Supplementary Material for "SwinDiD: Dimension-invariant Disentangling model with Swin Transformer for Light Field Super-Resolution"

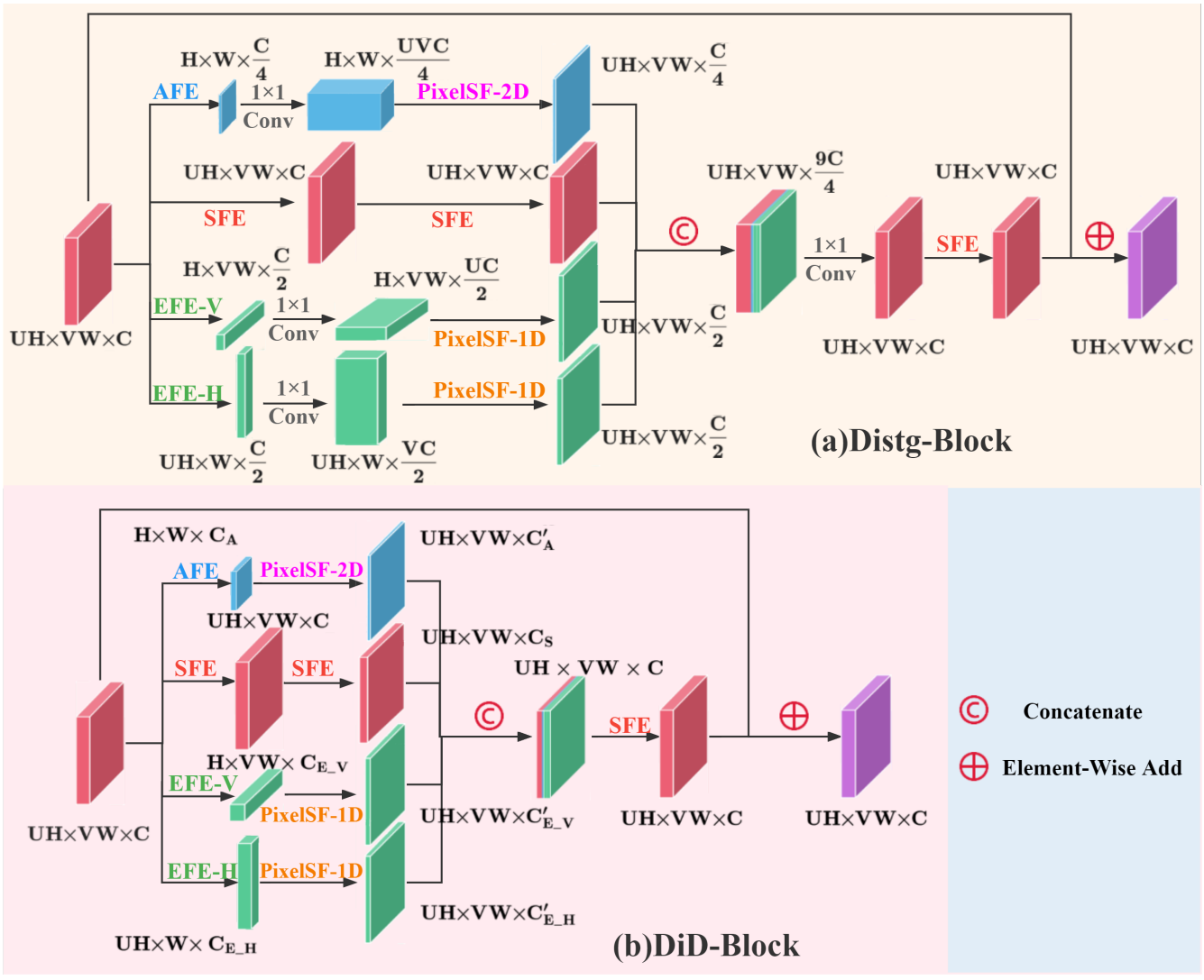

Figure 1: The convolution operation of one Distg-Block and DiD-Block.

In this supplemental material, we provide additional time complexity calculation details and three additional comparison methods have been added to the main paper .

