Frequency-Dynamic Attention Modulation for Dense Prediction
(supplementary material)

Linwei Chen[1] Lin Gu[2,3] Ying Fu[1*]
[1]Beijing Institute of Technology [2]RIKEN AIP [3]The University of Tokyo
chenlinwei@bit.edu.cn; lin.gu@riken.jp; fuying@bit.edu.cn

This supplementary material provides more details and results that are not included in the main paper due to space limitations. The contents are organized as follows:

- Section A formulates how to ***analyze the frequency response*** of the attention mechanism.
- Section B provides a ***mathematical proof*** for attention as low-pass filters.
- Section C describes the details of an ***simple implementation*** for the proposed Attention Inversion.
- Section D provides more details about the ***experimental settings***.
- Section E introduces details of the ***ablation study*** results to evaluate each module, parameter, and time complexity.
- Section F shows more details for ***effective rank analysis***.
- Section G shows more details and results for ***feature similarity analysis***.
- Section H provides more ***visualized results***.

A. Frequency Response Analysis for Attention

Understanding the inner workings of attention mechanisms is crucial for interpreting their role in enhancing feature representations. In this section, we propose a frequency response analysis framework that quantifies how attention modulates the frequency content of input features. We first introduce how to calculate the global frequency response of attention, and then we describe how to calculate the local frequency response. The visualized frequency response analysis can be found in Section H.

A.1. Global Frequency Response.

Following [16], we regard attention as a system and define the attention frequency response as *the ratio between the frequency spectrum of the output and that of the input*.

Let $\mathbf{V} \in \mathbb{R}^{H\times W\times C}$ denote an input value map and $\mathbf{Y} \in \mathbb{R}^{H\times W\times C}$ be the corresponding output after applying an attention module. We analyze the frequency content of these feature maps using the Discrete Fourier Transform (DFT). For each channel c, consider the two-dimensional value map $\mathbf{V}_c(p,q)$ where $0 \le p < H$ and $0 \le q < W$. The DFT of $\mathbf{V}_c$, $\mathbf{Y}_c$ are defined as:

$$\begin{aligned}\mathcal{F}(\mathbf{V}_c)(u,v) &= \sum_{p=0}^{H-1}\sum_{q=0}^{W-1} \mathbf{V}_c(p,q)\, e^{-j2\pi\left(\frac{up}{H}+\frac{vq}{W}\right)},\\ \mathcal{F}(\mathbf{Y}_c)(u,v) &= \sum_{p=0}^{H-1}\sum_{q=0}^{W-1} \mathbf{Y}_c(p,q)\, e^{-j2\pi\left(\frac{up}{H}+\frac{vq}{W}\right)},\end{aligned} \tag{1}$$

where (u,v) are the frequency coordinates with $0 \le u < H$ and $0 \le v < W$, and $j = \sqrt{-1}$ is the imaginary unit.

After obtaining the Fourier spectra, we define the attention frequency response $\mathbf{R}(u,v)$ as:

$$\mathbf{R}(u,v) = \frac{|\mathcal{F}(\mathbf{Y})(u,v)|}{|\mathcal{F}(\mathbf{V})(u,v)|}. \tag{2}$$

Here, $\mathcal{F}(\mathbf{V})(u,v)$ and $\mathcal{F}(\mathbf{Y})(u,v)$ denote the Fourier transform of the input and output features respectively, computed either per-channel or as an aggregated spectrum. A value of $\mathbf{R}(u,v) > 1$ indicates an amplification of the frequency component (u,v) by the attention mechanism, whereas $\mathbf{R}(u,v) < 1$ suggests a suppression.

A.2. Local Frequency Response.

It is non-trival to obtain the local frequency response for each pixel in the feature map output by attention mechanism. To achieve this, we firstly view the attention mechanism as dynamic convolution, as shown in Figure 1, and the obtain the local frequency response by convolution theory.

Standard Attention Mechanism. Given an input tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, self-attention derives queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ via linear projections:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable weights. The attention output is computed as:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{C}}\right)\mathbf{V}, \tag{4}$$

*Corresponding Author

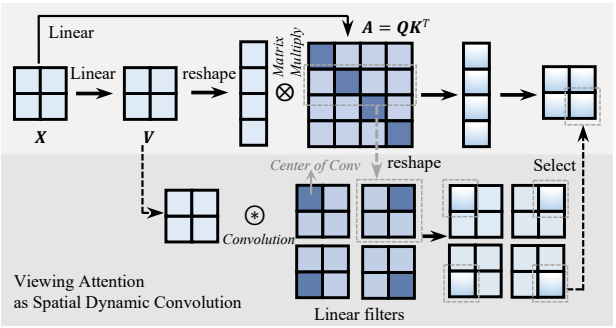


Figure 1. An illustration of interpreting attention as spatial dynamic convolution. The top row shows the conventional attention operation, where the attention matrix $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$ is computed from query $\mathbf{Q}$ and key $\mathbf{K}$. The bottom row demonstrates how reshaping $\mathbf{A}$ into position-specific kernels allows it to be viewed as a set of dynamic filters convolving with the value map $\mathbf{V}$. This perspective unifies attention and dynamic convolution under a single framework.

Dynamic Convolution. Traditional convolution applies fixed kernels $\mathbf{W}$ to local neighborhoods. In contrast, *dynamic convolution* [7, 9, 23] generates input-dependent kernels $\mathbf{W}_{p,q}$ at each spatial position (p, q):

$$\mathbf{Y}(p, q) = (\mathbf{W}_{p,q} * \mathbf{X})(p, q). \tag{5}$$

In this formulation, $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ denotes the input feature map with height H, width W, and C channels, while $\mathbf{Y}(p, q)$ represents the output feature at position (p, q). The symbol $*$ indicates the convolution operation, and $\mathbf{W}_{p,q}$ is the dynamic kernel specifically generated for the spatial location (p, q). This dynamic kernel is applied over a local neighborhood of $\mathbf{X}$ to produce the corresponding output, allowing the convolution process to adapt based on the input content.

Attention Weights as Dynamic Kernels. The attention matrix $\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{c}}\right)$ encodes pairwise affinity scores. By reshaping $\mathbf{A} \in \mathbb{R}^{HW\times HW}$ into spatially localized grids (i.e., $H \times W$, where each grid contains a kernel of size $H \times W$), we reinterpret $\mathbf{A}$ as a set of dynamic convolution kernels:

$$\mathbf{W}^A_{p,q} = \text{Reshape}\left(\mathbf{A}_{p,q}\right) \in \mathbb{R}^{H\times W}. \tag{6}$$

Here, $\mathbf{A}_{p,q}$ corresponds to the attention weights for the position (p, q), and the reshaping enforces a convolutional structure.

