Enhancing Vision Transformer: Amplifying Non-Linearity in Feedforward Network Module

Yixing Xu [1] **Chao Li** [1] **Dong Li** [1] **Xiao Sheng** [1] **Fan Jiang** [1] **Lu Tian** [1] **Ashish Sirasao** [1] **Emad Barsoum** [1]

Abstract

Transformer models have been gaining substantial interest in the field of computer vision tasks nowadays. Although a vision transformer contains two important components which are self-attention module and feedforward network (FFN) module, the majority of research tends to concentrate on modifying the former while leaving the latter in its original form. In this paper, we focus on improving the FFN module within the vision transformer. Through theoretical analysis, we demonstrate that the effect of the FFN module primarily lies in providing non-linearity, whose degree corresponds to the hidden dimensions. Thus, the computational cost of the FFN module can be reduced by enhancing the degree of non-linearity in the nonlinear function. Leveraging this insight, we propose an improved FFN (IFFN) module for vision transformers which involves the usage of the arbitrary GeLU (AGeLU) function and integrating multiple instances of it to augment non-linearity so that the number of hidden dimensions can be effectively reduced. Besides, a spatial enhancement part is involved to further enrich the non-linearity in the proposed IFFN module. Experimental results show that we can apply our method to a wide range of state-of-the-art vision transformer models irrespective of how they modify their self-attention part and the overall architecture, and reduce FLOPs and parameters without compromising classification accuracy on the ImageNet dataset.

1. Introduction

Transformer models with self-attention operation have been applied to the field of computer vision (Han et al., 2022) and achieve impressive results on many tasks such as image classification (Dosovitskiy et al., 2021; Touvron et al., 2021; Wu et al., 2023; Guo et al., 2024), object detection (Fang et al., 2021; Zheng et al., 2023), semantic segmentation (Strudel et al., 2021) and video analysis (Neimark et al., 2021) nowadays. Compared to convolutional neural networks (CNNs), transformer models have less inductive bias due to the low-pass filter property of the self-attention (Park & Kim, 2022) and have the capability to utilize more training data to enhance generalization ability (Chen et al., 2024). However, when given a limited amount of training data, the original Vision Transformer (ViT) model (Dosovitskiy et al., 2021) cannot perform on par with state-of-the-art CNN models (Hao et al., 2023), making it difficult to apply ViT to complicated vision tasks.

The modification of the vanilla ViT model primarily lies in two parts. The first one is to change the basic architecture of ViT. Hierarchical ViTs (Heo et al., 2021; Liu et al., 2021; Xu et al., 2023) leverage the advantage of hierarchical architecture of CNNs and reduce the spatial size as well as expand the channel dimensions multiple times with the help of pooling layers. A convolution stem with multiple convolutional layers is introduced in (He et al., 2019) to replace the non-overlapping patch embedding operation. The second one is to modify the self-attention module in ViT. Local-enhanced vision transformers (Huang et al., 2021; Wu et al., 2022) constrain the range of attention and generate patches within a local region, and facilitate interactions between patches to extract and interpret global information. Efficient self-attention operations reduce computational complexity of previous operation from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ (Wang et al., 2020) or $\mathcal{O}(nlog(n))$ (Kitaev et al., 2020).

Although a substantial number of works concentrate on studying the variations of vision transformers, very few of them pay attention to modifying the feedforward network (FFN) module. CMT (Guo et al., 2022) uses an inverted residual feed-forward network to replace the original FFN module, CoAtNet (Dai et al., 2021) uses MBConv blocks (Sandler et al., 2018) to replace some of the ViT blocks in its network architecture. However, there are multiple modifications in their architectures and the effectiveness of modifying FFN module remains unclear. Furthermore, there is a lack of theoretical analysis explaining why these

[1]Advanced Micro Devices, Inc., Beijing, China. Correspondence to: Yixing Xu <yixing.xu@amd.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

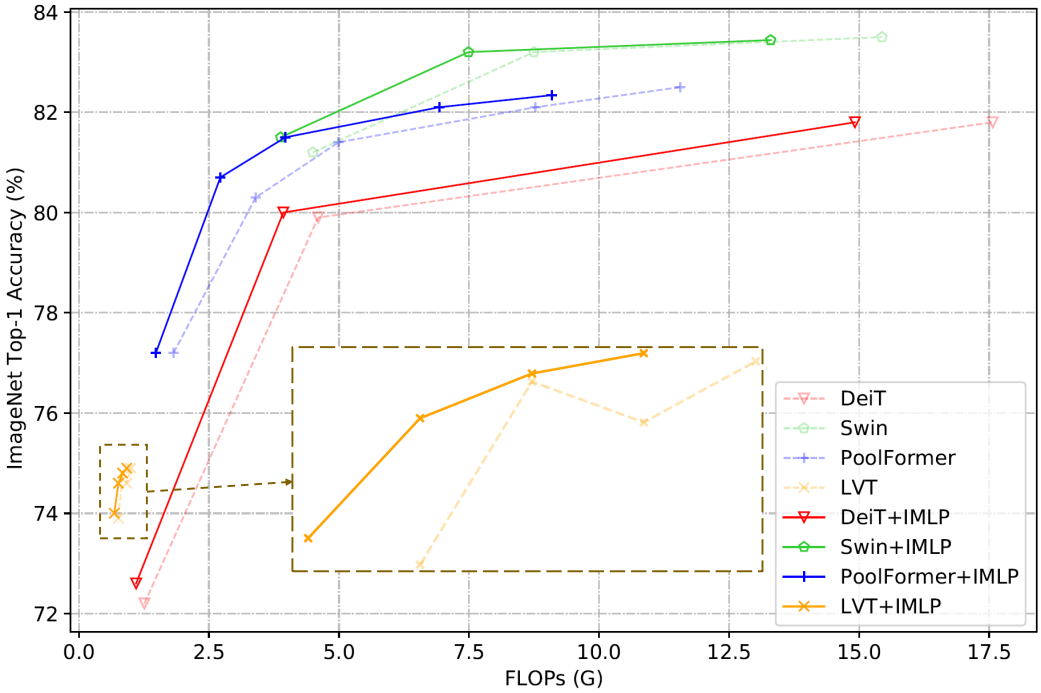


Figure 1. Top-1 classification accuracy versus FLOPs for different models on ImageNet-1k dataset. Our IFFN module can reduce FLOPs without sacrificing classification performance on different baseline vision transformer models.

changes are effective.

