OPE-SR: Orthogonal Position Encoding for Designing a Parameter-free Upsampling Module in Arbitrary-scale Image Super-Resolution

Supplementary Material

A. OPE Theory

Our proposed Orthogonal Position Encoding (OPE) is constructed using four basic functions

$$\{\cos(ux)\cos(vy), \cos(ux)\sin(vy), \sin(ux)\cos(vy), \sin(ux)\sin(vy)\}$$

where u and v are frequencies greater than or equal to zero. These functions represent the real form of the 2D-Fourier basis for a **periodic** and **continuous** 2D signal. Representing real 2D signals with complex 2D-Fourier basis, we use the conjugate symmetry to derive the real form basis. Our derivation extends the real form basis of 1D-Fourier transform [7, Chap. 3.3.1] to the 2D case.

Suppose $f_{XY}(x, y)$ represents a **periodic and continuous** complex signal in 2D, where X and Y is the minimum positive period for x- and y-axis, respectively. In our paper, we set $X = Y = 2$. Applying the Fourier and inverse Fourier transforms, we have:

$$\begin{aligned} F(u,v) &= \frac{1}{XY}\int_{-X/2}^{+X/2}\int_{-Y/2}^{+Y/2} f_{XY}(x,y)e^{-iu(2\pi/X)x}e^{-iv(2\pi/Y)y}\mathrm{d}x\mathrm{d}y \\ &= \frac{1}{XY}\int_{-X/2}^{+X/2}\int_{-Y/2}^{+Y/2} f_{XY}(x,y)e^{-2\pi i(ux/X+vy/Y)}\mathrm{d}x\mathrm{d}y \end{aligned} \tag{1}$$

and

$$f_{XY}(x,y) = \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} F(u,v)e^{2\pi i(ux/X+vy/Y)} \tag{2}$$

where $u, v \in \mathbb{Z}$. When $f_{XY}(x, y) \in \mathbb{R}$ is a real signal, we have $\overline{f_{XY}(x,y)} = f_{XY}(x,y)$ and the following equation

$$f_{XY}(x,y) = \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} \overline{F(u,v)}e^{-2\pi i(ux/X+vy/Y)} \tag{3}$$

Replacing u, v in Eq.(2) by $-u$ and $-v$ yields

$$f_{XY}(x,y) = \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} F(-u,-v)e^{-2\pi i(ux/X+vy/Y)} \tag{4}$$

Comparing Eq.(3) and Eq.(4), we have the conjugate symmetry

$$F(-u,-v) = \overline{F(u,v)} \tag{5}$$

Adding Eq.(2) and Eq.(3), we obtain

$$\begin{aligned} f_{XY}(x,y) &= \frac{1}{2}\left[\sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} F(u,v)e^{2\pi i(ux/X+vy/Y)} + \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} \overline{F(u,v)}e^{-2\pi i(ux/X+vy/Y)}\right] \\ &= \frac{1}{2}\sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty}\left[F(u,v)e^{2\pi i(ux/X+vy/Y)} + \overline{F(u,v)}e^{-2\pi i(ux/X+vy/Y)}\right] \\ &= \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} \mathrm{Re}\left[F(u,v)e^{2\pi i(ux/X+vy/Y)}\right]. \end{aligned} \tag{6}$$

Let $F(u,v) \triangleq A_{u,v} + iB_{u,v}$, where $A_{u,v}, B_{u,v} \in \mathbb{R}$ are real numbers. Expanding Eq.(6) yields

$$\begin{aligned} f_{XY}(x,y) &= \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty} \mathrm{Re}\left[(A_{u,v}+iB_{u,v})\left(\cos\left(2\pi\left(\frac{ux}{X}+\frac{vy}{Y}\right)\right) + i\sin\left(2\pi\left(\frac{ux}{X}+\frac{vy}{Y}\right)\right)\right)\right] \\ &= \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty}\left[A_{u,v}\cos\left(2\pi\left(\frac{ux}{X}+\frac{vy}{Y}\right)\right) - B_{u,v}\sin\left(2\pi\left(\frac{ux}{X}+\frac{vy}{Y}\right)\right)\right] \\ &= \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty}\left[A_{u,v}\cos\left(\frac{2\pi}{X}ux\right)\cos\left(\frac{2\pi}{Y}vy\right) - A_{u,v}\sin\left(\frac{2\pi}{X}ux\right)\sin\left(\frac{2\pi}{Y}vy\right)\right. \\ &\quad \left. -B_{u,v}\sin\left(\frac{2\pi}{X}ux\right)\cos\left(\frac{2\pi}{Y}vy\right) - B_{u,v}\cos\left(\frac{2\pi}{X}ux\right)\sin\left(\frac{2\pi}{Y}vy\right)\right] \end{aligned} \tag{7}$$