The attention output is computed by aggregating values $\mathbf{V}$ with the dynamic kernels $\mathbf{W}_{p,q}$:

$$\mathbf{Y}(p, q) = (\mathbf{W}^A_{p,q} * \mathbf{V})(p, q). \tag{7}$$

This mirrors dynamic convolution.

Convolution Theorem. We can obtain the frequency response at location (p, q) as follows:

$$\mathbf{R}_{p,q}(u, v) = \frac{\left|\mathcal{F}(\mathbf{W}^A_{p,q} * \mathbf{V})(u, v)\right|}{|\mathcal{F}(\mathbf{V})(u, v)|}, \tag{8}$$

where $\mathbf{W}^A_{p,q}$ denotes the attention-based dynamic kernel at location (p, q), $\mathbf{V}$ is the value feature map, $\mathcal{F}(\cdot)$ represents the Fourier transform, (u, v) are the frequency coordinates, and ϵ is a small constant for numerical stability.

According to the convolution theorem [3], the spatial convolution of $\mathbf{W}^A_{p,q}$ and $\mathbf{V}$ corresponds to an element-wise multiplication of their frequency representations:

$$\mathcal{F}(\mathbf{W}^A_{p,q} * \mathbf{V})(u, v) = \mathcal{F}(\mathbf{W}^A_{p,q})(u, v) \cdot \mathcal{F}(\mathbf{V})(u, v). \tag{9}$$

Thus, we can express the local frequency response at (p, q) as:

$$\begin{aligned}\mathbf{R}_{p,q}(u, v) &= \frac{\left|\mathcal{F}(\mathbf{W}^A_{p,q})(u, v) \cdot \mathcal{F}(\mathbf{V})(u, v)\right|}{|\mathcal{F}(\mathbf{V})(u, v)|} \\ &= \left|\mathcal{F}(\mathbf{W}^A_{p,q})(u, v)\right|.\end{aligned} \tag{10}$$

Therefore, the local frequency response corresponds to the magnitude of the frequency spectrum of the local linear filter $\mathbf{W}^A_{p,q}$ obtained from attention matrix $\mathbf{A}$.

B. Attention as Low-Pass Filters

Given $\mathbf{W}^A \in \mathbb{R}^{H\times W}$ represents a reshaped filter derived from the softmax normalized attention matrix, it is evident that the filter satisfies the properties:

$$0 \le \mathbf{W}^A(m, l) \le 1, \quad \text{and} \quad \sum_{m=0}^{H-1}\sum_{l=0}^{W-1} \mathbf{W}^A(m, l) = 1. \tag{11}$$

We can determine its frequency response by applying the Discrete Fourier Transform (DFT). The two-dimensional discrete Fourier transform (DFT) of $\mathbf{W}^A$ is defined as:

$$\mathcal{F}(\mathbf{W}^A)(u, v) = \sum_{m=0}^{H-1}\sum_{l=0}^{W-1} \mathbf{W}^A(m, l) e^{-2\pi j\left(\frac{um}{H} + \frac{vl}{W}\right)}, \tag{12}$$

where $u = 0, 1, \ldots, H-1$ and $v = 0, 1, \ldots, W-1$. This transform provides the frequency response of the filter $\mathbf{W}^A$ across different spatial frequencies u and v.

Lowest Frequency. For the lowest zero-frequency (DC) component, we have:

$$\begin{aligned}\mathcal{F}(\mathbf{W}^A)(0, 0) &= \sum_{m=0}^{H-1}\sum_{l=0}^{W-1} \mathbf{W}^A(m, l) \cdot e^{-2\pi j\left(\frac{0}{H} + \frac{0}{W}\right)} \\ &= 1.\end{aligned} \tag{13}$$

This follows directly from the softmax normalization property. It means that the filters in the softmax-normalized attention matrix preserve the magnitude of the lowest frequency component.

High Frequency. Parsevals theorem [1, 21] states that the total energy in the spatial domain equals the total energy in the frequency domain:

$$\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\left|\mathbf{W}^A(m,l)\right|^2=\frac{1}{HW}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1}\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|^2. \tag{14}$$

Since $\mathbf{W}^A$ is a probability distribution, by Jensens inequality we have:

$$\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\left(\mathbf{W}^A(m,l)\right)^2<\left(\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\mathbf{W}^A(m,l)\right)^2=1^2. \tag{15}$$

Thus, it follows that:

$$\sum_{u=0}^{H-1}\sum_{v=0}^{W-1}\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|^2<HW. \tag{16}$$

Since the DC component satisfies:

$$\left|\mathcal{F}(\mathbf{W}^A)(0,0)\right|^2=1, \tag{17}$$

we obtain:

$$\sum_{(u,v)\neq(0,0)}\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|^2<HW-1. \tag{18}$$

This implies that for at least some $(u,v)\neq(0,0)$ we must have:

$$\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|<1. \tag{19}$$

For any (u,v), we consider the magnitude of the Fourier transform, by the triangle inequality, we have

$$\begin{aligned}\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|&=\left|\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\mathbf{W}^A(m,l)\,e^{-2\pi j\left(\frac{um}{H}+\frac{vl}{W}\right)}\right|\\&\leq\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\left|\mathbf{W}^A(m,l)\,e^{-2\pi j\left(\frac{um}{H}+\frac{vl}{W}\right)}\right|\\&=\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\mathbf{W}^A(m,l)\left|e^{-2\pi j\left(\frac{um}{H}+\frac{vl}{W}\right)}\right|\\&=\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\mathbf{W}^A(m,l)\cdot 1\\&=\sum_{m=0}^{H-1}\sum_{l=0}^{W-1}\mathbf{W}^A(m,l)\\&=1\end{aligned} \tag{20}$$

Equality in the triangle inequality holds if and only if all the complex numbers being summed share the same phase. In our case, this means that

$$e^{-2\pi i\left(\frac{um}{H}+\frac{vl}{W}\right)}=e^{i\theta}\quad\text{for all }(m,l)\text{ with }\mathbf{W}^A(m,l)>0, \tag{21}$$

for some constant $\theta\in\mathbb{R}$. This condition can be rewritten as

$$-2\pi\left(\frac{um}{H}+\frac{vl}{W}\right)\equiv\theta\pmod{2\pi}, \tag{22}$$

for every (m,l) in the sumport of $\mathbf{W}^A$.