In this paper, we first give a thorough analysis of the FFN module in the vision transformer and show that the effect of the FFN module primarily lies in providing non-linearity whose degree corresponds to the hidden dimensions. Then, if we can enhance the degree of non-linearity in the non-linear function, we could potentially decrease the hidden dimensions of the FFN module, thereby reducing the computational cost. Based on this thought, we introduce the arbitrary GeLU (AGeLU) function which is easy to combine to generate stronger non-linearity. Besides, a spatial-wise enhancement part is added to further enrich the non-linearity of the module. By combining them together, we introduce our improved FFN (IFFN) module for vision transformer. We conduct several experiments on different popular vision transformer models with various designs of the whole architecture and self-attention module including DeiT, Swin, PoolFormer, LVT, *etc.*, by replacing their original FFN module into the proposed IFFN module. Results on ImageNet-1k dataset show that we can effectively reduce FLOPs and parameters without sacrificing the classification accuracy as shown in Fig. 1 and the experiment section.

2. Related Works

Vision transformer (ViT) was first introduced by (Dosovitskiy et al., 2021) to extend the transformer architecture to vision tasks. Since then, researches have focused on improving the performance of vanilla ViT. For example, DeiT (Touvron et al., 2021) leveraged the knowledge distillation method and introduced a series of new training techniques to enhance the classification performance on ImageNet-1k. Swin (Liu et al., 2021) utilized a hierarchical architecture and adopted a local self-attention mechanism to reduce computational complexity while using shift operation to add interaction across different sub-windows. PoolFormer (Yu et al., 2022) argued that the whole architecture of ViT was more important than the self-attention operation and replaced the multi-head self-attention (MHSA) modules with pooling operations.

Methods mentioned above focus on modifying the training strategy, the whole architecture of ViT and the MHSA module (Tang et al., 2024). Very little research studied the FFN module in ViT. CMT (Guo et al., 2022) and PVTv2 (Wang et al., 2022) introduced ViT models with several modifications, and one of them was to use the inverted residual feed-forward network to replace the original FFN module. CoAtNet (Dai et al., 2021) found that vertically stacking convolution layers and attention layers was surprisingly effective and replaced some of the ViT blocks with MBConv blocks through neural architecture search method. These studies generated new ViT models with various modifications to the fundamental architecture. However, the impact of altering the FFN module alone remains uncertain.

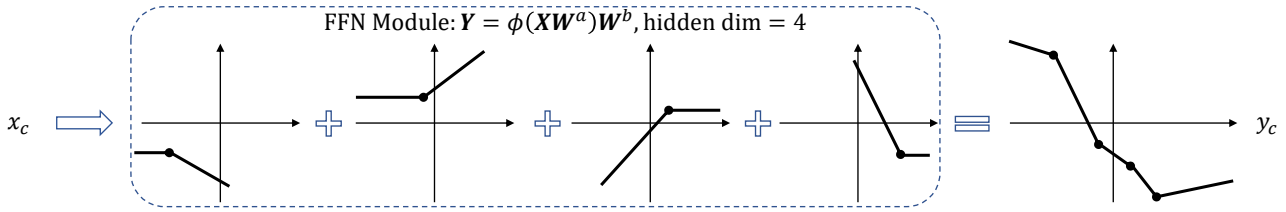


Figure 2. An intuitive illustration of the corollary that the FFN module is a non-linearity generator. We use $\phi(\cdot) = \mathrm{ReLU}(\cdot)$ in this figure for simplicity. Other formats of nonlinear function can also be used here to derive the same conclusion.

3. FFN Module is a Non-linearity Generator

Considering an input matrix $\boldsymbol{X} \in \mathbb{R}^{N\times C}$ in which N is the number of patches and C is the dimension of each patch, the output of the FFN module can be calculated as:

$$\boldsymbol{Y} = \mathrm{FFN}(\boldsymbol{X}) = \phi(\boldsymbol{X}\boldsymbol{W}^a)\boldsymbol{W}^b, \tag{1}$$

where $\boldsymbol{W}^a = \{w^a_{ij}\} \in \mathbb{R}^{C\times C'}$ and $\boldsymbol{W}^b = \{w^b_{ij}\} \in \mathbb{R}^{C'\times C}$ are weight matrices of two FC layers, C' controls the number of hidden dimensions, and $\phi(\cdot)$ represents the non-linear function. $C' = 4C$ and $\phi(\cdot) = \mathrm{GeLU}(\cdot)$ are used in the original ViT model.

Without loss of generality, we assume $N = 1$ and the input matrix $\boldsymbol{X}$ degrades into an input vector $\boldsymbol{x} \in \mathbb{R}^C$. Then, we can represents Eq. 1 in its element-wise form:

$$\begin{aligned}
\boldsymbol{x}\boldsymbol{W}^a &= \left(\sum_{i=1}^{C} w^a_{ic'}x_i\right)_{c'=1}^{C'}, \\
\phi(\boldsymbol{x}\boldsymbol{W}^a) &= \left(\phi(\sum_{i=1}^{C} w^a_{ic'}x_i)\right)_{c'=1}^{C'}, \\
\boldsymbol{y} = \phi(\boldsymbol{x}\boldsymbol{W}^a)\boldsymbol{W}^b &= \left(\sum_{j=1}^{C'} w^b_{jc}\phi(\sum_{i=1}^{C} w^a_{ij}x_i)\right)_{c=1}^{C} \\
&= \left(\sum_{j=1}^{C'} w^b_{jc}\phi(m_{cj}x_c + n_{cj})\right)_{c=1}^{C},
\end{aligned} \tag{2}$$

in which $m_{cj} = w^a_{cj}$ and $n_{cj} = f(x_1, \cdots, x_{c-1}, x_{c+1}, \cdots, x_C) = \sum_{i=1, i\neq c}^{C} w^a_{ij}x_i$. Given Eq. 2, we can derive the following corollary:

Corollary 3.1. *Given an input vector $\boldsymbol{x} \in \mathbb{R}^C$, the output of the FFN module in Eq. 1 is denoted as $\boldsymbol{y} \in \mathbb{R}^C$. Then:*

***(1)** Each element y_c in $\boldsymbol{y}$ is the linear combination of C' different nonlinear functions to the input element x_c.*

***(2)** Distinct scales and biases are applied to different input elements x_c before passing through the nonlinear function $\phi(\cdot)$.*

***(3)** The scale is a learnable weight independent to the input element x_c, while the bias is dependent to all other input elements in $\boldsymbol{x}$.*

The above conclusion brings to light that the FFN module in the vision transformer is no more than a non-linearity generator with a nonlinear degree of C', as intuitively shown in Fig. 2.

