	91.60
Uniform-LS (same compute as LS)	44.93	**1.93**	71.57±0.22	−0.45	0.008	91.42
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (FS)	43.90	1.92	71.71±0.22	−0.31	0.011	91.12
$\mathcal{A}^2\mathcal{D}^2\mathcal{C}^+$ (LS, MRS)	44.93	**1.93**	**73.47**±0.18	**+1.45**	0.004	92.72

Table 12: Local sampling (LS) vs. compute/parameter–matched controls on ImageNet with ResNet-18. Mean±std over 5 seeds; Δ is relative to *FLOPs-matched w/o LS*. *MRS*: Multi-point Random Sampling; *w/o LS*: without Local Sampling; *FS*: Fixed Sampling; p-values are from paired *t*-tests across seeds.

Removing LS reduces Top-1 by 1.98 points (73.47 → 71.49). Under compute- and parameter-matched controls, the LS model still exceeds the best non-LS control by +1.45 Top-1 (paired t-test, $p < 0.01$), indicating that the gain is not attributable to reduced compute or parameters. *Random-LS* and *Uniform-LS*, which keep identical compute (same M and router) while removing informative sampling, also underperform, confirming that *where* we sample matters. The LS overhead is modest (1.93G vs. 1.88G MAdds) relative to the observed gain.

6. Limitations & Failure Cases

While the $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ framework has shown significant improvements on standard datasets, several limitations remain. One major challenge of the framework is the hyperparameter tuning process. Although fine-tuning parameters such as the number of convolution kernels, learning rates, and batch sizes are essential for optimizing performance, it requires extensive experimentation that is both time-consuming and computationally expensive. This intensive tuning process may hinder the practical deployment of the framework in environments with limited computational resources or where rapid implementation is required. Another limitation lies in its scalability to larger datasets and more complex tasks. The model's performance on these larger-scale and more intricate datasets has not yet been fully validated, raising concerns about its ability to maintain efficiency and accuracy under such conditions. In addition, we observe reduced gains or failures on (i) low-contrast or small objects that require fine localization, (ii) aquatic or glass-enclosed scenes with color cast and specular highlights (domain shift), and (iii) shape-dependent classes where global silhouette is more informative than local textures. Addressing these issues will be essential for ensuring the broader applicability of $\mathcal{A}^2\mathcal{D}^2\mathcal{C}$ in diverse real-world scenarios.

Fig. 8 shows a representative failure where the ground-truth class is newt, yet the model predicts coral reef. The Grad-CAM++ map concentrates on the highlight and rock region instead of the animal, indicating a background-driven decision. We diagnose three interacting factors. (i) Target-background low contrast and soft, translucent boundaries: the salamander shares color with pebbles and tank water, making local cues weak. (ii) Limited spatial resolution of the last stage: with a large stride, the final feature map is coarse, which yields sparse or shifted CAM responses and degrades localization. (iii) Context bias in aquarium environments: strong scene priors (rocks, reef-like textures) bias the classifier toward fish or coral-related labels. Similar behavior is also observed in other aquatic samples in our validation set.

(a) Input image (GT: *newt*)

(b) Grad-CAM++ heatmap (Pred: *coral reef*)

Figure 8: Background-driven mis-localization in an aquatic scene. The heatmap concentrates on the specular region rather than the animal, suggesting context bias under low target-background contrast and translucent boundaries, compounded by the coarse resolution of the final feature map.

7. Conclusion

In this paper, we proposed a significant advancement in dynamic convolution techniques by integrating attention mechanisms with convolution kernels, offering a novel approach that surpasses traditional methods in both adaptability and effectiveness. By introducing an innovative adaptive adjustment mechanism tailored to local image characteristics, we have substantially enhanced the network's capability to capture and process local features. Comprehensive evaluations of the ImageNet and CIFAR-100 datasets have demonstrated the superior performance of our approach, particularly in tasks requiring detailed feature analysis and representation. The insights gained from this study underscore the importance of dynamic convolution and attention mechanisms in enhancing the discriminative power of convolutional neural networks, offering promising directions for future research and applications in deep learning. Future work could explore the application of our approach on larger datasets to validate further and enhance its effectiveness and scalability.

References

[1] T. Acharya, A. K. Ray, Image processing: principles and applications, John Wiley & Sons, 2005.

