Frequency Dynamic Convolution for Dense Image Prediction

Supplementary Material

Linwei Chen[1] Lin Gu[2,3] Liang Li[4] Chenggang Yan[5,6] Ying Fu[1*]
[1]Beijing Institute of Technology [2]RIKEN [3]The University of Tokyo
[4]Chinese Academy of Sciences [5]Hangzhou Dianzi University [6]Tsinghua University
chenlinwei@bit.edu.cn; lin.gu@riken.jp; liang.li@ict.ac.cn; cgyan@hdu.edu.cn; fuying@bit.edu.cn

This supplementary material provides additional details and results that could not be included in the main paper due to space constraints. The content is organized as follows:

- Section A analyzes weight similarity across methods, and offers mathematical analysis for FDConv.
- Section B provides a frequency-domain analysis of the learned parallel weights from prior works and our proposed FDConv.
- Section C includes additional t-SNE visualization results.
- Section D presents more qualitative feature visualizations.
- Section E reports detailed ablation studies for FDConv.
- Section F outlines further training details for various challenging datasets.
- Section G explains how Frequency Disjoint Weights reduce parameter usage by leveraging the Hermitian symmetry property.
- Section H describes an equivalent implementation of frequency band modulation.

A. Weight Similarity Analysis

In this section, we analyze the diversity of learned weights across different methods to demonstrate the superiority of our FDConv. The results, visualized in Figures 1, 2, and 3, highlight the differences in weight similarity between FDConv and existing approaches [6, 7, 16, 18].

Comparison with ODConv. Figure 1 (top row) presents the cosine similarity between the four learned weights from prior dynamic convolution methods [6, 7, 16, 18]. We present representative results from stages 1-4 of the models [5]. For a fair comparison, the number of weights in FDConv was set to 4, consistent with the configurations of the compared methods.

These methods exhibit very high similarity among their weights (*e.g.*, $>$0.94 at stage 2 for ODConv), indicating a lack of diversity. This limited diversity constrains the adaptability of the models and their ability to effectively capture a broad spectrum of features.

The bottom row of Figure 1 demonstrates the weight similarity for FDConv. In stark contrast to previous methods, each weight in FDConv shows a cosine similarity of 1.0 only with itself and 0 with others. This result confirms the high diversity of the learned weights, enabling FDConv to better adapt to varying input patterns and capture features across different frequency bands.

Comparison with Latest KW. We analyze the weight similarity in KW [6], as illustrated in Figure 2. KW employs a weight-sharing strategy across layers, where an attention mechanism linearly combines weights from a shared weight warehouse to generate the final weights. Consequently, the attention module plays a crucial role in determining weight diversity. Our analysis indicates that the learned attention weights in KW exhibit high similarity across layers, resulting in similar attention values for weight mixing. This, in turn, leads to a high similarity in the final combined weights. In contrast, as shown in the bottom part of Figure 1, our FDConv demonstrates zero similarity between weights, highlighting its superior diversity.

Impact of Varying Weight Numbers in FDConv. To further evaluate the robustness of our method, we analyze FDConv's weight similarity when the number of weights is set to 4, 16, and 64, as shown in Figure 3. Across all settings, each weight maintains a cosine similarity of 1.0 only with itself and 0 with others, demonstrating FDConv's consistent ability to learn highly diverse weights regardless of the configuration. This flexibility is critical for scaling the model to more complex tasks without compromising diversity.

The results presented in this section demonstrate the clear advantages of FDConv over existing methods. By learning highly diverse weights, FDConv achieves greater adaptability, which is essential for capturing complex input variations and delivering superior performance across a range of tasks.

Mathematical Analysis of Weight Similarity. Actually, Fourier Disjoint Weight (FDW) leverages the orthogonality of disjoint Fourier indices to ensure that the constructed