For $(u,v)\neq(0,0)$, the phases $-2\pi\left(\frac{um}{H}+\frac{vl}{W}\right)$ vary with p and q unless the sumport of $\mathbf{W}^A$ is a singleton (*i.e.*, all the probability mass is concentrated at a single point). In typical settings with softmax-normalized attention, $\mathbf{W}^A$ is non-degenerate and has strictly positive values over indices. Hence, the exponential terms are not all aligned, and the convex combination lies strictly inside the unit circle. Therefore, we have

$$\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|<1\quad\text{for all }(u,v)\neq(0,0). \tag{23}$$

It means that the filters in the softmax-normalized attention matrix attenuate the magnitude of highier frequency $|(u,v)|>(0,0)$ component.

Conclusion. From the above derivation, we conclude:

$$\begin{aligned}\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|&=1,\text{if }(u,v)=(0,0),\\\left|\mathcal{F}(\mathbf{W}^A)(u,v)\right|&<1,\text{if }(u,v)\neq(0,0).\end{aligned} \tag{24}$$

Therefore, the filters in the softmax-normalized attention matrix preserve the lowest frequency component, attenuate high-frequency components, and thus behave as a low-pass filter.

Now, consider a simple model with L layers of pure self-attention. Let $\mathcal{F}(\mathbf{X}^{(i)})$ denote the Fourier transformed spectrum of the feature map at layer i, and let $\mathcal{F}(\mathbf{W}^{A,(i)})$ denote the frequency response of the attention matrix at the same layer. The transformation across layers follows the recursive relation:

$$\mathcal{F}(\mathbf{X}^{(L)})(u,v)=\prod_{i=1}^{L}\mathcal{F}(\mathbf{W}^{A,(i)})(u,v)\cdot\mathcal{F}(\mathbf{X}^{(0)})(u,v). \tag{25}$$

Since $|\mathcal{F}(\mathbf{W}^{A,(i)})(u,v)|<1$ for all nonzero frequencies $(u,v)\neq(0,0)$, we observe that:

$$\lim_{L\to\infty}\prod_{i=1}^{L}\left|\mathcal{F}(\mathbf{W}^{A,(i)})(u,v)\right|=0,\quad\forall(u,v)\neq(0,0). \tag{26}$$

This means that, all high-frequency components are exponentially suppressed with layers for each location, leaving only the lowest frequency component $(0,0)$ dominant. Consequently, the model suffers from frequency vanishing, where fine-grained details and textures are lost, impairing the model to capture crucial information for dense prediction vision tasks.

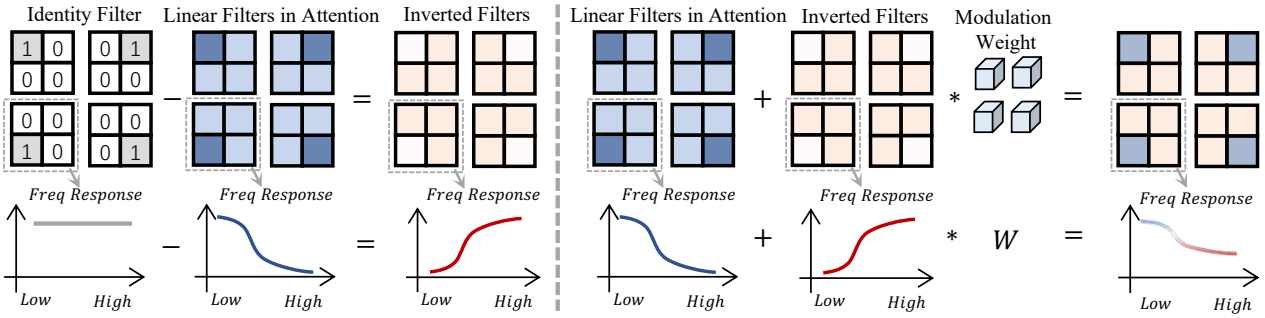


Figure 2. Illustration of inverting linear filters in the attention matrix. The identity filter has a all-pass frequency response. The linear filters in attention suppress high-frequency components, resulting in a low-pass effect. By subtracting these filters from the identity filter, we obtain their inverted counterparts, which emphasize high-frequency components. Combining the original and inverted filters with modulation weights leads to a flexible frequency response, where different spatial structures can be selectively enhanced or suppressed.

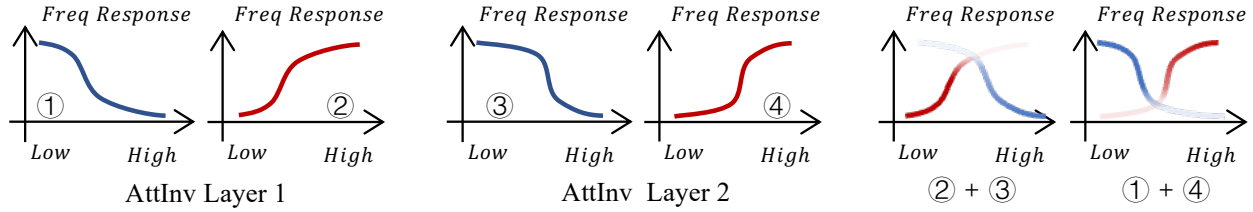


Figure 3. Illustration of stacking AttInv layers to obtain more complex filters. We illustrate how to use two layers of simple combinations of low-pass and high-pass filters to obtain more complex band-pass and band-stop filters. By stacking deeper layers in this way, a more complex frequency response can be achieved.

C. Implementation of Attention Invertion

Here, we introduce an efficient and simple way to implement Attention Inversion.

Inverting Filters in Attention. In Section A, we interpreted attention as a spatial dynamic convolution (illustrated in Figure 1). Building on that, we consider the attention matrix $\mathbf{A}$ as a set of $H \times W$ linear filters [18], where each spatial location (p, q) corresponds to a filter $\hat{\mathbf{A}}_{p,q} \in \mathbb{R}^{H \times W}$.

Mathematically, inverting the filters in the attention matrix is expressed as:

$$\hat{\mathbf{A}}_{p,q} = \mathcal{F}^{-1}\left(\mathbf{I}_f - \mathcal{F}(\mathbf{A}_{p,q})\right), \tag{27}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier Transform and its inverse, respectively. $\mathbf{I}_f$ is an all-pass filter in the frequency domain, with real part 1 and imaginary part 0. This inversion ensures that the high-pass filter $\hat{\mathbf{A}}$ has a frequency response complementary to $\mathbf{A}_{p,q}$.