[2] R. Gao, F. Wan, D. Organisciak, J. Pu, H. Duan, P. Zhang, X. Hou, Y. Long, Privacy-enhanced zero-shot learning via data-free knowledge transfer, in: 2023

IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2023, pp. 432–437.

[3] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Computational intelligence and neuroscience 2018 (2018) 7068349.

[4] Y. Quan, Y. Chen, Y. Shao, H. Teng, Y. Xu, H. Ji, Image denoising using complex-valued deep cnn, Pattern Recognition 111 (2021) 107639.

[5] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural computation 29 (2017) 2352–2449.

[6] A. M. Obeso, J. Benois-Pineau, M. S. G. Vázquez, A. Á. R. Acosta, Visual vs internal attention mechanisms in deep neural networks for image classification and object detection, Pattern Recognition 123 (2022) 108411.

[7] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE transactions on neural networks and learning systems 30 (2019) 3212–3232.

[8] V. Chalavadi, P. Jeripothula, R. Datla, S. B. Ch, et al., msodanet: A network for multi-scale object detection in aerial images using hierarchical dilated convolutions, Pattern Recognition 126 (2022) 108548.

[9] Y. Guo, Y. Liu, T. Georgiou, M. S. Lew, A review of semantic segmentation using deep neural networks, International journal of multimedia information retrieval 7 (2018) 87–93.

[10] Q. Zhou, X. Wu, S. Zhang, B. Kang, Z. Ge, L. J. Latecki, Contextual ensemble network for semantic segmentation, Pattern Recognition 122 (2022) 108290.

[11] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[12] S. Tang, T. Lu, X. Liu, H. Zhou, Y. Zhang, Catnet: Convolutional attention and transformer for monocular depth estimation, Pattern Recognition 145 (2024) 109982.

[13] B. Yang, G. Bender, Q. V. Le, J. Ngiam, Condconv: Conditionally parameterized convolutions for efficient inference, Advances in neural information processing systems 32 (2019).

[14] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, Z. Liu, Dynamic convolution: Attention over convolution kernels, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11030–11039.

[15] C. Li, A. Zhou, A. Yao, Omni-dimensional dynamic convolution, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=DmpCfq6Mg39.

[16] T. Zhang, F. Wan, K. Sun, X. Miao, Y. Sun, M. Shao, Y. Long, Dynamic convolution and graph-coupled attention for cross-subject eeg-vision decoding, in: 36th British Machine Vision Conference, 2025, pp. 24–27.

[17] T. Zhang, F. Wan, H. Duan, K. W. Tong, J. Deng, Y. Long, Fmdconv: Fast multi-attention dynamic convolution via speed-accuracy trade-off, Knowledge-Based Systems 317 (2025) 113393.

[18] T. Lindeberg, Scale invariant feature transform, 2012.

[19] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9, Springer, 2006, pp. 404–417.

[20] K. O'Shea, R. Nash, An introduction to convolutional neural networks, arXiv preprint arXiv:1511.08458 (2015).

[21] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[22] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[23] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[24] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, M. Shah, Transformers in vision: A survey, ACM computing surveys (CSUR) 54 (2022) 1–41.

[25] M. Lou, Y. Yu, Overlock: An overview-first-look-closely-next convnet with context-mixing dynamic kernels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025, pp. 128–138.

[26] Y. Xiong, Z. Li, Y. Chen, F. Wang, X. Zhu, J. Luo, W. Wang, T. Lu, H. Li, Y. Qiao, et al., Efficient deformable convnets: Rethinking dynamic and sparse operator for vision applications, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 5652–5661.

[27] Y. Li, Y. Chen, X. Dai, M. Liu, D. Chen, Y. Yu, L. Yuan, Z. Liu, M. Chen, N. Vasconcelos, Revisiting dynamic convolution via matrix decomposition, arXiv preprint arXiv:2103.08756 (2021).

[28] L. Chen, L. Gu, L. Li, C. Yan, Y. Fu, Frequency dynamic convolution for dense image prediction, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 30178–30188.

[29] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149 (2015).

[30] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2704–2713.

[31] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).

[32] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[33] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[35] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Technical Report, University of Toronto, 2009.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[37] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al., Mmdetection: Open mmlab detection toolbox and benchmark, arXiv preprint arXiv:1906.07155 (2019).

[38] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al., Internimage: Exploring large-scale vision foundation models with deformable convolutions, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 14408–14419.

[39] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of tricks for image classification with convolutional neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 558–567.