Therefore, we obtain the real basis for $u, v \in \mathbb{Z}$.
For brevity we set u, v to be natural numbers $\in \mathbb{N}$ from now on and define

$$\begin{cases} e_1 \triangleq \cos\left(\frac{2\pi}{X}ux\right)\cos\left(\frac{2\pi}{Y}vy\right) \\ e_2 \triangleq \sin\left(\frac{2\pi}{X}ux\right)\sin\left(\frac{2\pi}{Y}vy\right) \\ e_3 \triangleq \sin\left(\frac{2\pi}{X}ux\right)\cos\left(\frac{2\pi}{Y}vy\right) \\ e_4 \triangleq \cos\left(\frac{2\pi}{X}ux\right)\sin\left(\frac{2\pi}{Y}vy\right) \end{cases} \tag{8}$$

Based on the conjugate symmetry in Eq.(5), we have

$$A_{-u,-v} + iB_{-u,-v} = A_{u,v} - iB_{u,v}, \tag{9}$$

which implies

$$\begin{aligned} A_{-u,-v} &= A_{u,v} \\ B_{-u,-v} &= -B_{u,v} \end{aligned} \tag{10}$$

Combining Eq.(7), Eq.(8) and Eq.(10), we can rewrite $f_{XY}(x,y)$ as

$$\begin{aligned} f_{XY}(x,y) &= \sum_{u=-\infty}^{+\infty}\sum_{v=-\infty}^{+\infty}\left[A_{u,v}e_1 - A_{u,v}e_2 - B_{u,v}e_3 - B_{u,v}e_4\right] \\ &= \sum_{u=1}^{+\infty}\sum_{v=1}^{+\infty} 2\left(A_{u,v}e_1 - A_{u,v}e_2 - B_{u,v}e_3 - B_{u,v}e_4\right) + 2\left(A_{u,-v}e_1 + A_{u,-v}e_2 - B_{u,-v}e_3 + B_{u,-v}e_4\right) \\ &\quad + \sum_{u=1}^{+\infty} 2\left(A_{u,0}e_1 - B_{u,0}e_3\right) + \sum_{v=1}^{+\infty} 2\left(A_{0,v}e_1 - B_{0,v}e_4\right) + A_{0,0} \\ &= \sum_{u=1}^{+\infty}\sum_{v=1}^{+\infty}\left[2\left(A_{u,v}+A_{u,-v}\right)e_1 - 2\left(A_{u,v}-A_{u,-v}\right)e_2 - 2\left(B_{u,v}+B_{u,-v}\right)e_3 - 2\left(B_{u,v}-B_{u,-v}\right)e_4\right] \\ &\quad + \sum_{u=1}^{+\infty} 2\left(A_{u,0}e_1 - B_{u,0}e_3\right) + \sum_{v=1}^{+\infty} 2\left(A_{0,v}e_1 - B_{0,v}e_4\right) + A_{0,0} \end{aligned} \tag{11}$$

Since $A_{0,0}$, $A_{u,0}$, $A_{0,v}$, $A_{u,v}$, $A_{u,-v}$, $B_{0,0}$, $B_{u,0}$, $B_{0,v}$, $B_{u,v}$, and $B_{u,-v}$ are real numbers and are independent to each other when $u, v \geq 0$ in Eq.(11), we can express $f_{XY}(x, y)$ as a linear combination of e_1, e_2, e_3, and e_4.

Notice when $uv = 0$, some basis in Eq.(8) would degenerate into one variable function or constant term, where Eq.(11) has already distinguished them. For a simpler way to construct e_1, e_2, e_3, e_4, we could also multiply each item of every variable's 1D-Fourier real form basis.

B. Additional Results

B.1. DIV2K

For DIV2K validation set [1], we achieve competitive visual results against SOTA [5]. Notice for larger scale factor ($\times 24$ - $\times 30$), there are ringing effects at the edges of the images (Fig. 1, Fig. 2, Fig. 4) in LIIF [3] and our method, while LTE's [5] are less obvious. This indicates even MLP is hard to restore high-frequency information [3] while using position encoding to enhance it could mitigate this [5]. However, as an uninterpretable method, LTE still has some stains at the edges and cannot obtain smooth edges (see $\times 30$ scale factor). For our method, we can easily locate the problem at the selection of max frequency of OPE. To achieve better results, larger max frequency of OPE and larger training scale factor is necessary, this would be a promising future work.

B.2. Benchmark

For Benchmark datasets [2, 4, 6, 8], we persist competitive results against SOTA [5]. There are negligible difference for most of the results (Fig. 5, Fig. 6, Fig. 7). For those images with complex texture, like Urban100 [4], we choose 3 representational images. For regular texture (Fig. 8), our method keeps neat edges, while LIIF [3] and LTE [5] suffers from severe distortions (see $\times 8$) and rough edge. For elongated texture (Fig. 9), we still keep minimum deformation and do not lost it (see $\times 8$) , while LTE, as a enhanced MLP with complex continuous image representation, is unstable to capture this slight detail (see $\times 8$). For those more complex texture with perspective relation and varying element size (Fig. 10), existing arbitrary-scale SR methods do not obtain favorable results, this would be a challenging future work.