4. Method

4.1. A More Powerful Nonlinear Function

Based on the corollary in the previous section, a straightforward way is to use the combination of C' different nonlinear functions to replace the original FFN module. However, the bias which depends on the input elements makes it challenging to attain a comparable degree of non-linearity by merely combining multiple nonlinear functions, and the classification performance does not match that of using the original FFN module (as shown in Tab. 5).

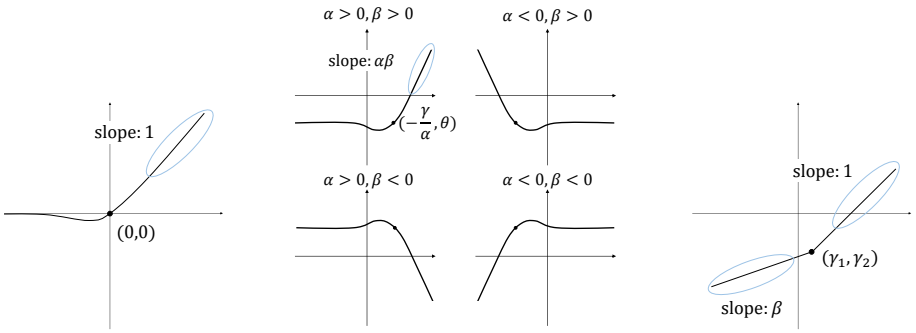
In the following paragraph, we first introduce the arbitrary nonlinear function which is flexible and easy to be concatenated together to form a more powerful nonlinear function. Subsequently, we demonstrate that the hidden dimension of the FFN module can be effectively reduced with this enhanced nonlinear function.

Arbitrary nonlinear function. Arbitrary nonlinear function is defined as

$$\phi'(x) = \beta\phi(\alpha x + \gamma) + \theta, \tag{3}$$

in which x is the input of the arbitrary nonlinear function, α and β are learnable coefficients before and after applying the basic nonlinear function $\phi(\cdot)$, and γ and θ are learnable biases. The inspiration for introducing arbitrary nonlinear function arises from Eq. 2 where distinct weights and biases are employed to each element x_c before and after applying the basic nonlinear function. Since GeLU is used as a basic nonlinear function in ViT, we introduce the arbitrary GeLU (AGeLU) to our model:

$$\mathrm{AGeLU}(x) = \beta\mathrm{GeLU}(\alpha x + \gamma) + \theta. \tag{4}$$



(a) GeLU (b) AGeLU (c) RPReLU

Figure 3. The comparison among the shapes of GeLU, AGeLU and RPReLU.

AGeLU is more flexible than other modified nonlinear functions such as the RPReLU function proposed in ReActNet (Liu et al., 2020). The latter can only adjust the position of the turning point compared to PReLU, while AGeLU can also provide a learnable slope of the function and switch the whole shape by using different positive and negative coefficients α and β. Fig. 3 gives a comparison among the shapes of GeLU, AGeLU, and RPReLU. Note that other basic activation functions such as ReLU, PReLU, *etc*. can be extended using the same way as AGeLU to form AReLU and APReLU.

Reducing the hidden dimension of FFN module with powerful nonlinear function. Rather than using the original FFN module introduced in Eq. 1, we propose our AFFN module that integrates two AGeLU functions and forms a powerful nonlinear function to replace the original GeLU and halve the hidden dimension of the module. Specifically, we have:

$$\begin{aligned}\boldsymbol{Y}' &= \mathrm{AFFN}(\boldsymbol{X}) \\ &= \mathrm{concat}(\mathrm{AGeLU}(\boldsymbol{X}\boldsymbol{W}^d), \mathrm{AGeLU}'(\boldsymbol{X}\boldsymbol{W}^d))\boldsymbol{W}^e, \end{aligned} \quad (5)$$

where $\boldsymbol{W}^d = \{w^d_{ij}\} \in \mathbb{R}^{C\times\frac{C'}{2}}$ and $\boldsymbol{W}^e = \{w^e_{ij}\} \in \mathbb{R}^{C'\times C}$ are weight matrices of two FC layers, and $\mathrm{AGeLU}(\cdot)$ and $\mathrm{AGeLU}'(\cdot)$ are two nonlinear functions proposed in Eq. 4 with different parameters. With this simple modification, the first FC layer has half the output channels compared to the original FFN module, and can effectively reduce the FLOPs and parameters in the vision transformer model. In the following section, we show that the proposed AFFN module can also be treated as the linear combination of C' different nonlinear functions.