In the spatial domain, the all-pass filter corresponds to the identity filter, where the center value is 1, and all other values are zero, as shown in Figure 2. Thus, the inversion can be simplified to:

$$\hat{\mathbf{A}}_{p,q} = \mathbf{I} - \mathbf{A}_{p,q}. \tag{28}$$

Figure 2 illustrates this process. The identity filter exhibits an all-pass frequency response, while attention filters suppress high-frequency components, resulting in a low-pass effect. By subtracting these filters from the identity filter, we obtain their inverted counterparts, which emphasize high-frequency components. Combining the original and inverted filters with modulation weights allows us to flexibly adjust the frequency response, enabling selective enhancement or suppression of spatial structures:

$$\tilde{\mathbf{A}}_{p,q} = \bar{\mathbf{S}}(p,q) \cdot \mathbf{A}_{p,q} + \hat{\mathbf{S}}(p,q) \cdot \hat{\mathbf{A}}_{p,q}, \tag{29}$$

where $\bar{\mathbf{S}}$ and $\hat{\mathbf{S}}$ are the combination weights obtained via a convolutional layer. This spatially adaptive combination dynamically balances low- and high-pass filtering, ensuring that each region retains the most relevant frequency components. When cascading L such layers, the frequency response across L layers can be expressed as:

$$\mathcal{F}(\mathbf{X}^{(L)}) = \prod_{i=1}^{L} \left[\bar{\mathbf{S}}^{(i)}\mathcal{F}(\mathbf{A}^{(i)}) + \hat{\mathbf{S}}^{(i)}\mathcal{F}(\hat{\mathbf{A}}^{(i)})\right] \cdot \mathcal{F}(\mathbf{X}^{(0)}). \tag{30}$$

This recursive composition of these hybrid filters expands into L^2 distinct weighted combinations of low- and high-pass operations, enabling flexible amplification or suppression of specific frequency bands. As shown in Figure 3, we illustrate how to use two layers of simple combinations of low-pass and high-pass filters to obtain more complex band-pass and band-stop filters. By stacking deeper layers in this way, a more complex frequency response can be achieved.

In contrast, stacking L standard attention layers monotonically attenuates high frequencies, leading to exponential vanishing as Equation (26) described. AttInvs quadratic complexity in frequency operations preserves both global structures (via low-pass) and fine details (via high-pass), overcoming the spectral limitations of attention mechanism.

Reformulating AttInv with the Identity Matrix. To further simplify implementation, we introduce $\mathbf{E} \in \mathbb{R}^{H\times W}$, an identity matrix with 1 on the diagonal and 0 elsewhere. The high-pass filter can then be expressed as:

$$\hat{\mathbf{A}} = \mathbf{E} - \mathbf{A}. \tag{31}$$

Substituting this into the dynamic combination from Eq. (29), we get the reformulated AttInv output:

$$\begin{aligned}\mathbf{X}' &= \tilde{\mathbf{A}} \times \mathbf{X} \\ &= (\bar{\mathbf{S}} \cdot \mathbf{A} + \hat{\mathbf{S}} \cdot \hat{\mathbf{A}}) \times \mathbf{X} \\ &= \bar{\mathbf{S}} \cdot \mathbf{A} \times \mathbf{X} + \hat{\mathbf{S}} \cdot \hat{\mathbf{A}} \times \mathbf{X} \\ &= \bar{\mathbf{S}} \cdot \mathbf{A} \times \mathbf{X} + \hat{\mathbf{S}} \cdot (\mathbf{E} - \mathbf{A}) \times \mathbf{X} \\ &= \bar{\mathbf{S}} \cdot \mathbf{A} \times \mathbf{X} + \hat{\mathbf{S}} \cdot (\mathbf{X} - \mathbf{A} \times \mathbf{X}).\end{aligned} \tag{32}$$

where $\mathbf{X}'$ is the output, and $\mathbf{A} \times \mathbf{X}$ can be regarded as the low-pass filtered low-frequency component of $\mathbf{X}$ due to the low-pass filtering effect of attention. Thus, the residual $\mathbf{X} - \mathbf{A} \times \mathbf{X}$ can be regarded as the high-pass filtered result.

Simple Implementation of AttInv. On the basis of analysis above, the implementation of AttInv involves a straightforward two-step process: obtaining the high-pass filtered results and dynamically combining frequency components.

First, given an input feature $\mathbf{X}$, we compute the high-pass filtered output by subtracting the attention-filtered feature from the original feature:

$$\mathbf{X}_{\text{hf}} = \mathbf{X} - \mathbf{A} \times \mathbf{X}, \tag{33}$$

where $\mathbf{A} \times \mathbf{X}$ is the output of the attention matrix multiplied by the input, which extracts the low-frequency components of $\mathbf{X}$. This subtraction removes the low-frequency components, preserving the high-frequency details.

Next, we use a spatially varying modulation mechanism to balance low- and high-frequency components. We learn two spatially dynamic coefficients $\bar{\mathbf{S}}, \hat{\mathbf{S}} \in \mathbb{R}^{H\times W}$, which control the contribution of low- and high-frequency components at each spatial location:

$$\mathbf{X}' = \bar{\mathbf{S}} \cdot \mathbf{A} \times \mathbf{X} + \hat{\mathbf{S}} \cdot \mathbf{X}_{\text{hf}}, \tag{34}$$

where the modulation weights $\bar{\mathbf{S}}$ and $\hat{\mathbf{S}}$ are generated via a convolutional layer conditioned on $\mathbf{X}$.

This formulation allows the model to selectively emphasize different frequency components based on local spatial structures, ensuring effective preservation of fine-grained details while maintaining smoothness in homogeneous regions.

D. Experimental Settings

Datasets and Metrics. We evaluate our methods on three challenging benchmarks: ADE20K [22], and COCO [12], and DOTA [20].

ADE20K. ADE20K [22] is a more diverse semantic segmentation dataset containing 150 categories. It includes 20,210 training images, 2,000 validation images, and 3,352 test images. This dataset covers a wide range of scenes, making it ideal for evaluating the generalizability of segmentation models.

COCO. For object detection, instance segmentation, and panoptic segmentation tasks, we use the COCO dataset [12], a standard benchmark in these domains. COCO provides a comprehensive set of annotations for object instances and is widely used to evaluate model performance on detection and segmentation tasks.

DOTA. For remote sensing object detection, we use the DOTA-v1.0 [20] dataset, which consists of 2,806 remote sensing images. It contains 188,282 instances across 15 categories: Plane (PL), Baseball diamond (BD), Bridge (BR), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Ship (SH), Tennis court (TC), Basketball court (BC), Storage tank (ST), Soccer-ball field (SBF), Roundabout (RA), Harbor (HA), Swimming pool (SP), and Helicopter (HC).

Metrics. For evaluation metrics, we use mean Intersection over Union (mIoU) to assess semantic segmentation performance. For object detection and instance segmentation, we use Average Precision (AP), which measures the accuracy of detection models across different categories and scales. For panoptic segmentation, we use Panoptic Quality (PQ) [8]. The GLOPS are measured on an image size of 512×2048, except for SegFormer, which uses an image size of 512×512.