C. Flipping Consistency Verification

This section we will provide more samples for verifying flipping consistency. When the feature map is sent to the upsampling module, we perform four transformations. flip-0: do not flip; flip-1: flip horizontal; flip-2: flip vertical; flip-3: flip horizontal and vertical. As shown in Fig. 11, Fig. 12, Fig. 13, for LIIF [3], the SR results of flip-1, flip-2 and flip-3 become blurred compared to flip-0, this is due to the INR-based upsampling module is lack of symmetry inductive bias while it needs extra data augmentation to learn the symmetry in training. Notice the original SR results of flip-1, flip-2 and flip-3 are flipped following feature map's pattern, we flip them back to align flip-0's result for the convenience of comparison. As for our method, we observe all the results are totally the same, this also can be explained that 2D Fourier bases are naturally symmetrical.

D. More Training Details

For constructing LR-HR pairs in training, currently arbitrary-scale SR methods take similar procedure. A scale factor r is uniformly sampled from $[1, 4]$, then cropping HR patch ($48 \cdot r \times 48 \cdot r$) from the training set and then generating fixed-size (48×48) LR patch by bicubic. Then 2304 pixels are sampled randomly from HR and mapped into 2D domain ($[-1, 1] \times [-1, 1]$) by position coords. Finally, the coords and LR are inputs while the HR pixels are ground truth. Notice different r will cause different coordinate density to capture multi-scale information, hence to keep r be sampled uniformly is essential.

In Sec. 3.3 and Sec. 4.1, we designed an interesting "inverse operation" experiment and discussed the selection of max frequency of OPE. However, we believe the experiment results' potential are more than that. We find the delicate relationship of max frequency and scale factor r, it may implicitly affect the effectiveness of training. For example, sampling $r = 4$ when constructing LR-HR pairs may be best for OPE frequency $n = 3$ as shown in Sec. 4.1, while smaller r could introduce redundant information into the result, or in other words, it is redundancy for representing such a low target resolution with such a high OPE frequency. However, keep scale factor r fixed to 4 is unreasonable since the coordinate density will be fixed and could not capture multi-scale information. To balance the problems, we fix $r = 4$ and interpolate cropped HR patch with bicubic to higher resolution before sampling pixels.