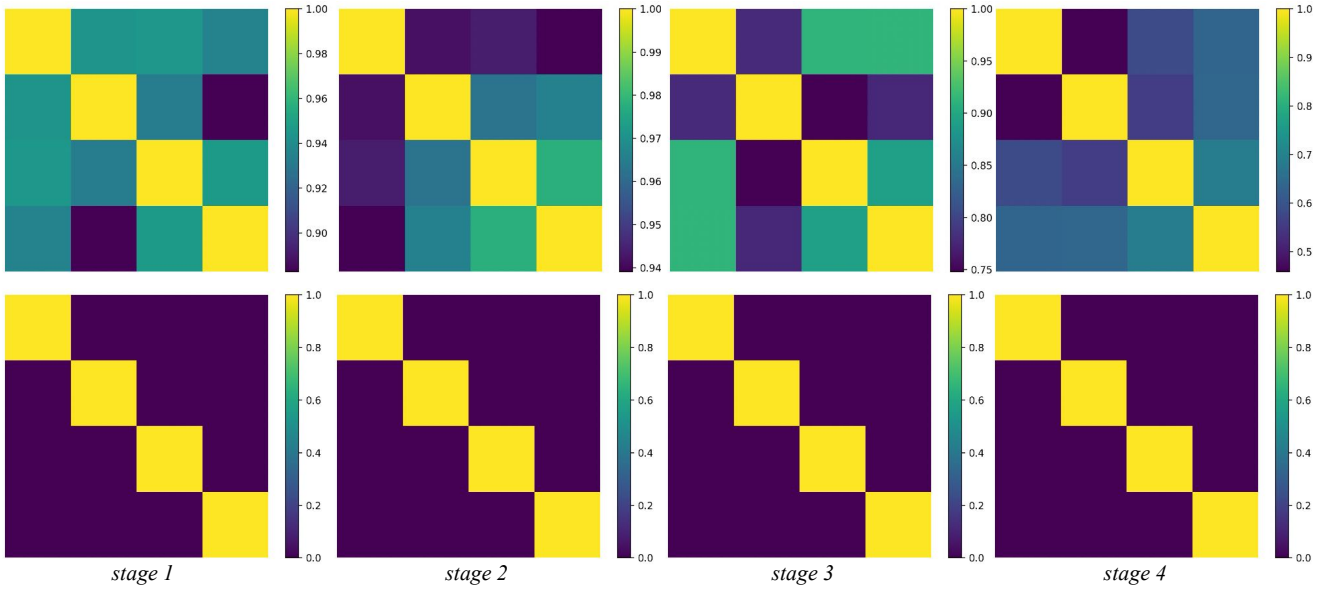


Figure 1. Weight similarity analysis. We present representative results from stages 1-4 of the models [5]. The top row in the figure illustrates the cosine similarity between the four learned weights from previous methods [6, 7, 16, 18]. They exhibit very high similarity with each other (*e.g.*, >0.94 at stage 2), indicating limited diversity. The bottom row displays the cosine similarity between the four learned weights from our FDConv. In contrast to existing methods, each weight shows a cosine similarity of 1.0 only with itself and 0 with the others, demonstrating high diversity. To ensure a fair comparison, we set the number of weights to 4, consistent with the configurations of the compared methods.

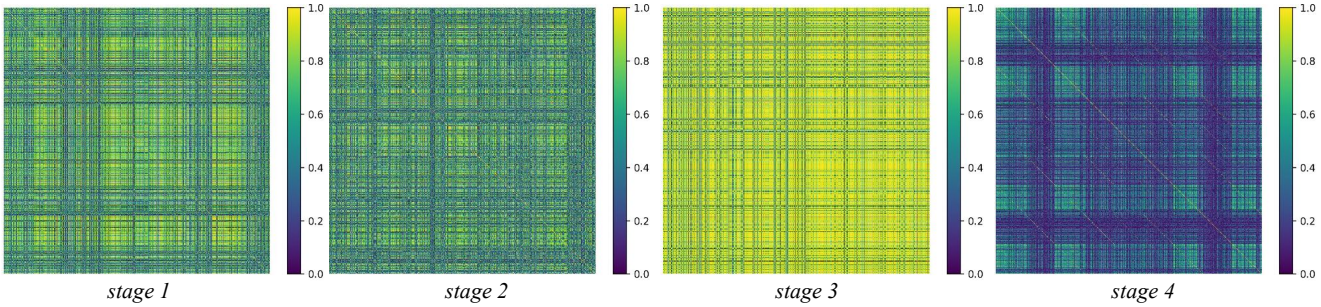


Figure 2. Weight similarity analysis for recent KW [6]. We present representative results from stages 1-4 of the models [5]. KW adopts a weight-sharing strategy across different layers, where each layer uses an attention mechanism to linearly mix the same shared weight warehouse as the final weight. Therefore, we analyze the similarity of the learned attention weights to evaluate final combined weight. We observe that, for each layer, the learned attention weights exhibit high similarity to each other. This leads to predicting similar attention values for weight mixing, ultimately resulting in high similarity in the final combined weights.

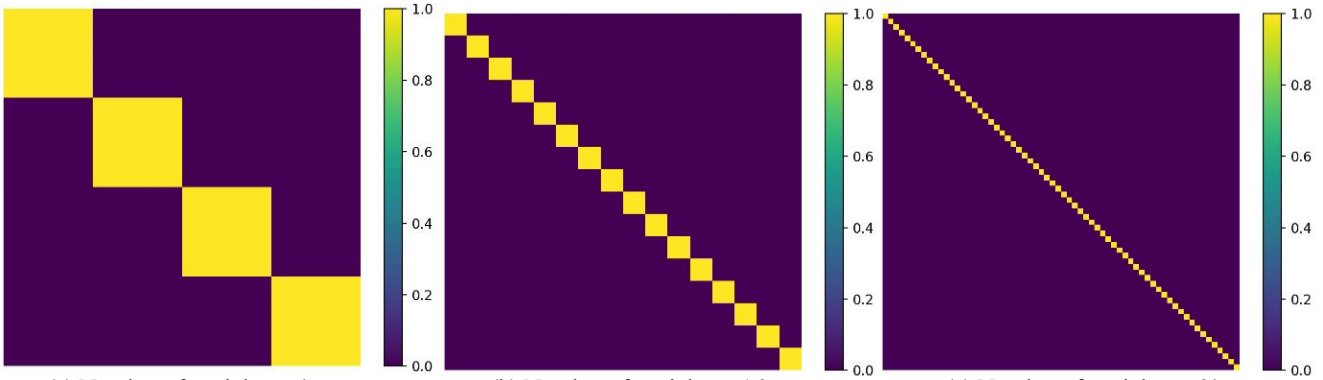
weights are highly diverse. The key principle is that Fourier indices assigned to different parameter groups are mutually disjoint, meaning they share no overlapping frequency components. This disjoint property inherently leads to orthogonal frequency responses for the corresponding weights.