Implementation Details. For Mask DINO [10], we adhere to the original training protocols [10], with the exception of setting the batch size to 8 instead of 16. For UPerNet with ViT [5], all models are trained for 160k iterations using AdamW [13] with a batch size of 16. On COCO, we follow standard practices [4, 15, 19] 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 models that do not adopt the standard self-attention mechanism, such as LSKNet [11], we bypass the AttInv module of FDAM when combining it. The results of the competitors in the tables are copied from their original papers and the widely used standard benchmark codebase, mmsegmentation.

E. Ablation Study

In this section, we conduct an ablation study to investigate the impact of each component in our Frequency-

Table 1. Ablation studies of our Frequency Dynamic Attention Modulation (FDAM) on the ADE20K validation set [22], showcasing the integration of Attention Inversion (AttInv) and Frequency Dynamic Scaling (FreqScale).

Models	Params	mIoU
DeiT-S [17]	52.1M	42.9
+ AttInv	+ 0.1M	43.5 (+0.6)
+ AttInv + FreqScale (Static)	+ 1.2M	43.8 (+0.9)
+ AttInv + FreqScale (Dynamic)	+ 4.4M	44.1 (+1.2)
+ AttInv + FreqScale (Dynamic + Group)	+ 0.5M	**44.3 (+1.4)**

Table 2. Ablation study on the number of static frequency scaling weights (n) in FreqScale. The results are reported on the ADE20K validation set [22].

Number of Weights (n)	$n = 2$	$n = 4$	$n = 8$	$n = 16$
mIoU	44.0	**44.3**	44.1	43.8

Table 3. Ablation study on the group of static frequency scaling weights (g) in FreqScale. The results are reported on the ADE20K validation set [22].

Group of Weights (g)	$g = 4$	$n = 8$	$n = 16$	$n = 32$
mIoU	44.0	44.1	**44.3**	44.0

Table 4. Inference Speed Evaluation We apply our method to the widely used ViT model DeiT [17]. The frames per second (FPS) results are measured on a single RTX 3090 with a batch size of 1 and an image size of 512×2048. AttInv has a minor impact on FPS, while FreqScale introduces a slight overhead due to the DFT operation, as speed optimizations for frequency transformations such as FFT/iFFT have not yet been fully implemented.

Model	DeiT-S-MLN	+AttInv	+AttInv+FreqScale (Ours)
FPS	8.3	8.1	7.36
Time	121.1ms	124.2ms	135.8ms

Dynamic Attention Modulation (FDAM) mechanism on the performance of semantic segmentation. The study is performed on the ADE20K validation set [22] using the Vision Transformer (DeiT-S) [17] as the base model. We evaluate the contributions of two key components of FDAM: Attention Inversion (AttInv) and Frequency Dynamic Scaling (FreqScale), as well as the effectiveness of dynamic frequency scaling and group-based frequency modulation.

Impact of Attention Inversion (AttInv). The first component of FDAM, AttInv, aims to mitigate the low-pass filtering effect inherent in the self-attention mechanism of Vision Transformers. The results show that incorporating AttInv leads to an improvement of +0.6 mIoU, increasing the baseline mIoU from 42.9 to 43.5. Adding AttInv incurs a minimal parameter increase of only 0.1M, as shown in Table 1. This demonstrates that AttInv effectively enhances the models frequency representation by introducing high-pass filtering components, which help preserve fine-grained details and textures that are otherwise lost due to the low-pass nature of standard attention.

Effect of Frequency Dynamic Scaling (FreqScale). We next investigate the contribution of FreqScale, which refines the frequency representation by dynamically adjusting the scaling of frequency bands. FreqScale is implemented in two forms: static and dynamic. In the static version, the frequency scaling weights are learned and remain fixed throughout training. The dynamic version, on the other hand, computes scaling weights adaptively.

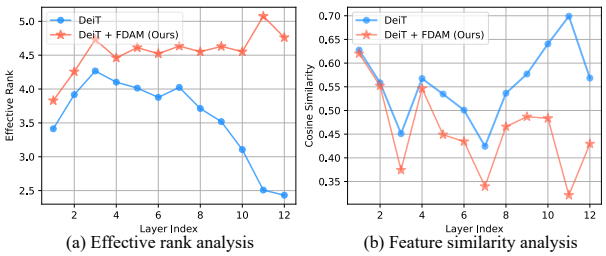
As shown in Table 1, when implemented with static scaling weights, it adds 1.2M parameters and boosts mIoU by +0.9 (to 43.8). In contrast, the fully dynamic version adds 4.4M parameters and achieves a slightly higher mIoU improvement of +1.2 (to 44.1), indicating that adaptive scaling can further enhance performance, albeit with higher parameter overhead.

Effectiveness of Group-wise Reassembly Strategy. To further optimize the frequency modulation process, we introduce a group-wise dynamic frequency scaling strategy, where the static scaling weights are grouped and combined with dynamic coefficients to generate efficient frequency scaling weights. This method reduces the parameter overhead compared to the fully dynamic approach while still providing the flexibility to adjust frequency components. As shown in Table 1, this approach adds only 0.5M parameters and attains the best performance, with an mIoU increase of +1.4 (to 44.3). The group-wise strategy thus provides an optimal trade-off between parameter efficiency and performance gain.

Number of Static Frequency Scaling Weights (n). Table 2 shows the results when varying n over $\{2, 4, 8, 16\}$. With $n = 2$, the model achieves an mIoU of 44.0. Increasing n to 4 yields the best performance with an mIoU of 44.3, indicating that four static weights provide an optimal balance between capturing diverse frequency components and keeping the parameter cost moderate. When n is increased to 8, the performance slightly drops to 44.1, and at $n = 16$ the mIoU decreases to 43.8. These results suggest that $n = 4$ offers the best accuracy while avoiding over-parameterization.

Group of Static Frequency Scaling Weights (g). Table 3 presents the effect of varying the Group number g among $\{4, 8, 16, 32\}$. The mIoU scores obtained are 44.0, 44.1, 44.3, and 44.0, respectively. The highest performance is achieved with $g = 16$, indicating that grouping the static scaling weights into 16 groups strikes an effective balance between dynamic reassembly flexibility and parameter efficiency. Both lower ($g = 4$ and $g = 8$) and higher ($g = 32$) grouping configurations result in slightly inferior performance.