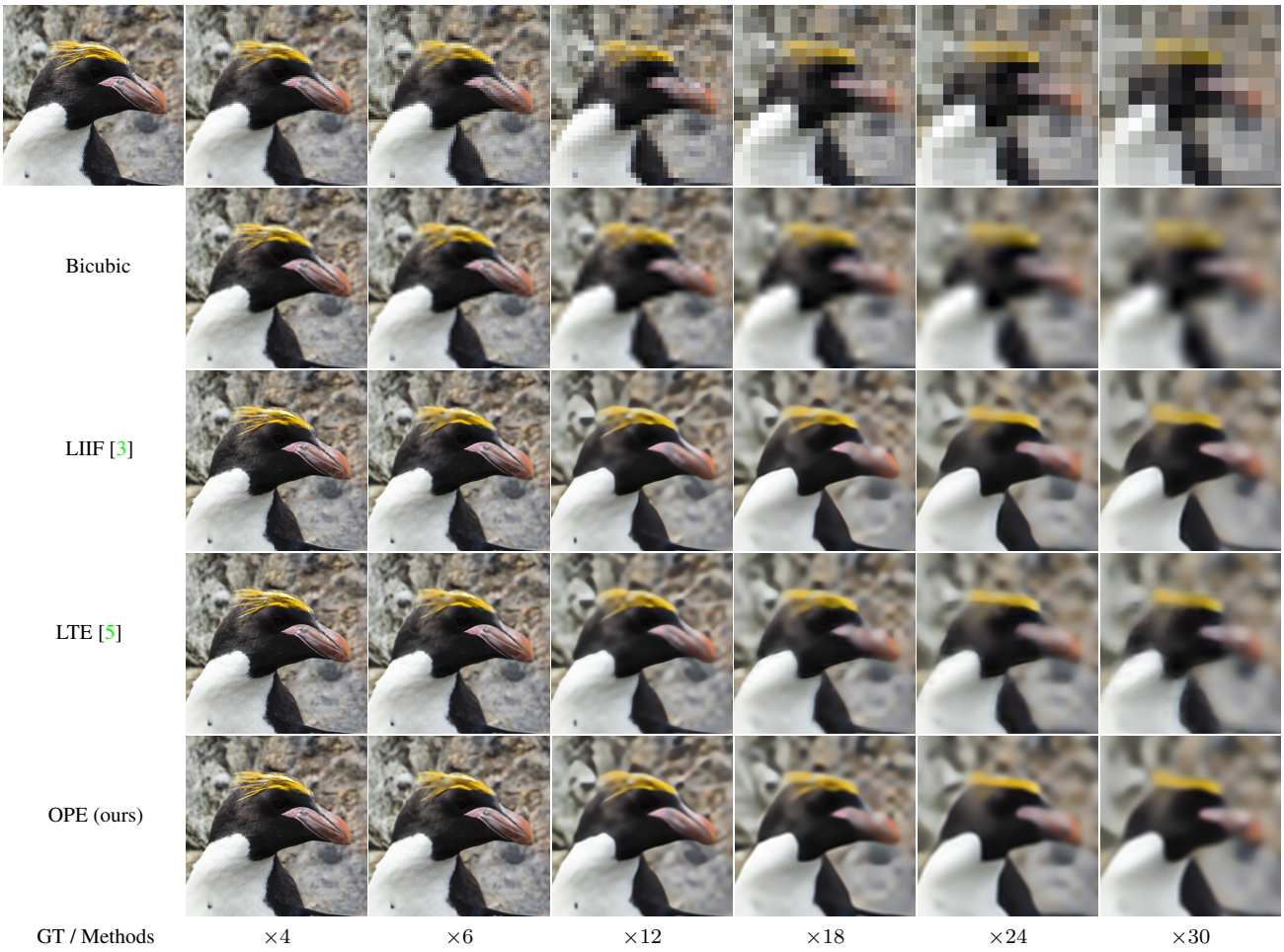

Figure 1. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On div2k 0801.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 3

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 3

[3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

[4] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 3

[5] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1929–1938, 2022. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[6] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 3

[7] Alan V. Oppenheim and Alan S. Willsky. *Signals and Systems*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1997. 1

[8] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 3

Figure 2. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On div2k 0866.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

Figure 3. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On div2k 0882.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

Figure 4. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On div2k 0896.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