Let $\mathbf{P}^i$ and $\mathbf{P}^j$ denote two disjoint groups of Fourier parameters, where $i \neq j$. After applying the inverse Discrete Fourier Transform (iDFT), the spatial representations of these groups are denoted as $\mathbf{S}^i$ and $\mathbf{S}^j$, respectively. Since $\mathbf{P}^i$ and $\mathbf{P}^j$ correspond to non-overlapping Fourier indices, the iDFT of each group results in orthogonal spatial components. This can be formally expressed as:

$$\langle \mathbf{S}^i, \mathbf{S}^j \rangle = 0, \quad \text{for } i \neq j, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. This orthogonality ensures that the cosine similarity between any two spatial weights $\mathbf{W}^i$ and $\mathbf{W}^j$, reshaped from $\mathbf{S}^i$ and $\mathbf{S}^j$, is zero:

$$\text{CosSim}(\mathbf{W}^i, \mathbf{W}^j) = \frac{\langle \mathbf{W}^i, \mathbf{W}^j \rangle}{\|\mathbf{W}^i\| \|\mathbf{W}^j\|} = 0, \quad \text{for } i \neq j. \tag{2}$$



(a) Number of weights = 4 (b) Number of weights = 16 (c) Number of weights = 64

Figure 3. Weight similarity analysis with varying numbers of weights for our FDConv. We present the cosine similarity between the learned weights from our FDConv when the number of weights is set to 4, 16, and 64. Each weight exhibits a cosine similarity of 1.0 only with itself and 0 with the others, demonstrating high diversity across all settings.

Figures 1 and 3 empirically validates this property by comparing the cosine similarity of weights generated by FDW and existing dynamic convolution methods [6, 7, 16, 18]. While previous methods exhibit high similarity between weights, FDW achieves zero similarity across all pairs of weights. This property is crucial for ensuring diverse frequency responses, as each weight captures unique information without redundancy.

In summary, the orthogonality of Fourier Disjoint Weight stems from its frequency-domain design, where each group of Fourier indices contributes independently to the overall representation. This mathematical property highlights the superiority of FDW in generating highly diverse weights, thereby enabling dynamic convolution layers to capture richer and more adaptive feature representations.

B. Weight Frequency Response Analysis

To validate the effectiveness of the proposed FDConv, we conduct a comprehensive analysis of the frequency responses of learned weights in comparison to prior dynamic convolution methods [6, 7, 16, 18]. Figure 4 presents the results of this analysis.

The top row of Figure 4 shows the frequency responses of weights learned by previous methods, such as ODConv [7]. We present representative results from stages 1-4 of the models [5]. These methods operate in the spatial domain and exhibit highly similar frequency responses across their parallel weights. For instance, the four parallel weights of ODConv demonstrate a lack of diversity, with responses clustered closely in the frequency domain. This indicates limited adaptability in capturing features across different frequency bands.

In contrast, the bottom row of Figure 4 highlights the frequency responses of weights learned by our FDConv, which operates directly in the frequency domain. To ensure a fair comparison, we configure FDConv to use four weights, matching the setup of ODConv. Notably, FDConv produces distinct frequency responses for each weight, effectively spanning various regions of the frequency spectrum. This diversity enables FDConv to better capture features at both high and low frequencies, leading to superior adaptability and enhanced feature extraction capabilities.

The results clearly demonstrate that FDConv addresses the limitations of prior methods by introducing diverse and complementary frequency responses. This ability to effectively decompose and process information across a wide frequency range is pivotal for tasks requiring detailed feature representations, such as segmentation, detection, and classification.

C. Weight t-SNE Analysis

To further evaluate the diversity of learned weights, we conduct a t-SNE [15] analysis on the weights obtained from stages 1-4 of the models [5]. The results are presented in Figure 5, where the top and bottom rows correspond to previous methods [6, 7, 16, 18] and our proposed FDConv, respectively. To ensure a fair comparison, we set the number of weights to 4 for FDConv.

In the top row, the t-SNE visualizations show that the filters from the four weights of previous method [7] exhibit a high degree of overlap, indicating limited diversity. Conversely, the bottom row demonstrates that the weights learned by our FDConv exhibit significantly more diverse distributions across the t-SNE projections. This enhanced diversity is evident in all four stages and suggests that FDConv learns more discriminative representations.