Inference Speed Evaluation. Table 4 presents an evaluation of the inference speed on the widely used DeiT-S-MLN model [17] when integrated with our FDAM components. The measurements are taken on a single RTX 3090 with a



(a) Effective rank analysis (b) Feature similarity analysis

Figure 4. (a) Effective rank analysis for feature rank collapse. Higher *effective rank* [6] indicates a greater ability to capture complex patterns and nuanced details from the input data. FDAM maintains a consistently higher effective rank across all layers compared to the DeiT model using standard attention, demonstrating enhanced expressiveness of the attention mechanisms. (b) Feature similarity analysis. The cosing similarity increases with depth in the baseline DeiT model, indicating a loss of diversity in patch representations [14, 18]. The proposed FDAM method largely reduces this similarity, promoting more diverse features.

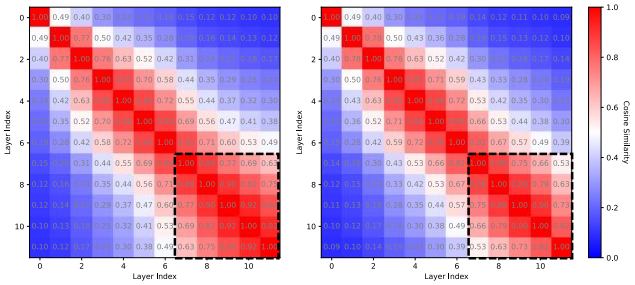


Figure 5. Cross-layer similarity analysis. The left shows the vanilla DeiT model [17], and the right shows DeiT combined with the proposed FDAM method.

batch size of 1 and an input image size of 512×2048. The baseline DeiT-S-MLN model achieves 8.3 FPS (121.1 ms per image). Incorporating AttInv leads to a slight decrease in speed to 8.1 FPS (124.2 ms per image), indicating a minimal computational overhead. When both AttInv and FreqScale are applied (our full FDAM), the FPS further drops to 7.36 (135.8 ms per image). This additional overhead is mainly due to the DFT operations required by FreqScale, as optimizations such as FFT/iFFT have not yet been fully implemented. Overall, the modest inference overhead introduced by FDAM is well justified by the substantial performance improvements in dense prediction tasks.

F. Effective Rank Analysis

The *effective rank* [6] is a measure that captures the diversity of singular values in a matrix, providing a more nuanced understanding of the matrixs structure and its implications for model behavior. Unlike the traditional rank, which counts the number of nonzero singular values, the effective rank takes their magnitudes into account, offering deeper insights into the matrixs capacity to represent diverse features.

Mathematically, the effective rank r_{eff} is defined as the Shannon entropy of the normalized singular values of a matrix [6]:

$$r_{\text{eff}} = \exp\left(-\sum_{i=1}^{n} \sigma_i \log \sigma_i\right), \tag{35}$$

where σ_i are the normalized singular values of the matrix (*e.g.*, a given feature map). A higher effective rank indicates a more uniform distribution of singular values, suggesting a greater capacity to capture diverse features.

As shown in Figure 4(a), the effective rank of the DeiT [17] model decreases rapidly with depth, indicating a loss of feature anisotropy and limiting the models ability to capture complex patterns. In contrast, our FDAM maintains a consistently higher effective rank across all layers, demonstrating its effectiveness in mitigating rank collapse and enhancing the expressiveness of the attention mechanism.

G. Feature Similarity Analysis

Understanding the similarity of feature representations across different layers is crucial for analyzing model behavior, particularly in ViTs. We employ a similarity metric based on pairwise cosine similarity to quantify how feature representations evolve through the network.

Following the methodology in [2, 18], we compute the average pairwise cosine similarity between token representations at each layer. Formally, given a layer index l and its corresponding feature matrix $X^{(l)} \in \mathbb{R}^{n \times d}$, the feature similarity $M_{\text{feat}}^{(l)}$ is computed as:

$$M_{\text{feat}}^{(l)} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{\left| X_{i,:}^{(l)} \cdot X_{j,:}^{(l)} \right|}{\|X_{i,:}^{(l)}\|_2 \|X_{j,:}^{(l)}\|_2}, \tag{36}$$

where $X_{i,:}^{(l)}$ denotes the i-th row of $X^{(l)}$, representing the feature vector of the i-th token. This metric captures how similar token representations are at a given layer, helping to diagnose potential oversmoothing issues.

To further investigate the similarity patterns, we visualize the evolution of feature similarity across layers for different architectures.

We assess our model's feature similarity using cosine similarity across layers in Figure 4(b). The DeiT shows a sharp rise in patch-wise cosine similarity with depth, hitting 0.70 by layer 11, signaling feature homogenization from repeated self-attention operations that erode discriminative spatial information, which aligns with previous findings on representation collapse in deep ViTs [14]. Our FDAM reduces late-layer similarity by up to 35%, enhancing robustness and task performance through more diverse representations. This analysis shows our methods curb over-