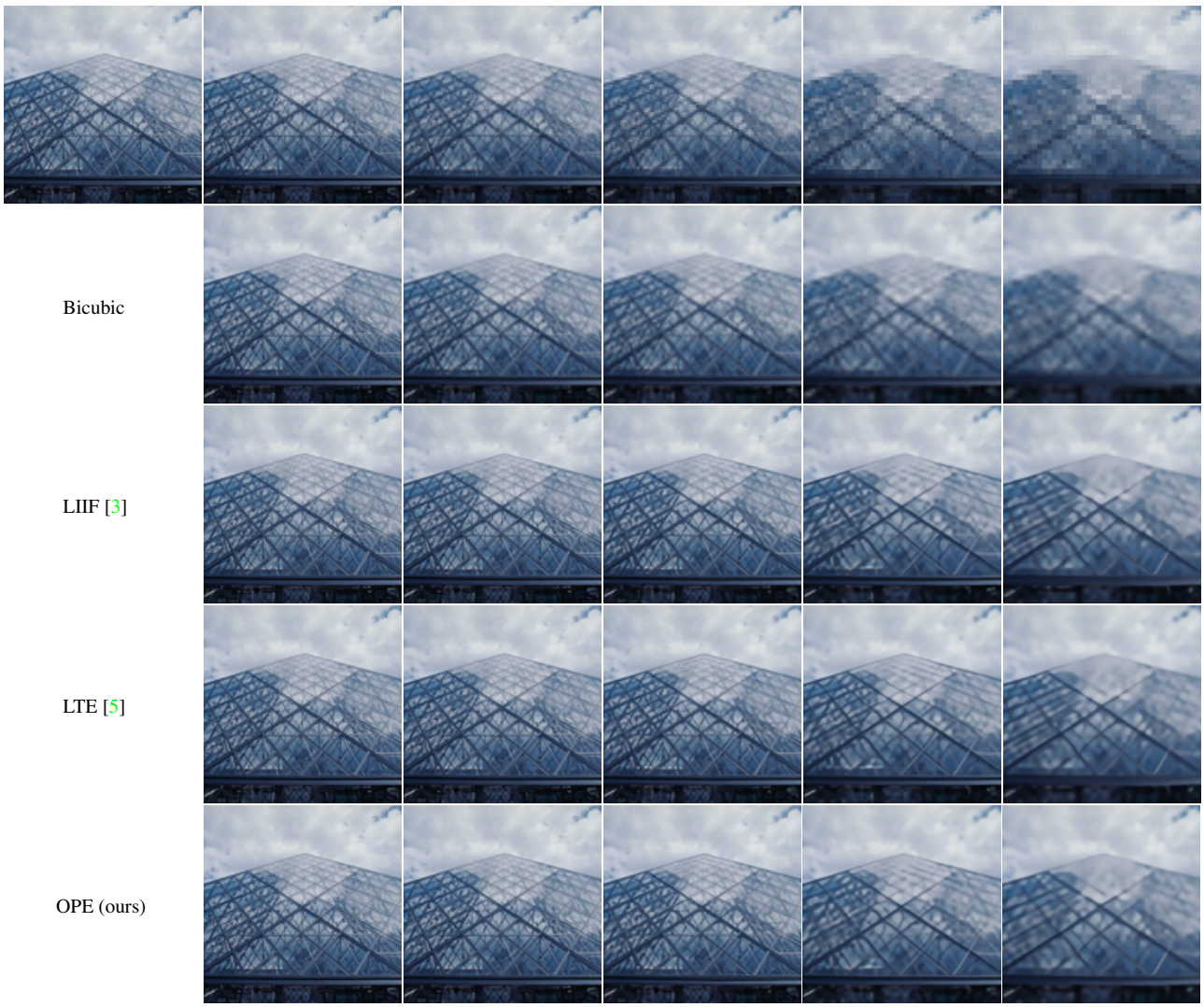

Figure 5. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On b100 223061.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

Figure 6. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On set5 butterfly.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

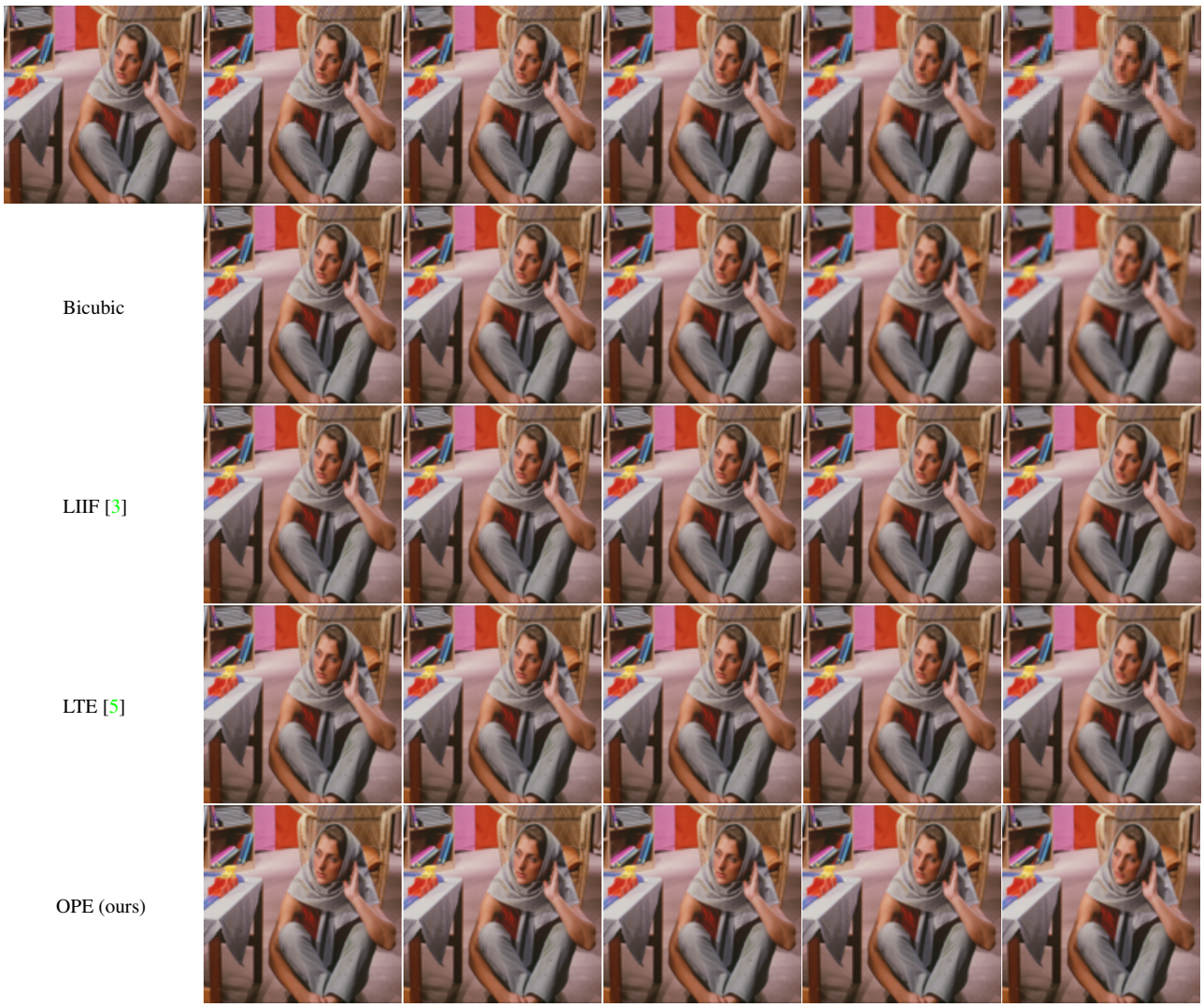

Figure 7. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On set14 barbara.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