1 Appendix A

A feature map of size $N \times N$ is fed to a convolution layer with a kernel $K \times K$ to output a feature map of size $M \times M$. The corresponding time complexity of a single convolutional layer is formulated as

$$time \sim \mathcal{O}(M^2 \times K^2 \times C_{in} \times C_{out}) \tag{1}$$

where C_{in} represents the number of input channels, that is, the number of output channels of previous layer. C_{out} represents the number of convolution kernels in this convolutional layer, that is, the number of input channels of next layer. It can be seen that the time complexity of each convolutional layer is completely determined by the area of the output feature map M^2, the area of the convolution kernel K^2, and the number of input C_{in} and output channels C_{out}. The overall time complexity of a convolutional neural network is the sum of the time complexity of each convolutional layer.

As shown in Figure 1, the input feature is of size $UH \times VW \times C$, where U and V are the angular resolution, H and W are the spatial resolutions. We define the angular resolution $U = V = A$, so the input feature of size can be express as $AH \times AW \times C$. Pixel shuffle and Concatenate operations have less time complexity, therefore it is not computed in the calculation of the time complexity. Next, we will calculate the time complexity of Distg-Block and DiD-Block respectively.

1.1 Time Complexity of Distg-Block

In the convolution operation of the AFE, the input feature F_{in} of size $AH \times AW \times C$ is fed to AFE with a kernel $A \times A$ to output F_A of size $H \times W \times C_A$. The time complexity of AFE is $\mathcal{O}(\frac{1}{4}C^2A^2HW)$ with $C_A = \frac{C}{4}$. The feature F_A is fed into a 1×1 Conv operation to output $\hat{F}_A$ of size $H \times W \times \frac{A^2C}{4}$. The time complexity of 1×1 Conv operation is $\mathcal{O}(\frac{1}{16}C^2A^2HW)$.

The time complexity of the angular feature extraction branch of Distg-Block is calculated as follows:

$$\mathcal{O}(\frac{1}{4}C^2A^2HW + \frac{1}{16}C^2A^2HW) \tag{2}$$

In the convolution operation of the SFE, the input feature F_{in} of size $AH \times AW \times C$ is fed to SFE with a kernel 3×3 to output F_S of size $AH \times AW \times C_S$. The time complexity of SFE is $\mathcal{O}(9C^2A^2HW)$ with $C_S = C$. The feature F_S is fed into another SFE with a kernel 3×3 to output F'_S of size $AH \times AW \times C'_S$. The time complexity of this SFE is $\mathcal{O}(9C^2A^2HW)$ with $C'_S = C$.

The time complexity of the spatial feature extraction branch of Distg-Block is calculated as follows:

$$\mathcal{O}(9C^2A^2HW + 9C^2A^2HW) \tag{3}$$

In the convolution operation of the EFE-H, the input feature F_{in} of size $AH \times AW \times C$ is fed to EFE-H with a kernel $1 \times A^2$ to output F_{E_H} of size $AH \times W \times C_{E\_H}$. The time complexity of EFE-H is $\mathcal{O}(\frac{1}{2}C^2A^3HW)$ with $C_{E\_H} = \frac{C}{2}$. The feature F_{E_H} is fed into a 1×1 Conv operation to output $\hat{F}_{E\_H}$ of size $AH \times W \times \frac{AC}{2}$. The time complexity of 1×1 Conv operation is $\mathcal{O}(\frac{1}{4}C^2A^2HW)$.

The time complexity of the EFE-H branch branch of Distg-Block is calculated as follows:

$$\mathcal{O}(\frac{1}{2}C^2A^3HW + \frac{1}{4}C^2A^2HW) \tag{4}$$

In the convolution operation of the EFE-V, the input feature F_{in} of size $AH \times AW \times C$ is fed to EFE-V with a kernel $A^2 \times 1$ to output F_{E_V} of size $H \times AW \times C_{E\_V}$. The time complexity of EFE-V is $\mathcal{O}(\frac{1}{2}C^2A^3HW)$ with $C_{E\_V} = \frac{C}{2}$. The feature F_{E_V} is fed into a 1×1 Conv operation to output $\hat{F}_{E\_V}$ of size $H \times AW \times \frac{AC}{2}$. The time complexity of 1×1 Conv operation is $\mathcal{O}(\frac{1}{4}C^2A^2HW)$.