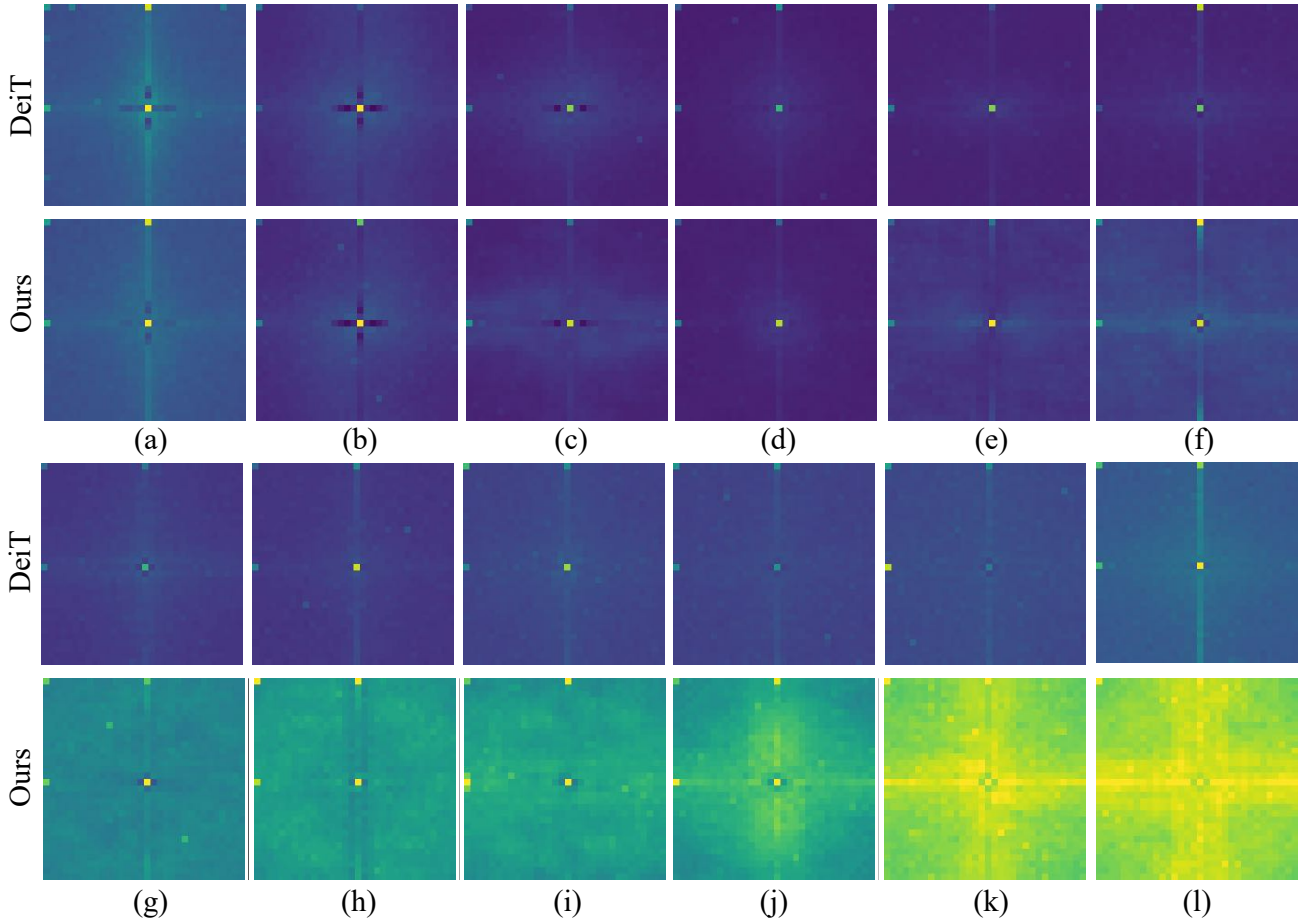


Figure 6. Layer-wise comparison of the two-dimensional frequency response in DeiT. Warmer colors denote higher response. Panels (a)-(l) show the frequency responses of feature maps from all 12 layers. In the early layers (a)-(d), our method and DeiT exhibit similar spectral characteristics. However, from the fifth layer onward (e)-(l), the difference becomes increasingly pronounced: our method consistently maintains higher energy in the high-frequency range, while DeiT progressively suppresses high-frequency details. This suggests that our approach effectively mitigates over-smoothing, preserving richer structural information across layers.

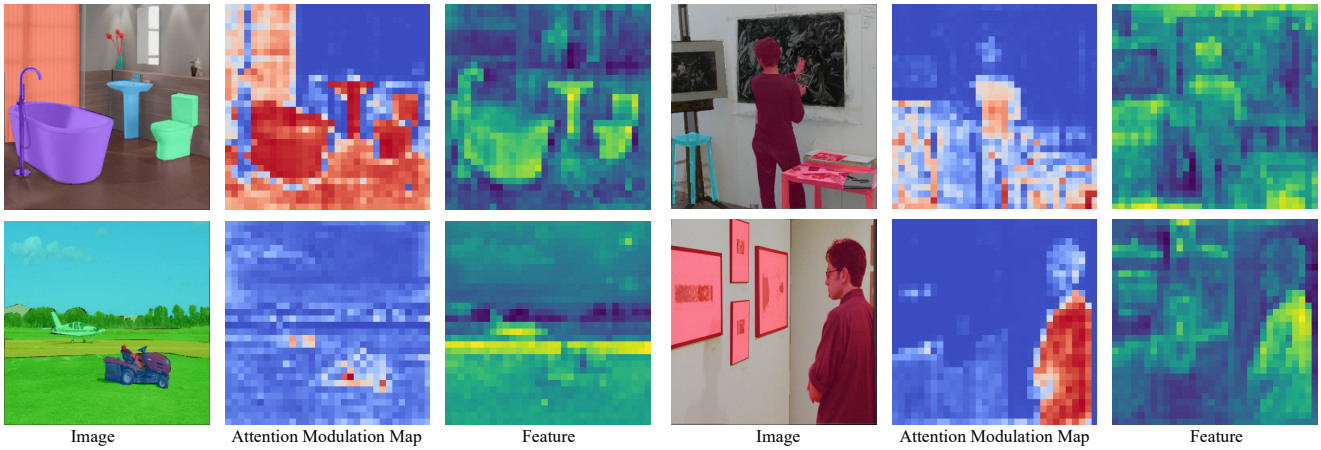


Figure 7. Visualization of attention modulation learned by AttInv. Warmer colors indicate higher values for high-pass filters. AttInv tends to assign higher values to foreground regions and semantic edges, emphasizing the focus on salient objects and boundaries.

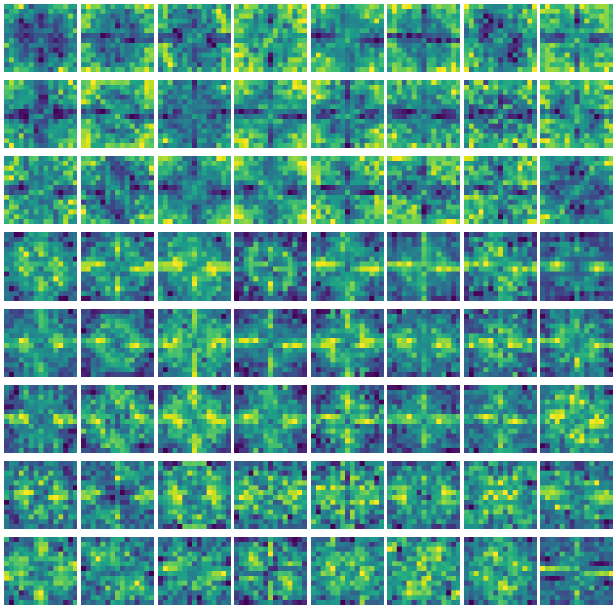


Figure 8. Visualization of frequency modulation map learned by FreqScale. From the center to the border are low- to high-frequency components. Brighter colors highlight amplified frequency components. We observe that the center appears relatively dim, implying a low modulation value for the low-frequency component. This demonstrates that FreqScale tends to enhance mid-to-high-frequency components in the feature maps, effectively preventing over-smoothing caused by the attention mechanism.

smoothing, promote diversity, improve representational capacity, and boost performance on vision tasks.

We further conduct a cross-layer similarity analysis, which reveals systematic patterns in feature representations across network depths. As illustrated in Figure 5, the last five layers of DeiT show saturated similarity ($\geq$0.63), implying that they do not explore new features or information as the model goes deeper. Our method alleviates this phenomenon and reduces the cosine similarity by about 0.1 for the last five layers, demonstrating that the proposed method encourages the model to explore new features.