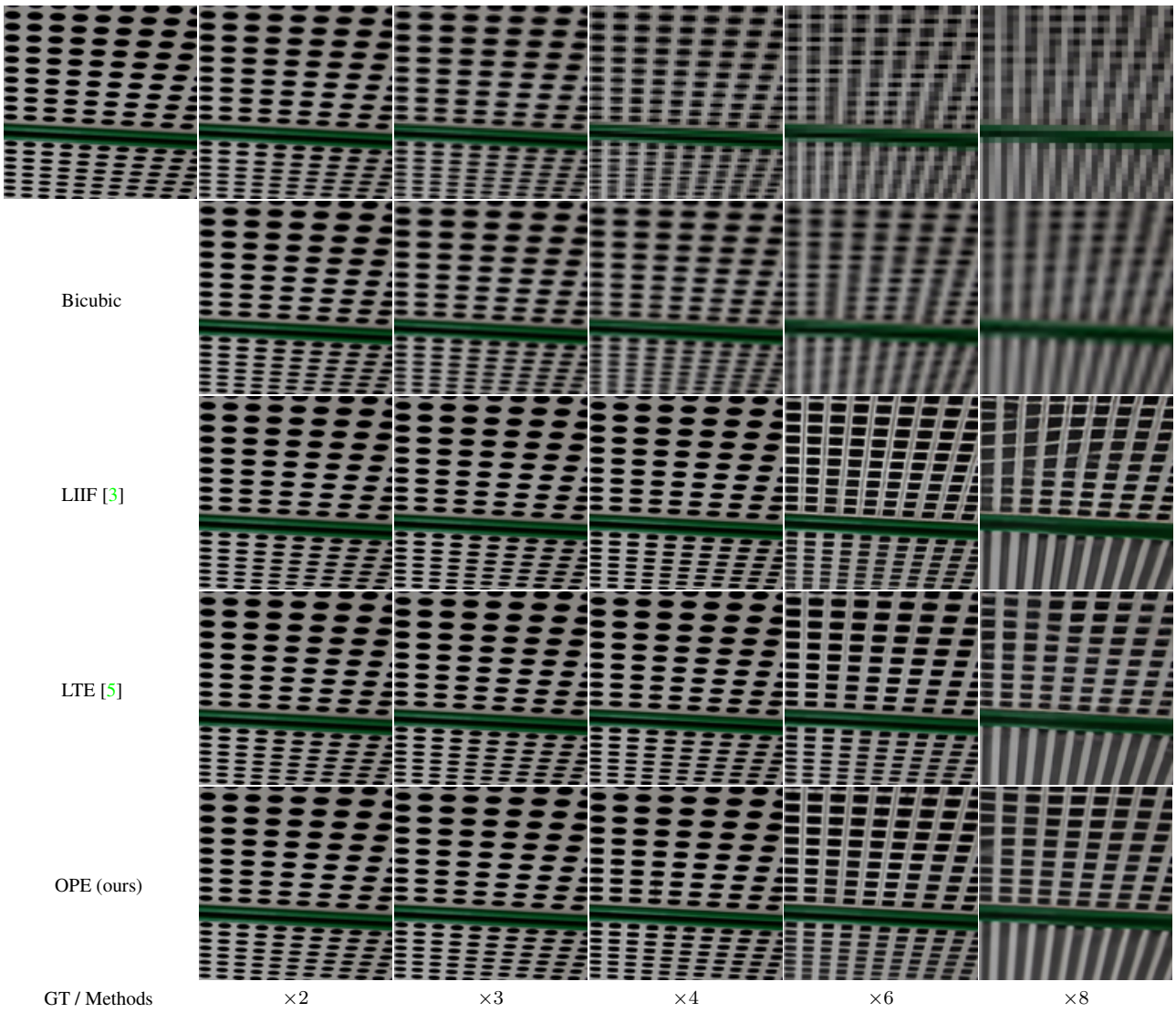

Figure 8. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On urban100 img004.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