The time complexity of the EFE-V branch branch of Distg-Block is calculated as follows:

$$\mathcal{O}(\frac{1}{2}C^2A^3HW + \frac{1}{4}C^2A^2HW) \tag{5}$$

After the pixel shuffle and feature concatenate, the feature becomes $F_c = AH \times AW \times \frac{9}{4}C$. The feature F_c is fed into a 1×1 Conv operation to output $\hat{F}_c$ of size $AH \times AW \times C$. The time complexity of 1×1 Conv is $\mathcal{O}(\frac{9}{4}C^2A^2HW)$. The feature $\hat{F}_c$ is fed into SFE to output F of size $AH \times AW \times C$. The time complexity of SFE is $\mathcal{O}(9C^2A^2HW)$. The time complexity of this two operation in Distg-Block is calculated as follows:

$$\mathcal{O}(\frac{9}{4}C^2A^2HW + 9C^2A^2HW) \tag{6}$$

The time complexity of Distg-Block is the sum of Eq. (2) to Eq. (6):

$$time_{Distg} = \mathcal{O}(C^2A^3HW + 30.0625C^2A^2HW) \tag{7}$$

1.2 Time Complexity of DiD-Block

In the convolution operation of the AFE, the input feature F_{in} of size $AH \times AW \times C$ is fed to AFE with a kernel $A \times A$ to output F_A of size $H \times W \times C_A$. The time complexity of AFE is $\mathcal{O}(CA^4HWY)$ with $C_A = A^2Y$.

The time complexity of the angular feature extraction branch of Distg-Block is calculated as follows:

$$\mathcal{O}(CA^4HWY) \tag{8}$$

In the convolution operation of the SFE, the input feature F_{in} of size $AH \times AW \times C$ is fed to SFE with a kernel 3×3 to output F_S of size $AH \times AW \times C_S$. The time complexity of SFE is $\mathcal{O}(9C^2A^2HW)$ with $C_S = C$. The feature F_S is fed into another SFE with a kernel 3×3 to output F'_S of size $AH \times AW \times C'_S$. The time complexity of this SFE is $\mathcal{O}(9CA^4HWY)$ with $C'_S = A^2Y$.

The time complexity of the spatial feature extraction branch of Distg-Block is calculated as follows:

$$\mathcal{O}(9C^2A^2HW + 9CA^4HWY) \tag{9}$$

In the convolution operation of the EFE-H, the input feature F_{in} of size $AH \times AW \times C$ is fed to EFE-H with a kernel $1 \times A^2$ to output F_{E_H} of size $AH \times W \times C_{E\_H}$. The time complexity of EFE-H is $\mathcal{O}(CA^5HWY)$ with $C_{E\_H} = A^2Y$.

The time complexity of the EFE-H branch branch of Distg-Block is calculated as follows:

$$\mathcal{O}(CA^5HWY) \tag{10}$$

In the convolution operation of the EFE-V, the input feature F_{in} of size $AH \times AW \times C$ is fed to EFE-V with a kernel $A^2 \times 1$ to output F_{E_V} of size $H \times AW \times C_{E\_V}$. The time complexity of EFE-V is $\mathcal{O}(CA^5HWY)$ with $C_{E\_V} = A^2Y$.

The time complexity of the EFE-V branch branch of Distg-Block is calculated as follows:

$$\mathcal{O}(CA^5HWY) \tag{11}$$

After the pixel shuffle and feature concatenate, the feature becomes $F_c = AH \times AW \times C$. The feature F_c is fed into SFE with a kernel 3×3 to output F of size $AH \times AW \times C$. The time complexity of SFE is as follows:

$$\mathcal{O}(9C^2A^2HW) \tag{12}$$

The time complexity of DiD-Block is the sum of Eq. (8) to Eq. (12):

$$time_{DiD} = \mathcal{O}((18C^2A^2HW + 10CA^4HWY + 2CA^5HWY) \tag{13}$$

from the formulas $C = (A+1)^2Y$ in the main paper, Eq. (13) can becomes:

$$time_{DiD} = \mathcal{O}((22 + 2A + \frac{2A^2 - 10A - 4}{(A+1)^2})C^2A^2HW) \tag{14}$$

Therefore, according to Eq. (7) and Eq. (14), the time complexity of Distg-Block and DiD-Block are $\mathcal{O}((30 + A)C^2A^2HW)$ and $\mathcal{O}((22 + 2A)C^2A^2HW)$, respectively.

Table 1: PSNR/SSIM values achieved by different methods for ×2 and ×4 SR. The best results are bolded.