We can degrade the input matrix $\boldsymbol{X}$ into an input vector $\boldsymbol{x} \in \mathbb{R}^C$, and represent Eq. 5 in its element-wise form:

$$\boldsymbol{t}_0 = \boldsymbol{x}\boldsymbol{W}^d = \left(\sum_{i=1}^{C} w^d_{ic'} x_i\right)_{c'=1}^{\frac{C'}{2}},$$

$$\begin{aligned}\boldsymbol{t}_1 &= \mathrm{AGeLU}(\boldsymbol{t}_0) \\ &= \left(\beta_{c'}\mathrm{GeLU}(\alpha_{c'}\sum_{i=1}^{C} w^d_{ic'}x_i + \gamma_{c'}) + \theta_{c'}\right)_{c'=1}^{\frac{C'}{2}}, \\ \boldsymbol{t}'_1 &= \mathrm{AGeLU}'(\boldsymbol{t}_0) \\ &= \left(\beta'_{c'}\mathrm{GeLU}(\alpha'_{c'}\sum_{i=1}^{C} w^d_{ic'}x_i + \gamma'_{c'}) + \theta'_{c'}\right)_{c'=1}^{\frac{C'}{2}}, \\ \boldsymbol{t}_2 &= \mathrm{concat}(\boldsymbol{t}_1, \boldsymbol{t}'_1) \\ &= \left(\beta_{c'}\mathrm{GeLU}(\alpha_{c'}\sum_{i=1}^{C} w^d_{i,f(c')}x_i + \gamma_{c'}) + \theta_{c'}\right)_{c'=1}^{C'}, \\ \boldsymbol{y}' &= \boldsymbol{t}_2\boldsymbol{W}^e \\ &= \left(\sum_{j=1}^{C'} w^e_{jc}\cdot[\beta_j\mathrm{GeLU}(\alpha_j\sum_{i=1}^{C} w^d_{i,f(j)}x_i + \gamma_j) + \theta_j]\right)_{c=1}^{C}, \\ &= \left(\sum_{j=1}^{C'} w'^e_{jc}\mathrm{GeLU}(m'_{cj}x_c + n'_{cj}) + \theta_j\right)_{c=1}^{C}, \end{aligned} \quad (6)$$

where in the fourth line, we define $\alpha'_1, \cdots, \alpha'_{\frac{C'}{2}} \triangleq \alpha_{\frac{C'}{2}+1}, \cdots, \alpha_{C'}$ (the same to β', γ' and θ'), $f(x) = x - \frac{C'}{2}\cdot\mathbb{1}_{x>\frac{C'}{2}}$ in which $\mathbb{1}$ is the indicator function, $w'^e_{jc} = w^e_{jc}\cdot\beta_j$, $m'_{cj} = w^d_{c,f(j)}$ and $n'_{cj} = \mathrm{func}(x_1, \cdots, x_{c-1}, x_{c+1}, \cdots, x_C) = \sum_{i=1,i\neq c}^{C} w^d_{i,f(j)}x_i + \gamma_j$.

Note that compared to the original FFN module (Eq. 2), the form of the proposed AFFN module (Eq. 6) is almost the same and can also be treated as a generator that generates the same degree of non-linearity. According to Collary 3.1, each element y'_c in $\boldsymbol{y}'$ can also be treated as a linear combination of C' different nonlinear functions to the input element x_c, each with distinct scales and biases. Each scale is a learnable weight independent to the input while each bias is dependent on other input elements.

4.2. Theoretical Analysis

In this section, we analyze the Lipschitz constant of the proposed AFFN module. Note that the Lipschitz constant serves as a metric for assessing the network's stability by bounding the rate of output change in response to input perturbations, while also highlighting the network's susceptibility to adversarial attacks. Thus, it is beneficial to study the Lipschitz constant that contributes to improving the reliability of our module.

Firstly, we give the definition of a Lipschitz constant:

Definition 4.1. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous if there exists a non-negative constant L such that

$$||f(x) - f(y)||_2 \leq L||x - y||_2 \quad \text{for all } x, y \in \mathbb{R}^n, \quad (7)$$

among which the smallest L is called the Lipschitz constant of function f.

In the following paragraph, we present a lemma to describe the conceptualization of nonlinear activation functions, and then use a theorem to derive the bound on the Lipschitz constant of our proposed AFFN module.

Lemma 4.2. *(Fazlyab et al., 2019) Suppose $\varphi : \mathbb{R} \to \mathbb{R}$ is slope-restricted on $[p, q]$. Define the set*

$$\mathcal{T}_n = \{T \in \mathbb{S}^n | T = \sum_{i=1}^{n} \lambda_{ii} e_i e_i^\top, \lambda_{ii} \geq 0\}. \quad (8)$$

Then for any $T \in \mathcal{T}_n$ the vector-valued function $\phi(x) = [\varphi(x_1), \cdots, \varphi(x_n)]^\top : \mathbb{R}^n \to \mathbb{R}^n$ satisfies

$$\begin{bmatrix} x - y \\ \phi(x) - \phi(y) \end{bmatrix}^\top \begin{bmatrix} -2pqT & (p+q)T \\ (p+q)T & -2T \end{bmatrix} \begin{bmatrix} x - y \\ \phi(x) - \phi(y) \end{bmatrix} \geq 0$$

for all $x, y \in \mathbb{R}^n$.

It is easy to prove that our proposed AGeLU activation function satisfies the slope-restricted condition when the parameters α and β in Eq. 4 are finite. The matrix T is used for deriving the Lipschitz bound of the AFFN module in the following theorem.

Theorem 4.3. *Given the AFFN module described by $f(x) = W^1 \text{concat}(\phi_1(W^0 x + b^0), \phi_2(W^0 x + b^0)) + b^1$. Suppose $\phi_i(x) : \mathbb{R}^n \to \mathbb{R}^n = [\varphi_i(x_1), \cdots, \varphi_i(x_n)]$, where φ_i is slope-restricted on $[p_i, q_i], i \in \{1, 2\}$. Define $\mathcal{T}_n$ as in Eq. 8. Suppose there exists $\rho_1, \rho_2 > 0$ such that the matrix inequalities*

$$M(\rho_i, T) := \begin{bmatrix} -2p_i q_i {W^0}^\top T W^0 - \rho_i I_{n_0} & (p_i + q_i){W^0}^\top T \\ (p_i + q_i) T W^0 & -2T + {W^{1i}}^\top W^{1i} \end{bmatrix} \preceq 0, \quad i \in \{1, 2\}, \quad (9)$$

holds for some $T \in \mathcal{T}_n$, where $W^1 = [W^{11} \ W^{12}]$. Then $||f(x) - f(y)||_2 \leq (\sqrt{\rho_1} + \sqrt{\rho_2})||x - y||_2$ for all $x, y \in \mathbb{R}^{n_0}$.