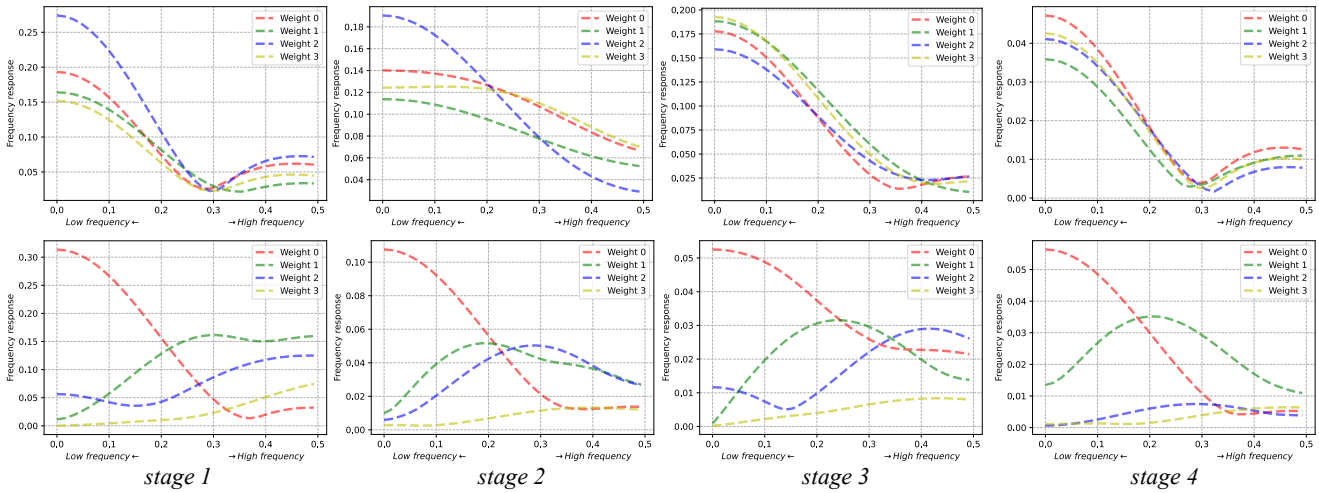


Figure 4. Weight frequency response analysis. We present representative results from stages 1-4 of the models [5]. The top row illustrates the frequency responses of learned weights from previous methods [6, 7, 16, 18]. These methods learn weights in the spatial domain, and the frequency responses of the four parallel weights in ODConv are highly similar, indicating limited diversity. The bottom row displays the frequency responses of learned weights from our FDConv, which learns parallel weights in the frequency domain. To ensure a fair comparison, we set the number of weights to 4, consistent with them. In contrast, FDConv exhibits distinct frequency responses for each weight, covering different regions of the frequency spectrum.

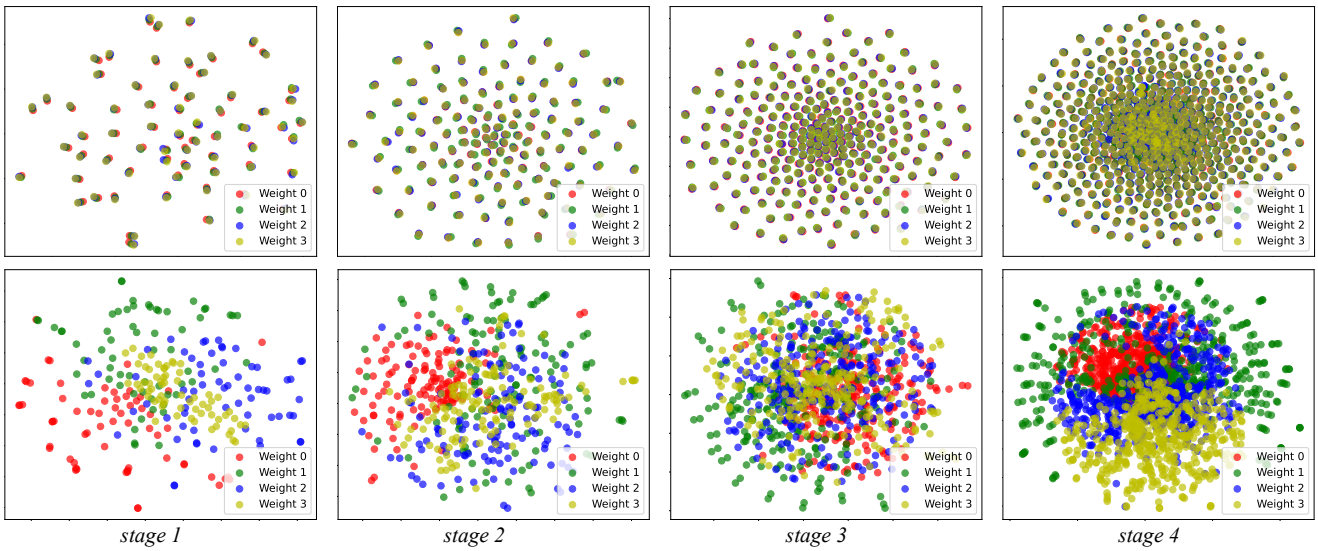


Figure 5. Weight t-SNE [15] analysis. We present representative results from stages 1-4 of the models [5]. The top row illustrates the t-SNE results of learned weights from previous methods [6, 7, 16, 18]. We can see that the filters in the four weights distribute closely to each other. The bottom row displays the t-SNE results of learned weights from our FDConv. The filters in the four weights of FDConv exhibit different distributions, indicating greater diversity. To ensure a fair comparison, we set the number of weights to 4, consistent with them.