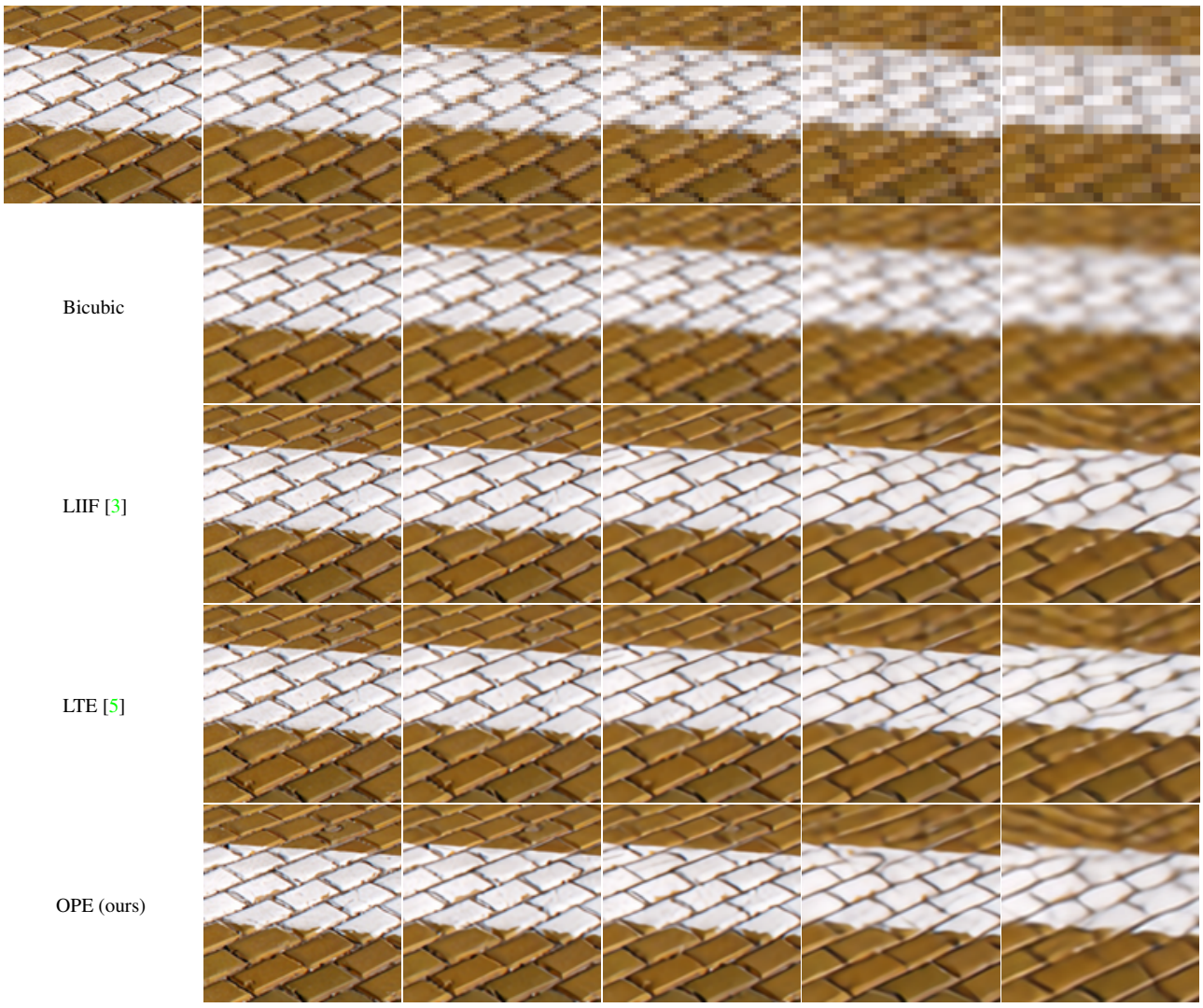

Figure 9. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On urban100 img091.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