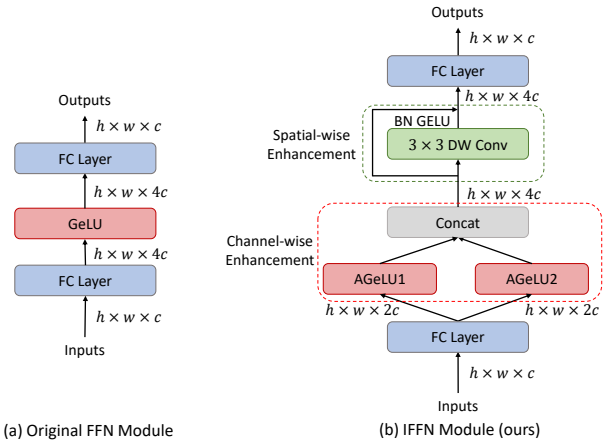


Figure 4. The architecture of (a) the original FFN module and (b) the proposed IFFN module. The channel-wise enhancement part includes the AGeLU function and concatenation operation. The spatial-wise enhancement part includes a depthwise block.

Theorem 4.3 gives an upper bound of $L(f) = \sqrt{\rho_1} + \sqrt{\rho_2}$ on the Lipschitz constant of the proposed AFFN module $f(x) = W^1 \text{concat}(\phi_1(W^0 x + b^0), \phi_2(W^0 x + b^0)) + b^1$. The above equation can be treated as a semi-definite program (SDP) which can be solved numerically to derive its global minimum. The proof of Theorem 4.3 is in the Appendix A.

4.3. Enhancing Non-linearity with Spatial Information

Although the AFFN module generates a same degree of non-linearity compared to the original FFN module, we notice that the degree of freedom of $\{w^d_{i,f(j)}\}_{j=1}^{C'}$ in Eq. 6 are halved compared to the original $\{w^a_{ij}\}_{j=1}^{C'}$ in Eq. 2. It is similar to the model quantization methods that halve the number of bits used for weights and activation and may degrade the performance.

In the previous section, we extend the non-linearity of the FFN module through the channel dimension. Therefore, in this section we further enhance non-linearity with spatial information. Many previous studies use convolution operation in vision transformers. For example, CMT (Guo et al., 2022) uses inverted residual FFN in the network, and CoAtNet (Dai et al., 2021) replaces some of the attention blocks with inverted bottlenecks. However, they do not mention the relationship between these blocks and the extension of non-linearity. VanillaNet (Chen et al., 2023) proposes series informed activation function to enrich the approximation ability which is formulated as:

$$\phi_s(x_{h,w,c}) = \sum_{i,j \in \{-n,n\}} a_{i,j,c} \phi(x_{i+h,j+w,c} + b_c), \quad (10)$$

Table 1. Image classification results on ImageNet-1k datasets. Several widely used state-of-the-art vision transformer models such as DeiT, Swin, PoolFormer and LVT are used as the baseline models, and the original FFN modules in them are replaced with the proposed IFFN module. The results are also intuitively shown in Fig. 1. '*' indicates that we use kernel size as 5 in the spatial-wise enhancement part.

Methods	Architecture	Parameters (M)	FLOPs (G)	Top-1 Accuracy (%)
DeiT	DeiT-Ti	5.72	1.26	72.2
	+ IFFN	5.00 (-12.6%)	1.10 (-12.7%)	72.6
	DeiT-S	22.05	4.60	79.9
	+ IFFN	18.84 (-14.6%)	3.93 (-14.6%)	80.0
	DeiT-B	86.57	17.57	81.8
	+ IFFN*	73.66 (-14.9%)	14.92 (-15.1%)	81.8
Swin	Swin-Ti	28.29	4.50	81.2
	+ IFFN	24.29 (-14.1%)	3.88 (-13.8%)	81.5
	Swin-S	49.61	8.75	83.2
	+ IFFN	42.40 (-14.5%)	7.49 (-14.4%)	83.2
	Swin-B	87.77	15.44	83.5
	+ IFFN*	75.45 (-14.0%)	13.34 (-13.6%)	83.4
PoolFormer	PoolFormer-S12	11.92	1.82	77.2
	+ IFFN	9.80 (-17.8%)	1.48 (-18.7%)	77.2
	PoolFormer-S24	21.39	3.40	80.3
	+ IFFN	17.15 (-19.8%)	2.72 (-20.0%)	80.7
	PoolFormer-S36	30.86	4.99	81.4
	+ IFFN	24.50 (-20.6%)	3.97 (-20.4%)	81.5
	PoolFormer-M36	56.17	8.78	82.1
	+ IFFN	44.19 (-21.3%)	6.93 (-21.1%)	82.1
	PoolFormer-M48	73.47	11.56	82.5
	+ IFFN*	58.62 (-20.2%)	9.46 (-18.2%)	82.3
Portable ViT	LVT-R1	5.52	0.76	73.9
	+ IFFN*	4.98 (-9.8%)	0.68 (-10.5%)	74.0
	LVT-R2	5.52	0.84	74.8
	+ IFFN*	4.98 (-9.8%)	0.76 (-9.5%)	74.6
	LVT-R3	5.52	0.92	74.6
	+ IFFN*	4.98 (-9.8%)	0.84 (-8.7%)	74.8
	LVT-R4	5.52	1.00	74.9
	+ IFFN*	4.98 (-9.8%)	0.92 (-8.0%)	74.9

where $\phi(\cdot)$ is the activation function. We found that this is equal to going through the non-linear function followed by a $n \times n$ depthwise convolution (DW Conv), which means that DW Conv after the non-linear function utilizes the spatial information and enhances non-linearity by learning global information from its neighbors. Thus, we modify our AFFN module by introducing a DW Block (DW Conv with BN and GeLU) after AGeLU, and form the final improved FFN (IFFN) module as shown in Fig. 4. The IFFN module has two main differences compared to the original FFN module. The first is the channel-wise enhancement part that includes the AGeLU function and concatenation operation proposed in section 4.1 to extend non-linearity through channel dimension. The second is the spatial-wise enhancement part with a DW Block to enhance non-linearity with spatial information.

5. Experiments

In this section, we conduct experiments on the ImageNet-1k dataset for image classification and then ablate different parts of IFFN through ablation studies. Experiments on object detection and semantic segmentation are shown in the Appendix B and C.