H. More Visulization Results

Frequency Response Visualization. We analyze the frequency spectra of feature maps across all 12 layers of DeiT to gain insights into the frequency characteristics of self-attention. Figure 6 compares the frequency responses layer by layer.

Subfigures (a)-(l) show the frequency responses from the 12 layers. In early layers (a)-(d), the spectral distributions of DeiT and our method are similar, indicating minimal impact on initial feature extraction. From the fifth layer (e) onwards, spectral differences become more pronounced. While DeiT suppresses high-frequency components, our method preserves significantly more high-frequency energy. This gap widens in deeper layers, demonstrating our approach's ability to counteract the over-smoothing effect of standard self-attention.

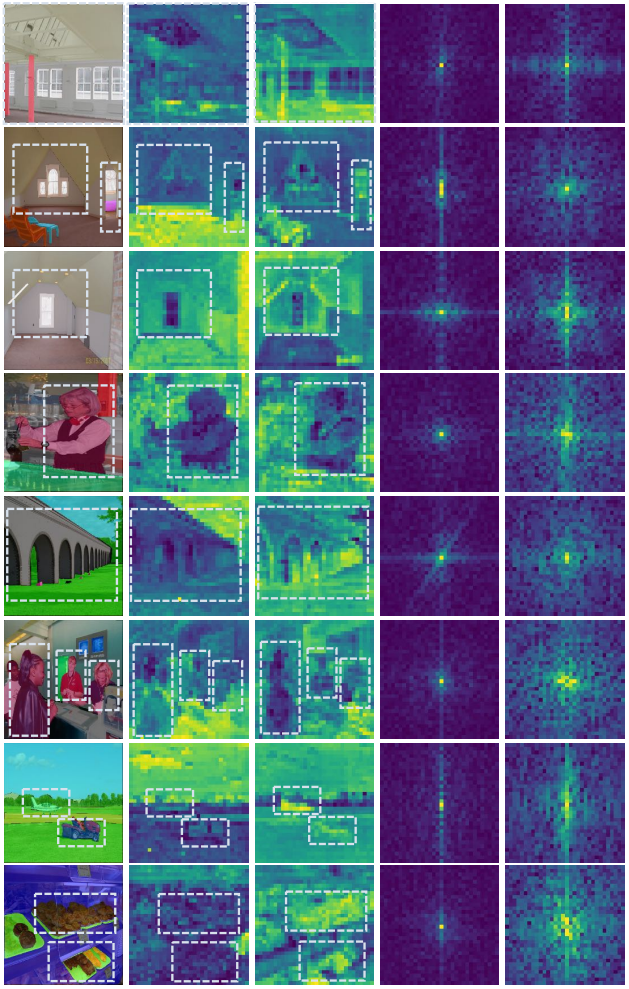


Figure 9. Feature and spectrum visualization. (a) Input image. (b), (c) feature maps. (d), (e) feature spectrum. Our approach generates semantically focused activations (c). Compared to DeiT's feature maps (b), our feature maps (c) capture sharper, more discriminative details, emphasizing object structures. The spectrum (d) shows DeiT's dominance in low-frequency components, whereas our method exhibits stronger high-frequency components, indicating better edge and detail preservation.

These results align with the proof in Section B that vanilla self-attention acts as a low-pass filter, discarding fine-grained details. Our method, by preserving more high-frequency components, maintains a richer spectral distribution, enhancing feature representations for tasks requiring fine details, such as fine-grained recognition and dense prediction.

Visualization of AttInv. Figure 7 illustrates the behavior

of AttInv, showing that it assigns higher high-pass filter values to foreground regions and semantic edges in the feature map. This highlights the models emphasis on discriminative objects and boundaries, which are crucial for capturing fine-grained textures and detailed information. By allocating more attention to these regions, AttInv enhances the model's ability to focus on spatially informative areas, improving the quality of feature representations for tasks that require precise recognition of objects and structures.

Visualization of FreqScale. Figure 8 visualizes the frequency modulation map learned by FreqScale. The map arranges frequency components spatially, with low frequencies concentrated at the center and high frequencies at the borders. Brighter areas represent amplified frequency components. The center of the map appears relatively dim, indicating that low-frequency components receive lower modulation values. This pattern suggests that FreqScale prioritizes mid-to-high-frequency components in the feature maps. By enhancing these frequencies, FreqScale counteracts the over-smoothing typically induced by self-attention, effectively preserving crucial structural information and enabling more detailed feature extraction for dense prediction tasks, where fine-grained detail is essential.

Feature Visualization. Figure 9 compares feature visualizations between DeiT and our method. DeiT features tend to blur important details and textures due to its inherent low-pass filtering characteristic. This results in a loss of fine-grained information, which is essential for tasks that demand high-precision visual understanding. In contrast, our method generates sharper, more discriminative feature maps, with enhanced emphasis on object structures. The frequency spectrum of DeiT predominantly shows a bias toward low-frequency components, reflecting its tendency to suppress high-frequency details. On the other hand, our feature spectrum exhibits a more balanced distribution across frequency bands, indicating a better preservation of fine-grained details and localized features, which are critical for tasks requiring precise spatial discrimination and detailed object recognition.

References

[1] Gregory A Baxes. *Digital image processing: principles and applications*. John Wiley & Sons, Inc., 1994. 3

[2] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pages 2793–2803. PMLR, 2021. 7

[3] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009. 2

[4] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. 2022. 5

[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. 5

[6] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, pages 1–15, 2023. 7

[7] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Proceedings of Advances in Neural Information Processing Systems*, 29:1–9, 2016. 2

[8] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 5

[9] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 12321–12330, 2021. 2

[10] 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. 5

[11] Yuxuan Li, Xiang Li, Yimain Dai, Qibin Hou, Li Liu, Yongxiang Liu, Ming-Ming Cheng, and Jian Yang. Lsknet: A foundation lightweight backbone for remote sensing. *International Journal of Computer Vision*, pages 1–22, 2024. 5

[12] 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. 5

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[14] Tam Nguyen, Tan Nguyen, and Richard Baraniuk. Mitigating over-smoothing in transformers via regularized nonlocal functionals. *Proceedings of Advances in Neural Information Processing Systems*, 36:80233–80256, 2023. 7

[15] 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. 5

[16] Steven W Smith et al. The scientist and engineer's guide to digital signal processing, 1997. 1

[17] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 6, 7

[18] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*. 4, 7

[19] 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. 5

[20] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018. 5

[21] Chii-Huei Yu. The application of parsevals theorem to integral problems. *Applied mathematics and physics*, 2(1):4–9, 2014. 3

[22] 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. 5, 6

[23] Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang. Decoupled dynamic filter networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6656, 2021. 2