D. Feature Visualization

To demonstrate the behavior of Frequency Band Modulation (FBM), we visualize the modulation maps for different frequency bands, as shown in Figure 6. For better performance, the modulation map for the lowest frequency band is empirically set to 1 across all spatial locations. This ensures consistent emphasis on low-frequency components.

We observe that higher frequency bands exhibit high modulation value around object boundaries, as seen in Figure 6(b)-(d). Conversely, lower frequency bands display

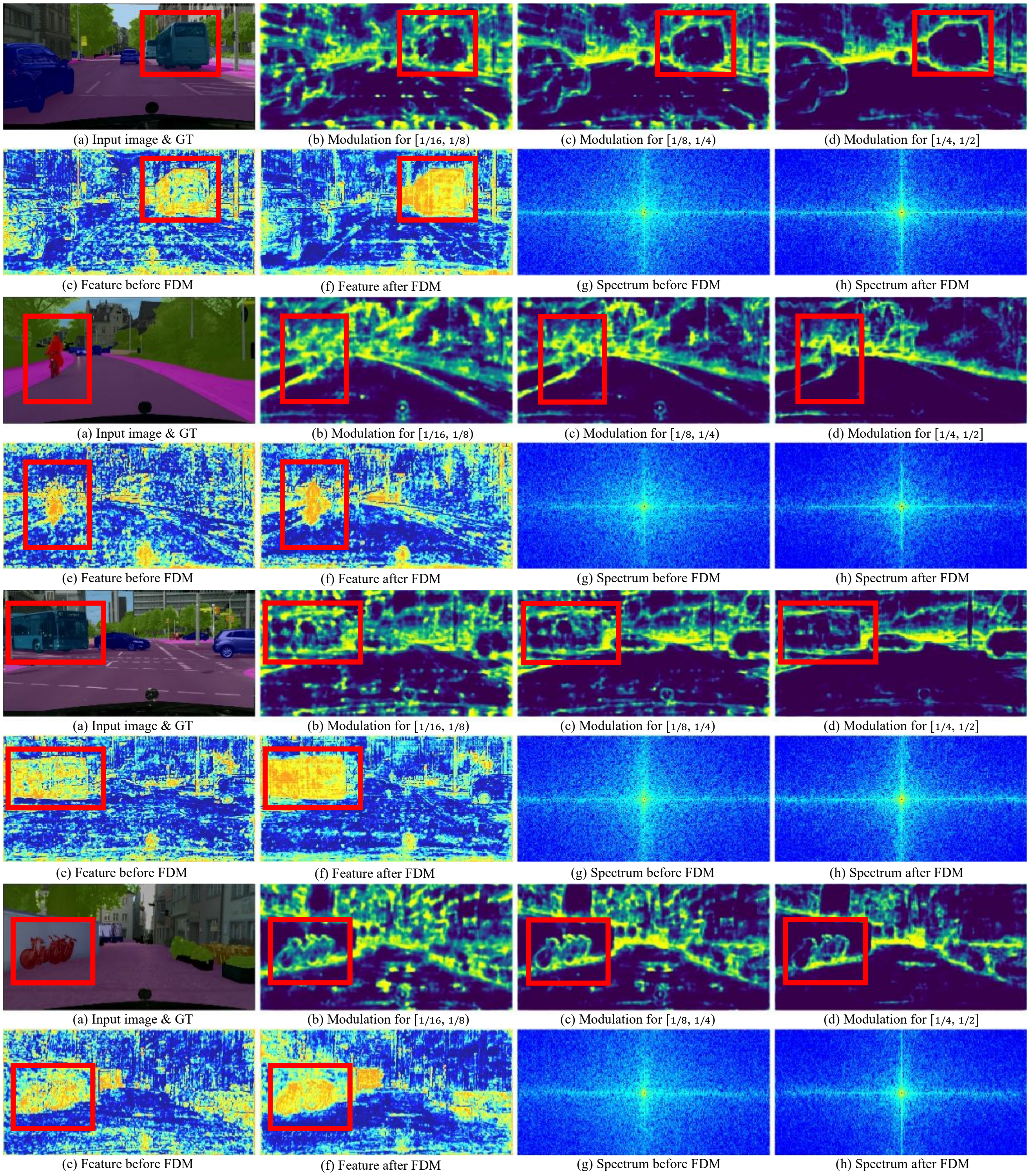


Figure 6. Visualization of Frequency Band Modulation. (a) shows the input images and their corresponding ground truth (GT). (b)–(d) display the modulation maps for different frequency bands, ranging from low to high. (e) and (f) visualize the feature frequency spectrum. Normalized frequency $[0, 0.5]$ is used for simplicity, $\times 2\pi$ yields normalized angular frequency.