5.1. Image Classification on ImageNet-1k

We empirically verify the effectiveness of the proposed IFFN module on the ImageNet-1k dataset which contains

Table 2. Ablation study on channel/spatial-wise enhancement part. The experiments are conducted using the DeiT-Ti model on the ImageNet dataset.

Methods	Parameters (M)	FLOPs (G)	Top-1 Accuracy (%)
DeiT-Ti	5.72	1.26	72.2
w/ channel	4.89	1.08	70.5
w/ spatial	5.83	1.28	72.8
w/ channel & spatial	5.00	1.10	72.6

1.28M training images from 1000 different classes and 50K validation images.

Implementation details. We treat our IFFN module as a plug-in and replacement module that is used to replace the original FFN module in different vision transformers. Other parts of the architecture of baseline model are remain unchanged. The training strategies are exactly the same as the original methods.

Baseline models. We select several widely used state-of-the-art vision transformer models as our baseline models, including DeiT (Touvron et al., 2021), Swin (Liu et al., 2021), PoolFormer (Yu et al., 2022) and portable vision transformer such as LVT (Yang et al., 2022).

Experimental results. We replace all the FFN modules in each baseline method with the proposed IFFN module. The experimental results are shown in Tab. 1. We can see that almost all the models can reduce over 10% FLOPs and parameters without loss of classification accuracy. For example, we can reduce the parameter count of the DeiT-Ti model by 12.6% and FLOPs by 12.7% while increasing the top-1 accuracy by 0.4%. As the model becomes larger, the amount of parameter/FLOPs reduction also increases as the proportion of the FFN module in the computation grows. Similar results can be seen in other baseline models. PoolFormer models exhibit higher FLOPs and parameter reduction (over 20%) since most of their calculations come from the FFN module.

5.2. Ablation Studies

In this section, we ablate various design choices for each part of the IFFN module to empirically verify the effectiveness of the proposed method.

Effect of channel/spatial-wise enhancement part. In Tab. 2 we separately use channel-wise and spatial-wise enhancement parts in the IFFN module. When using channel-wise enhancement alone, there is a performance degradation compared to the baseline but the model has fewer FLOPs and parameters. When using spatial-wise enhancement alone, the model gets a better performance but with more computations. Combining the channel-wise and spatial-wise enhancement brings about a smaller model with better classification accuracy. Note that although the performance gain mainly comes from spatial-wise enhancement part, the proposed channel-wise enhancement part copes very well with the former one and can reduce computational cost while maintaining accuracy to the greatest extent possible. For example, using spatial part alone (Line 3) only increases 0.6% Top-1 accuracy compared to the original DeiT-Ti model (Line 1). However, by adding channel part, DeiT-Ti can increase 2.1% Top-1 accuracy while saving over 10% FLOPs and parameters (compare Line 2 and Line 4). Another straightforward example is to compare the settings that both have 72.8% top-1 accuracy in Tab. 2 and Tab. 6. In Tab. 2 the result of 72.8% using only the spatial-wise enhancement part, and the results of 72.8% in Tab. 6 using both channel-wise and spatial-wise enhancement parts with 5x5 dwconv, resulting in fewer FLOPs and parameters. This is an empirical result to show the usefulness of the combination of channel-wise and spatial-wise enhancement part. We can achieve the same classification performance with fewer FLOPs and parameters when combining the channel and spatial parts compared to using the spatial part alone.

Table 3. Ablation study on replacing GeLU with different activation functions with and without using the proposed IFFN module. The experiments are conducted using the DeiT-Ti model on the ImageNet dataset.

DeiT-Ti	Top-1 Acc (%)
w/ GeLU (original)	72.2
+IFFN	**72.6**
w/ SoftPlus (Nair & Hinton, 2010)	71.6
+IFFN	**72.0**
w/ ELU (Clevert et al., 2015)	71.1
+IFFN	**71.7**
w/ Swish (Ramachandran et al., 2017)	72.0
+IFFN	**72.5**

Table 4. Using different number of nonlinear functions $\#N$ for concatenation. The experiments are conducted based on the DeiT-Ti model.

$\#N$	FLOPs (G)	Params (M)	Top-1 Acc (%)
2	1.10	5.00	72.6
4	1.01	4.55	68.6

Table 5. Using the addition of $4C$ number of nonlinear functions to replace the original FFN module. GeLU and AGeLU are used as the basic nonlinear functions. The experiments are conducted based on the DeiT-Ti model.

Methods	Top-1 Acc (%)
original FFN	72.2
$\phi(\cdot) = \mathrm{GeLU}(\cdot)$	50.4
$\phi(\cdot) = \mathrm{AGeLU}(\cdot)$	53.3

Ablations on different activation functions. In Eq. 3 we propose a general arbitrary nonlinear function and instantiate it with AGeLU using Eq. 4. To better verify the effectiveness of the proposed method, we replace the original GeLU in DeiT with other popular activation functions such as SoftPlus (Nair & Hinton, 2010), ELU (Clevert et al., 2015) and Swish (Ramachandran et al., 2017). We can see that in Tab. 3, when using GeLU activation function (the original setting) and the SoftPlus function, the proposed IFFN module can have +0.4% accuracy improvements. The performance gains when using ELU and Swish are +0.6% and +0.5%, respectively. Since the computational cost of different activation functions are all negligible, the FLOPs and parameters reductions are 12.7% and 12.6% which are the same as in Tab. 1.

Effect of using different number of nonlinear functions for concatenation. In Eq. 5, AGeLU and AGeLU' are introduced as two nonlinear functions. In Tab. 4, we show the results of using more nonlinear functions. It is not surprise that the performance dropped as the number of nonlinear functions N increases, since the degree of freedom of $\{w^d_{i,f(j)}\}_{j=1}^{C'}$ in Eq. 6 should be divided by N and impact the final classification performance as we analyze at the beginning of Sec. 4.3.

Effect of directly combining multiple nonlinear functions. The analysis in Sec. 3 shows that the FFN module in vision transformer is no more than a non-linearity generator that can be treated as a linear combination of $C' = 4C$ different activation functions. Thus, a straight-forward way to replace the FFN module is to directly using $4C$ different nonlinear functions to replace the original FFN module.