Method	scale	EPFL	HCInew	HCIold	INRIA	STFgantry
Bicubic	×2	29.74/0.938	31.89/0.936	37.69/0.979	31.33/0.958	31.06/0.950
VDSR [4]	×2	32.50/0.974	34.37/0.956	40.61/0.987	34.44/0.974	35.54/0.979
EDSR [1]	×2	33.09/0.963	34.83/0.959	41.01/0.987	34.98/0.976	36.30/0.982
RCAN [2]	×2	33.16/0.963	35.02/0.960	41.13/0.987	35.04/0.977	36.67/0.983
resLF [3]	×2	33.62/0.971	36.69/0.974	43.42/0.993	35.36/0.980	38.35/0.990
LFSSR [5]	×2	33.67/0.974	36.80/0.975	43.81/0.994	35.28/0.983	37.94/0.990
LF-ATO [6]	×2	34.27/0.976	37.24/0.977	44.21/0.994	36.17/0.984	39.64/0.993
LF-InterNet [7]	×2	34.11/0.976	37.17/0.976	44.57/**0.995**	35.83/0.984	38.43/0.991
DistgSSR [8]	×2	34.37/0.977	37.73/0.979	44.80/**0.995**	36.12/0.985	40.08/0.994
DiD	×2	34.64/**0.978**	37.79/0.979	**44.93/0.995**	36.53/0.985	40.21/0.994
SwinDiD	×2	**34.71/0.978**	**37.97/0.980**	44.92/**0.995**	**36.62/0.986**	**40.70/0.995**
Bicubic	×4	25.26/0.832	27.71/0.852	32.58/0.934	26.95/0.887	26.09/0.854
VDSR [4]	×4	27.25/0.878	29.31/0.882	34.81/0.952	29.19/0.920	28.51/0.901
EDSR [1]	×4	27.83/0.885	29.59/0.887	35.18/0.954	29.65/0.926	28.70/0.907
RCAN [2]	×4	27.90/0.886	29.69/0.889	35.36/0.955	29.80/0.928	29.02/0.913
resLF [3]	×4	28.26/0.904	30.72/0.911	36.71/0.968	30.34/0.941	30.19/0.937
LFSSR [5]	×4	28.59/0.912	30.93/0.914	36.90/0.970	30.58/0.947	30.57/0.943
LF-ATO [6]	×4	28.51/0.912	30.88/0.913	37.00/0.970	30.71/0.948	30.61/0.943
LF-InterNet [7]	×4	28.51/0.912	30.82/0.914	36.83/0.970	30.50/0.947	30.22/0.938
DistgSSR [8]	×4	28.77/0.915	31.16/0.918	37.25/0.972	30.82/0.949	31.03/0.949
DiD	×4	**28.86**/0.917	31.29/0.920	37.48/**0.973**	**31.04/0.951**	31.26/0.951
SwinDiD	×4	**28.86/0.918**	**31.30/0.921**	**37.52/0.973**	30.93/**0.951**	**31.37/0.952**

2 Appendix B

On the basis of the main paper, we have added two SISR methods which are EDSR [1], and RCAN [2] and a LF image SR method which is resLF [3] for comparison, and compare the whole image with each other.

1)Quantitative Results: Table 1 shows the quantitative results achieved by SwinDiD in comparison with other state-of-the-art SR methods. SwinDiD achieves competitive PSNR and SSIM results on all five datasets of ×2 SR.

2)Qualitative Results: Figure 2 and Figure 3 show the whole image visual quality comparisons of different methods for 2× SR and 4× SR, respectively. It can be seen that the SISR methods including VDSR [4], EDSR [1], and RCAN [2], fail to recover complex textures and details. In contrast, the deep learning based LF image SR methods can produce better visual effects than SISR methods, which is attribute to the use of different viewpoints information. However, edges and textures recovered by these methods are still suffer from blurring. Compared with state-of-the-art methods, our SwinDiD can recover complex structures with shaper edges and fine details. Besides, we can observe that SwinDiD generates more appealing result than DiD in 2× and 4× SR.