Table 1. Ablation studies for FDConv on the COCO validation set [9], showcasing the integration of Fourier Disjoint Weight (FDW), Kernel Spatial Modulation (KSM), and Frequency Band Modulation (FBM).

Models	Params	AP^{box}
Mask R-CNN	$46.5_{(23.5)}$M	39.6
+ FDW	+ 1.3M	40.6 (+1.0)
+ FDW + KSM (local only)	+1.6M	41.2 (+1.6)
+ FDW + KSM	+ 3.5M	41.8 (+2.2)
+ FDW + KSM + FBM	+ 3.6M	**42.4 (+2.8)**

Table 2. Ablation study on the number of weights (n) in FDConv. The results are reported on the COCO validation set [9].

Number of Weights (n)	$n=4$	$n=16$	$n=64$	$n=256$
AP^{box}	42.0	42.2	**42.4**	**42.4**

Table 3. Ablation study of Frequency Band Modulation (FBM) on the COCO dataset [9]. The phrase "Set lowest band to 1.0" refers to assigning a fixed modulation weight of 1.0 to the lowest frequency band. Normalized frequency $[0, 0.5]$ is used for simplicity, $\times 2\pi$ yields normalized angular frequency.

Frequency Band	Set lowest band to 1.0	AP^{box}
2: $[0, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$	✓	42.2
3: $[0, \frac{1}{8}), [\frac{1}{8}, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$	✓	42.3
4: $[0, \frac{1}{16}), [\frac{1}{16}, \frac{1}{8}), [\frac{1}{8}, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$	✓	**42.4**
5: $[0, \frac{1}{32}), [\frac{1}{32}, \frac{1}{16}), [\frac{1}{16}, \frac{1}{8}), [\frac{1}{8}, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$	✓	42.2
4: $[0, \frac{1}{16}), [\frac{1}{16}, \frac{1}{8}), [\frac{1}{8}, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$		42.1

Table 4. Inference speed evaluation. The frame-per-second (FPS) results were tested using a feature map size of $128 \times 64 \times 64$ on an i9-10850K CPU @ 3.60GHz. n indicates the number of parallel weights.

Model	Standard Conv	ODConv [7]	FDConv (Ours)
FPS ($n=1$)	736.1	-	-
FPS ($n=4$)		**498.7**	454.8
FPS ($n=16$)	-	325.3	**450.2**
FPS ($n=64$)	-	135.7	**447.1**

higher modulation values within object interiors, highlighting the selective focus of FBM. This selective modulation pattern enables FDConv to suppress high-frequency noise in areas such as the background and object centers, which contribute less to accurate predictions.

As depicted in Figure 6(e)-(f), the application of FBM leads to a significant reduction in high-frequency noise in the feature maps. Furthermore, the spectral analysis in Figure 6(g)-(h) confirms the suppression of unnecessary high-frequency components, while simultaneously enhancing critical foreground features. This results in more precise and complete representations of objects, which is particularly advantageous for dense prediction tasks such as segmentation and detection.

E. Ablation Study

To evaluate the effectiveness and efficiency of FDConv, we conducted ablation studies on the COCO validation set [9]. These experiments examine the contributions of Fourier Disjoint Weight (FDW), Kernel Spatial Modulation (KSM), and Frequency Band Modulation (FBM), as well as the parameter trade-offs and inference speed.

Effectiveness of FDW, KSM, and FBM. Table 1 demonstrates the progressive integration of FDConv components into the Mask R-CNN baseline. Adding FDW improves AP^{box} by +1.0, showcasing its ability to enhance frequency diversity with only a minor parameter increase (+1.3M), compared to prior works [6, 7, 16, 18], which often lead to significant parameter growth (*e.g.*, $4\times$). Incorporating KSM (local channel branch only) provides an additional gain of +0.6, while full KSM (both global and local channel branches) achieves a more substantial improvement of +1.2. Finally, FBM increases AP^{box} to 42.4 (+2.8 overall), emphasizing its pivotal role in optimizing frequency band-specific modulation across different spatial locations.

Weight Numbers in FDConv. Table 2 examines the impact of the number of parallel weights (n) in FDConv. Increasing n from 4 to 64 improves AP^{box} from 42.0 to 42.4, but further increases show diminishing returns. Notably, FDConv constructs weights by partitioning a fixed set of parameters in the Fourier domain into n disjoint groups, generating n parallel weights without increasing the parameter cost. This design ensures minimal additional overhead, maintaining efficiency across diverse tasks.

Frequency Band Modulation Design. Table 3 examines the impact of dividing the frequency spectrum into varying numbers of bands. By default, we decompose the frequency spectrum into four distinct bands using an octave-based partitioning strategy [14]. Normalized frequency $[0, 0.5]$ is used for simplicity, $\times 2\pi$ yields normalized angular frequency.