Table 6. Using different kernel sizes n for depthwise convolution in the spatial-wise enhancement part. The experiments are conducted using the DeiT-Ti model on the ImageNet dataset.

n	Parameters (M)	FLOPs (G)	Top-1 Acc (%)
1	4.92	1.08	72.0
3	5.00	1.10	72.6
5	5.15	1.13	72.8
7	5.37	1.17	72.9

However, in Sec. 4.1 we analyze that this method is challenging to attain a comparable degree of non-linearity since the biases should depend on the input elements. To verify this opinion, we use GeLU and AGeLU as the basic nonlinear functions, and use Eq. 11 to replace the original FFN module:

$$y = \phi_1(x) + \cdots + \phi_{4C}(x), \tag{11}$$

in which C is the number of input channel dimensions, and the basic nonlinear function $\phi(\cdot)$ can be $\mathrm{AGeLU}(\cdot)$ or $\mathrm{GeLU}(\cdot)$ function.

As the results shown in Tab. 5, none of these variants is comparable to the classification performance of the baseline with the FFN module, since according to Corollary 3.1 the biases of these nonlinear functions should be different and are dependent on all other input elements which is hard to apply in reality and is the main reason that causes the performance degradation. Experiments are conducted using the DeiT-Ti model on the ImageNet dataset.

Effect of using different kernel-size. Finally, we use different kernel size for depthwise convolution in the spatial-wise enhancement part to explore the relationship between the classification performance and the amount of spatial information used to enhance the non-linearity. In Tab. 6, we can see that as the kernel size n increases, the classification performances are getting better and better with a little increased FLOPs and parameters. The benefit is obvious from $n = 1$ to $n = 3$ since global information from the neighbors are used. The profit becomes marginal as the kernel size continues to grow.

6. Conclusion

In this paper, we analyze the effects of the FFN module in the vision transformer and show that the original FFN module is no more than a non-linearity generator whose nonlinear degree corresponds to the number of hidden dimensions. Based on this observation, we propose a flexible activation function AGeLU and combine multiple of them to form a more powerful nonlinear function that extends non-linearity through channel dimension. Furthermore, we

enhance non-linearity with spatial information using depth-wise block. With the above modification, we can use fewer hidden dimensions which reduces the FLOPs and parameters of the model without loss of classification performance. We also give a theoretical analysis of the Lipschitz bound of the proposed module by which the stability of the network can be measured. We conduct experiments on several state-of-the-art vision transformer models using the benchmark datasets including ImageNet-1k, COCO 2017 and ADE20k by replacing the original FFN module with the proposed IFFN module, and the results demonstrate the effectiveness of our method.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Chen, H., Wang, Y., Guo, J., and Tao, D. Vanillanet: the power of minimalism in deep learning. *arXiv preprint arXiv:2305.12972*, 2023.

Chen, H., Liu, Z., Wang, X., Tian, Y., and Wang, Y. Dijiang: Efficient large language models through compact kernelization. *arXiv preprint arXiv:2403.19928*, 2024.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., and Liu, W. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Guo, J., Han, K., Wu, H., Tang, Y., Chen, X., Wang, Y., and Xu, C. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12175–12185, 2022.

Guo, J., Hao, Z., Wang, C., Tang, Y., Wu, H., Hu, H., Han, K., and Xu, C. Data-efficient large vision models through sequential autoregression. *arXiv preprint arXiv:2402.04841*, 2024.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

Hao, Z., Guo, J., Han, K., Hu, H., Xu, C., and Wang, Y. Vanillakd: Revisit the power of vanilla knowledge distillation from small scale to large scale. *arXiv preprint arXiv:2305.15781*, 2023.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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

Heo, B., Yun, S., Han, D., Chun, S., Choe, J., and Oh, S. J. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11936–11945, October 2021.

Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., and Fu, B. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Liu, Z., Shen, Z., Savvides, M., and Cheng, K.-T. Reactnet: Towards precise binary neural network with generalized activation functions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV*, pp. 143–159, 2020.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, October 2021.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Neimark, D., Bar, O., Zohar, M., and Asselmann, D. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3163–3172, 2021.

Park, N. and Kim, S. How do vision transformers work? *ICLR*, 2022.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7262–7272, 2021.

Tang, Y., Liu, F., Ni, Y., Tian, Y., Bai, Z., Hu, Y.-Q., Liu, S., Jui, S., Han, K., and Wang, Y. Rethinking optimization and architecture for tiny language models. *arXiv preprint arXiv:2402.02791*, 2024.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Wu, S., Wu, T., Tan, H., and Guo, G. Pale transformer: A general vision transformer backbone with pale-shaped attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2731–2739, 2022.

Wu, X., Zeng, F., Wang, X., Wang, Y., and Chen, X. Ppt: Token pruning and pooling for efficient vision transformers. *arXiv preprint arXiv:2310.01812*, 2023.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Xu, Y., Li, C., Li, D., Sheng, X., Jiang, F., Tian, L., and Sirasao, A. Fdvit: Improve the hierarchical architecture of vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5950–5960, 2023.

Yang, C., Wang, Y., Zhang, J., Zhang, H., Wei, Z., Lin, Z., and Yuille, A. Lite vision transformer with enhanced self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11998–12008, 2022.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10819–10829, 2022.

Zheng, D., Dong, W., Hu, H., Chen, X., and Wang, Y. Less is more: Focus attention for efficient detr. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6674–6683, 2023.

A. Proof of Theorem 1

Given

$$f(x) = W^1 \text{concat}(\phi_1(W^0 x + b^0), \phi_2(W^0 x + b^0)) + b^1, \tag{12}$$

it is easy to rewrite the function as

$$\begin{aligned} f(x) &= g(x) + h(x) \\ &= \left(W^{11}\phi_1(W^0 x + b^0) + b^{11}\right) + \left(W^{12}\phi_2(W^0 x + b^0) + b^{12}\right), \end{aligned} \tag{13}$$

in which $W^1 = [W^{11} \;\; W^{12}]$ and $b^1 = [b^{11} \;\; b^{12}]$. Thus, the function $f(x)$ can be divided into two parts $g(x)$ and $h(x)$. In the following analysis, we give the proof of Lipschitz bound on $g(x)$, and the bound on $h(x)$ can be derived in the same way.