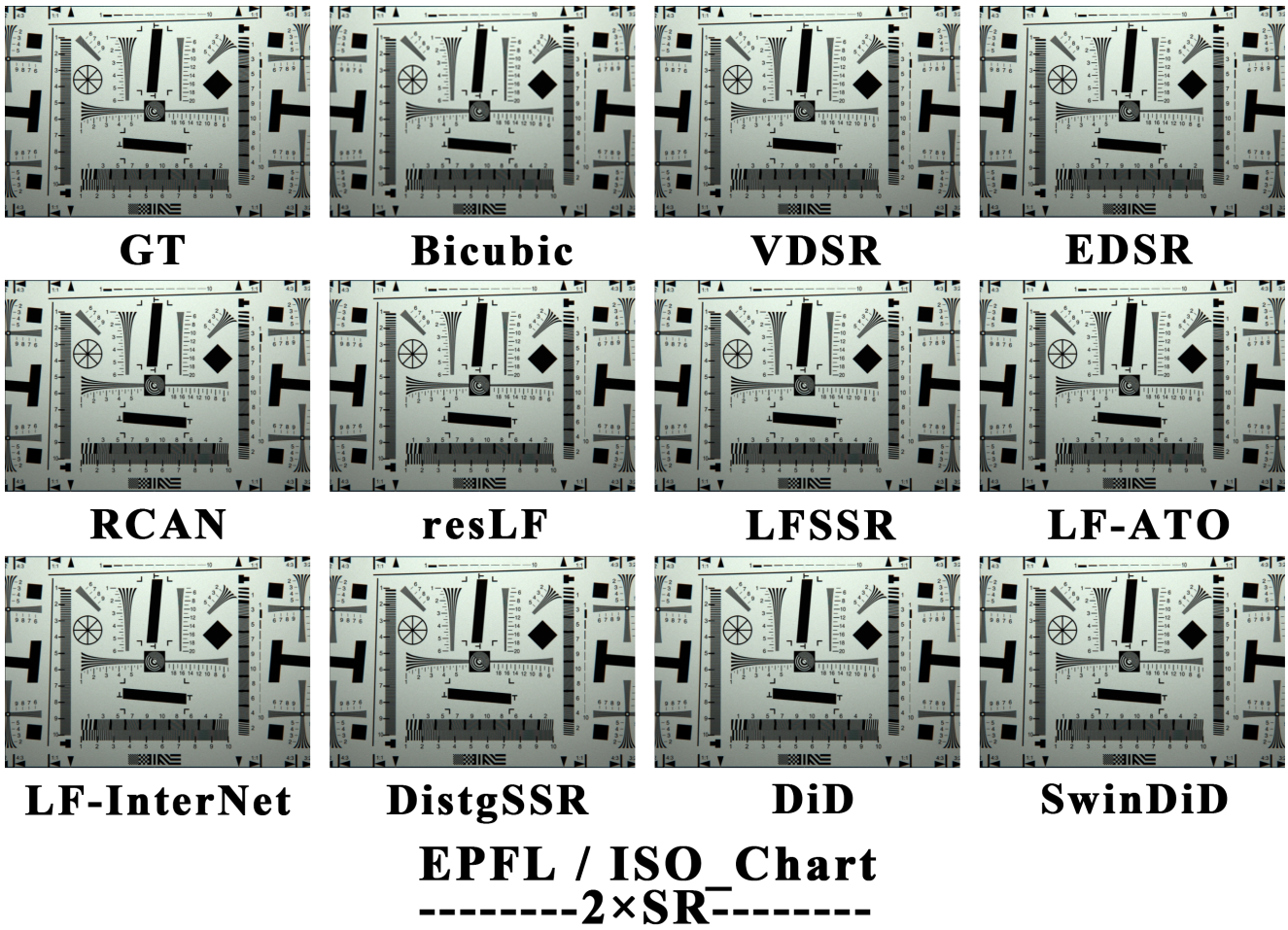

Figure 2: Visual quality comparisons of 2× SR for different methods.

Figure 3: Visual quality comparisons of 4× SR for different methods.

References

[1] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1132–1140.

[2] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," 2018.

[3] S. Zhang, Y. Lin, and H. Sheng, "Residual networks for light field image super-resolution," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11038–11047, 2019.

[4] J. K. Lee J. Kim and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654.

[5] H. W. F. Yeung, X. Chen J. Hou, J. Chen, Z. Chen, and Y. Y. Chung, "Light field spatial super-resolution using deep efficient spatial-angular separable convolution," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2319–2330, 2019.

[6] J. Jin, J. Hou, J. Chen, and S. Kwong, "Light field spatial super-resolution via deep combinatorial geometry embedding and structural consistency regularization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2257–2266.

[7] Y. Wang, L. Wang, J. Yang, W. An, J. Yu, and Y. Guo, "Spatial-angular interaction for light field image super-resolution," *ArXiv*, vol. abs/1912.07849, 2020.

[8] Y. Wang, L. Wang, G. Wu, J. Yang, W. An, J. Yu, and Y. Guo, "Disentangling light fields for super-resolution and disparity estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.