As shown in Table 3, increasing the granularity of frequency band divisions improves performance up to a certain point. Dividing the frequency spectrum into 4 bands ($[0, \frac{1}{16}), [\frac{1}{16}, \frac{1}{8}), [\frac{1}{8}, \frac{1}{4}), [\frac{1}{4}, \frac{1}{2}]$) achieves the best performance, with an AP^{box} of 42.4. However, further increasing the granularity to 5 bands slightly reduces performance, likely due to over-division of the frequency spectrum, which may dilute the modulation effect.

We also evaluate the impact of setting a fixed modulation weight of 1.0 for the lowest frequency band. When this condition is removed, the performance drops from 42.4 to 42.1. This demonstrates the importance of maintaining strong low-frequency responses for capturing global struc-

ture and stability during feature learning.

Inference Speed Analysis Table 4 compares inference speeds between standard convolution, ODConv [7], and FDConv. The FBM is bypassed. Tested on an i9-10850K CPU, FDConv achieves competitive frame-per-second (FPS) rates, maintaining over 447 FPS even with $n = 64$. This demonstrates FDConv's efficiency compared to ODConv, which suffers from a significant speed drop as n increases.

F. Experimental Settings

Datasets and Metrics. We evaluate our methods on three challenging benchmarks: Cityscapes [2], ADE20K [19], and COCO [10].

Cityscapes [2] is a widely used semantic segmentation dataset featuring 19 classes. It contains 5,000 finely annotated images at a resolution of 2048×1024 pixels, divided into training (2,975), validation (500), and test (1,525) sets. We use only the training set for learning. ADE20K [19] is a more diverse dataset, covering 150 semantic categories, with 20,210 training images, 2,000 validation images, and 3,352 test images.

To assess object detection and instance segmentation, we leverage the COCO dataset [10], a standard benchmark in these domains. For evaluation metrics, we adopt mean Intersection over Union (mIoU) for semantic segmentation tasks and Average Precision (AP) for object detection and instance segmentation.

Implementation Details. For Mask2Former [1], we adhere to the original training protocols [1]. Data augmentation includes random cropping, horizontal flipping, and scaling in the range [0.5, 2.0]. The training uses a poly learning rate decay strategy. Key hyperparameters include 90k iterations, an initial learning rate of $1e{-4}$, a weight decay of $5e{-2}$, cropped input size of 512×1024, and a batch size of 16, optimized with AdamW.

For UPerNet with ResNet backbones [5], all models are trained for 160k iterations using AdamW [11], with a batch size of 16. Mask2Former [1] and MaskDINO [8] follow their respective training setups outlined in their original papers.

On COCO, we follow common practices [4, 12, 17] for training object detection and instance segmentation models. All models are trained for 12 epochs using the $1\times$ schedule, ensuring compatibility with standard benchmarks.

For ImageNet, we follow standard training settings to ensure fair comparisons across methods. Specifically, ResNet-18 models are trained with SGD for 100 epochs, using a batch size of 256, momentum of 0.9, and weight decay of 0.0001. The initial learning rate is set to 0.1 and decayed by a factor of 10 every 30 epochs. These settings align with prior work [7, 16].

G. Parameters of Frequency Disjoint Weight

In Section 3.1 of the main paper, we set the parameter budget to $k \times k \times C_{\text{in}} \times C_{\text{out}}$. FDW first treats these parameters as learnable spectral coefficients in the Fourier domain, reshaping them into $\mathbf{P} \in \mathbb{R}^{kC_{\text{in}} \times kC_{\text{out}}}$. Since the Fourier domain coefficients are complex-valued, each coefficient requires two parameters: one for the real part and one for the imaginary part. Thus, a naive approach would require $2 \times k \times k \times C_{\text{in}} \times C_{\text{out}}$ parameters.

In FDConv, we exploit the inherent symmetry of the Fourier Transform for real-valued inputs to reduce the parameter cost. Since the input images, feature maps, and weights are real-valued, their Fourier domain representations exhibit Hermitian symmetry [3, 13]. Specifically, the Fourier Transform of a real-valued function $f(x, y)$ satisfies the following property:

$$F(u, v) = \overline{F(-u, -v)}, \tag{3}$$

where $F(u, v)$ is the Fourier coefficient at frequency (u, v), and $\overline{F(-u, -v)}$ is the complex conjugate of $F(-u, -v)$.

Proof of Hermitian Symmetry. The Discrete Fourier Transform (DFT) of a real-valued function $f(x, y)$ is given by:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)}. \tag{4}$$

If $f(x, y)$ is real, then:

$$\overline{F(u, v)} = \overline{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)}} \tag{5}$$

$$= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{j2\pi\left(\frac{ux}{N} + \frac{vy}{M}\right)} \tag{6}$$

$$= \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi\left(\frac{ux}{N} + \frac{-vy}{M}\right)} \tag{7}$$

$$= F(-u, -v). \tag{8}$$

Thus, $F(u, v) = \overline{F(-u, -v)}$, proving that the Fourier coefficients of real-valued inputs exhibit Hermitian symmetry.

Implications for Parameter Efficiency. Due to Hermitian symmetry, only half of the Fourier coefficients are unique, as the coefficients at negative frequencies are determined by their positive counterparts. This property eliminates the need to learn or store redundant frequency components.