Define $x^1 = \phi_1(W^0 x + b^0) \in \mathbb{R}^n$ and $y^1 = \phi_1(W^0 y + b^0) \in \mathbb{R}^n$ for two arbitrary inputs $x, y \in \mathbb{R}^{n_0}$. Using the conclusion in Lemma 4.2, we have:

$$\begin{bmatrix} (W^0 x + b^0) - (W^0 y + b^0) \\ x^1 - y^1 \end{bmatrix}^\top \begin{bmatrix} -2p_1 q_1 T & (p_1 + q_1)T \\ (p_1 + q_1)T & -2T \end{bmatrix} \begin{bmatrix} (W^0 x + b^0) - (W^0 y + b^0) \\ x^1 - y^1 \end{bmatrix} \geq 0,$$

where $T \in \mathcal{T}_n$ (Eq. 8). The above inequality can be rewritten as:

$$\begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}^\top \begin{bmatrix} -2p_1 q_1 {W^0}^\top T W^0 & (p_1 + q_1){W^0}^\top T \\ (p_1 + q_1) T W^0 & -2T \end{bmatrix} \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix} \geq 0, \tag{14}$$

By left and right multiply $M(\rho_1, T)$ in Eq. 9 by $[(x-y)^\top \; (x^1 - y^1)^\top]$ and $[(x-y)^\top \; (x^1 - y^1)^\top]^\top$ respectively, we have:

$$\begin{aligned} &\begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}^\top \begin{bmatrix} -2p_1 q_1 {W^0}^\top T W^0 & (p_1 + q_1){W^0}^\top T \\ (p_1 + q_1) T W^0 & -2T \end{bmatrix} \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix} \\ &\qquad \leq \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}^\top \begin{bmatrix} \rho_1 I_{n_0} & 0 \\ 0 & -{W^{11}}^\top W^{11} \end{bmatrix} \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}. \end{aligned} \tag{15}$$

Combining Eq. 14 and Eq. 15, we have:

$$0 \leq \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}^\top \begin{bmatrix} \rho_1 I_{n_0} & 0 \\ 0 & -{W^{11}}^\top W^{11} \end{bmatrix} \begin{bmatrix} x - y \\ x^1 - y^1 \end{bmatrix}, \tag{16}$$

which can also be written as:

$$(x^1 - y^1)^\top {W^{11}}^\top W^{11} (x^1 - y^1) \leq \rho_1 (x - y)^T (x - y). \tag{17}$$

Recall that $g(x) = W^{11} x^1 + b^1$ and $g(y) = W^{11} y^1 + b^1$, then the inequality 17 can be written as:

$$||g(x) - g(y)||_2 \leq \sqrt{\rho_1} ||x - y||_2 \quad \text{for all } x, y \in \mathbb{R}^n. \tag{18}$$

Similarly, we have:

$$||h(x) - h(y)||_2 \leq \sqrt{\rho_2} ||x - y||_2 \quad \text{for all } x, y \in \mathbb{R}^n. \tag{19}$$

Given $f(x) = g(x) + h(x)$ in Eq. 13, we can derive:

$$\begin{aligned} ||f(x) - f(y)||_2^2 &= ||(g(x) - g(y)) + (h(x) - h(y))||_2^2 \\ &= ||g(x) - g(y)||_2^2 + ||h(x) - h(y)||_2^2 + 2\left(g(x) - g(y)\right)^\top \left(h(x) - h(y)\right) \\ &\leq ||g(x) - g(y)||_2^2 + ||h(x) - h(y)||_2^2 + 2||g(x) - g(y)||_2 ||h(x) - h(y)||_2 \\ &\leq \rho_1 ||x - y||_2^2 + \rho_2 ||x - y||_2^2 + 2\sqrt{\rho_1 \rho_2} ||x - y||_2^2 \\ &= (\sqrt{\rho_1} + \sqrt{\rho_2})^2 ||x - y||_2^2. \end{aligned} \tag{20}$$

Finally, the above inequality implies

$$||f(x) - f(y)||_2 \leq (\sqrt{\rho_1} + \sqrt{\rho_2}) ||x - y||_2 \quad \text{for all } x, y \in \mathbb{R}^n, \tag{21}$$

which gives the upper bound of $L(f) = \sqrt{\rho_1} + \sqrt{\rho_2}$ on the Lipschitz constant of $f(\cdot)$ based on the Definition 4.1.

B. Object Detection on COCO

In order to better verify the effectiveness of the proposed IFFN module, we conduct experiments for object detection on the COCO 2017 dataset, which contains 118K training images, 5K validation images and 20K test-dev images. Mask R-CNN (He et al., 2017) is considered the object detection framework and Swin-Ti is used as the baseline model. Other training settings are the same as Swin-Ti.

Table 7. Results on COCO object detection.

Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param	FLOPs
Swin-Ti	46.0	67.1	50.3	48M	267G
Swin-Ti + IFFN	46.0	67.2	50.3	44M	251G

We can see in Tab. 7 that our IFFN module can reduce over 4M parameters and 16G FLOPs compared to the original Swin-Ti model with a same box AP, which shows the priority of the proposed method.

C. Semantic Segmentation on ADE20K

We also conduct experiments for the semantic segmentation task on the ADE20K dataset, which contains 20K training images, 2K validation images and 3K test images from 150 different semantic categories. As in Swin (Liu et al., 2021), we use UperNet (Xiao et al., 2018) as the base semantic segmentation framework and Swin-Ti as the baseline model. Other training settings are the same as Swin-Ti.

Table 8. Results on ADE20K semantic segmentation.

Backbone	mIoU	mAcc	#param	FLOPs
Swin-Ti	44.5	55.6	60M	945G
Swin-Ti + IFFN	45.0	57.3	56M	928G

As shown in Tab. 8, we achieve a 0.5 mIoU improvement while reducing FLOPs by 17G and parameters by 4M compared to the baseline model Swin-Ti.