As a result, the parameters required to construct convolution weights in the Fourier domain are effectively halved compared to a naive approach. Instead of requiring a parameter budget of $2 \times k \times k \times C_{\text{in}} \times C_{\text{out}}$, FDConv requires only $k \times k \times C_{\text{in}} \times C_{\text{out}}$, ensuring parameter efficiency while retaining full expressive power.

H. Equivalent Implementation of FBM

In the main paper, we introduce Frequency Dynamic Modulation (FBM), which operates in the following key steps: 1) Kernel frequency decomposition, decomposing the frequency response of the convolution weight into distinct frequency bands. 2) Convolution in the Fourier domain, performing convolution operations in the Fourier domain. 3) Spatially variant modulation, predicting spatially adaptive modulation values for each frequency band of the convolution weight. Here, we introduce an equivalent implementation of FBM. The core formulation of FBM is expressed as:

$$\mathbf{Y} = \sum_{b=0}^{B-1} (\mathbf{A}_b \odot (\mathbf{W}_b * \mathbf{X})), \tag{9}$$

where $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{h \times w}$ represent the input and output feature maps (the channel dimension is omitted for simplicity). $\mathbf{A}_b \in \mathbb{R}^{h \times w}$ denotes spatially varying modulation values specific to the b-th frequency band, and $\mathbf{W}_b$ represents the convolution weight associated with the b-th frequency band. FBM enables adaptive adjustment of frequency responses for each spatial location in the feature map.

The frequency band-specific weight $\mathbf{W}_b$ is computed as:

$$\mathbf{W}_b = \mathcal{F}^{-1}(\mathcal{M}_b \odot \mathcal{F}(\mathbf{W})), \tag{10}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Discrete Fourier Transform (DFT) and its inverse, respectively, and $\mathcal{M}_b$ is a binary mask isolating specific frequency ranges. According to the Convolution Theorem [3], convolution in the spatial domain is equivalent to pointwise multiplication in the frequency domain. This property allows us to rewrite $\mathbf{W} * \mathbf{X}$ as:

$$\mathbf{W} * \mathbf{X} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{W}) \odot \mathcal{F}(\mathbf{X})). \tag{11}$$

Substituting this equivalence into the FBM formulation yields:

$$\begin{aligned}
\mathbf{Y} &= \sum_{b=0}^{B-1} \left(\mathbf{A}_b \odot \mathcal{F}^{-1}\left((\mathcal{M}_b \odot \mathcal{F}(\mathbf{W})) \odot \mathcal{F}(\mathbf{X})\right)\right) \\
&= \sum_{b=0}^{B-1} \left(\mathbf{A}_b \odot \mathcal{F}^{-1}\left((\mathcal{M}_b \odot \mathcal{F}(\mathbf{X})) \odot \mathcal{F}(\mathbf{W})\right)\right) \\
&= \sum_{b=0}^{B-1} \left(\mathbf{A}_b \odot \mathcal{F}^{-1}\left(\mathcal{M}_b \odot \mathcal{F}(\mathbf{X})\right) * \mathbf{W}\right) \\
&= \sum_{b=0}^{B-1} \left(\mathbf{A}_b \odot \mathbf{X}_b * \mathbf{W}\right) \\
&= \sum_{b=0}^{B-1} \left(\mathbf{A}_b \odot \mathbf{X}_b\right) * \mathbf{W},
\end{aligned} \tag{12}$$

where $\mathbf{X}_b = \mathcal{F}^{-1}(\mathcal{M}_b \odot \mathcal{F}(\mathbf{X}))$ represents the input feature map filtered to the b-th frequency band.

Thus, FBM can be implemented by decomposing the input feature map into different frequency bands, applying spatially variant modulation, and performing convolution with the weight.

References

[1] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 7

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 7

[3] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009. 7, 8

[4] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. 2022. 7

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 2, 3, 4, 7

[6] Chao Li and Anbang Yao. Kernelwarehouse: Rethinking the design of dynamic convolution. In *Proceedings of International Conference on Machine Learning*, 2024. 1, 2, 3, 4, 6

[7] Chao Li, Aojun Zhou, and Anbang Yao. Omni-dimensional dynamic convolution. In *Proceedings of International Conference on Learning Representations*, 2022. 1, 2, 3, 4, 6, 7

[8] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023. 7

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of European Conference on Computer Vision*, pages 740–755, 2014. 6

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[12] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Proceedings of Advances in Neural Information Processing Systems*, 35:10353–10366, 2022. 7

[13] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In

Proceedings of Advances in Neural Information Processing Systems, volume 34, pages 980–993, 2021. 7

[14] Ajay Subramanian, Elena Sizikova, Najib J Majaj, and Denis G Pelli. Spatial-frequency channels, shape bias, and adversarial robustness. pages 1–10, 2024. 6

[15] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 3, 4

[16] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2320–2329, 2020. 1, 2, 3, 4, 6, 7

[17] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023. 7

[18] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Proceedings of Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 3, 4, 6

[19] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 7