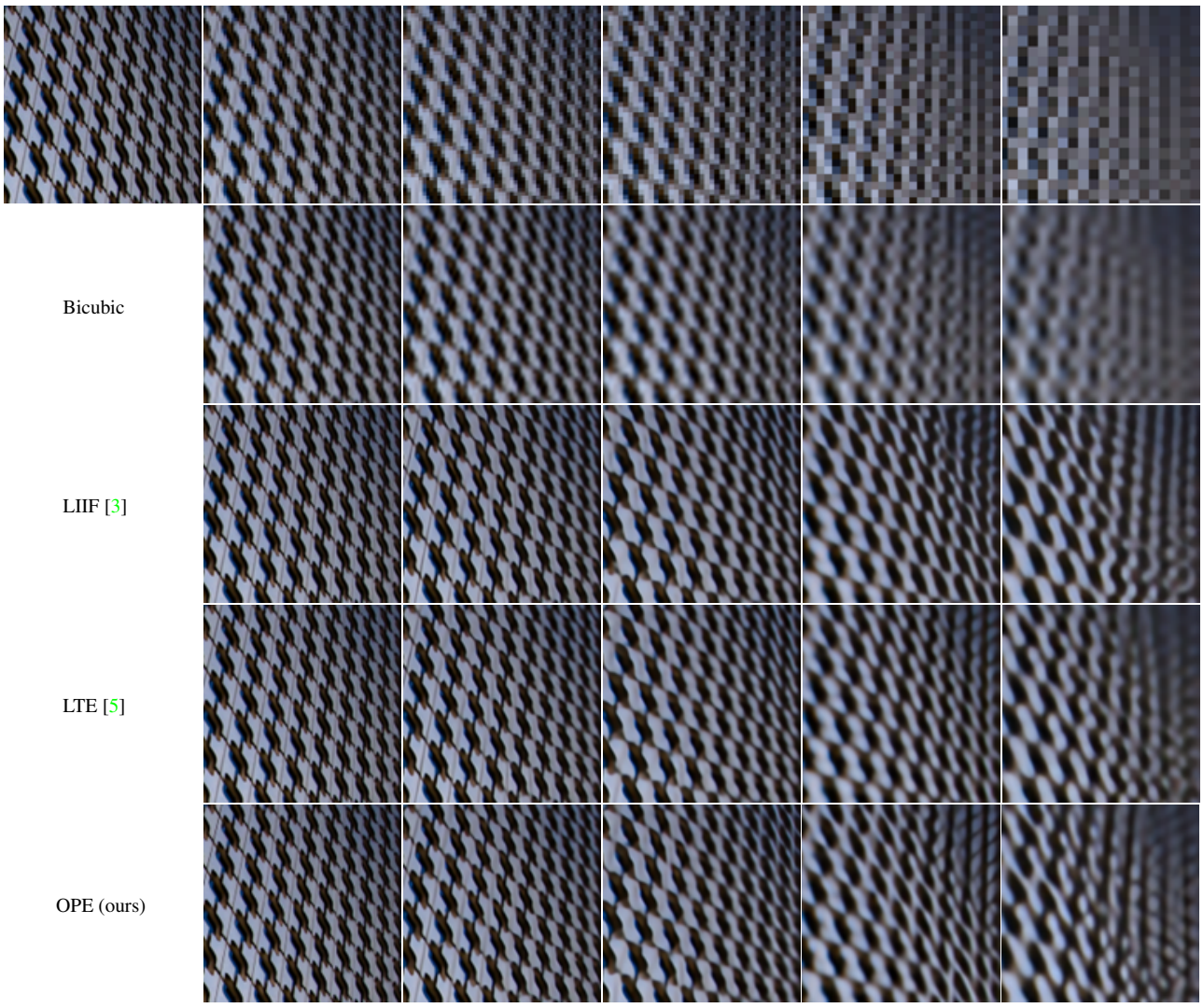

Figure 10. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. On urban100 img041.png. 1-st row: ground truth, input images with different scale factor (larger scale factor to lower resolution).

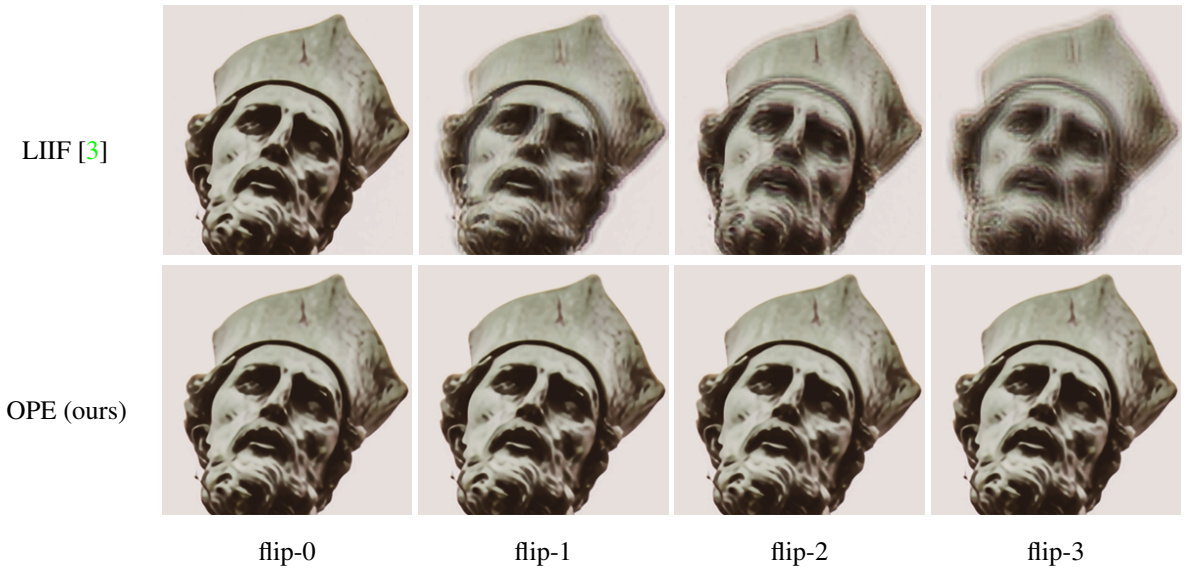

Figure 11. **Flipping consistency** on div2k 0868.png. flip-0: do not flip; flip-1: flip horizontal; flip-2: flip vertical; flip-3: flip horizontal and vertical.

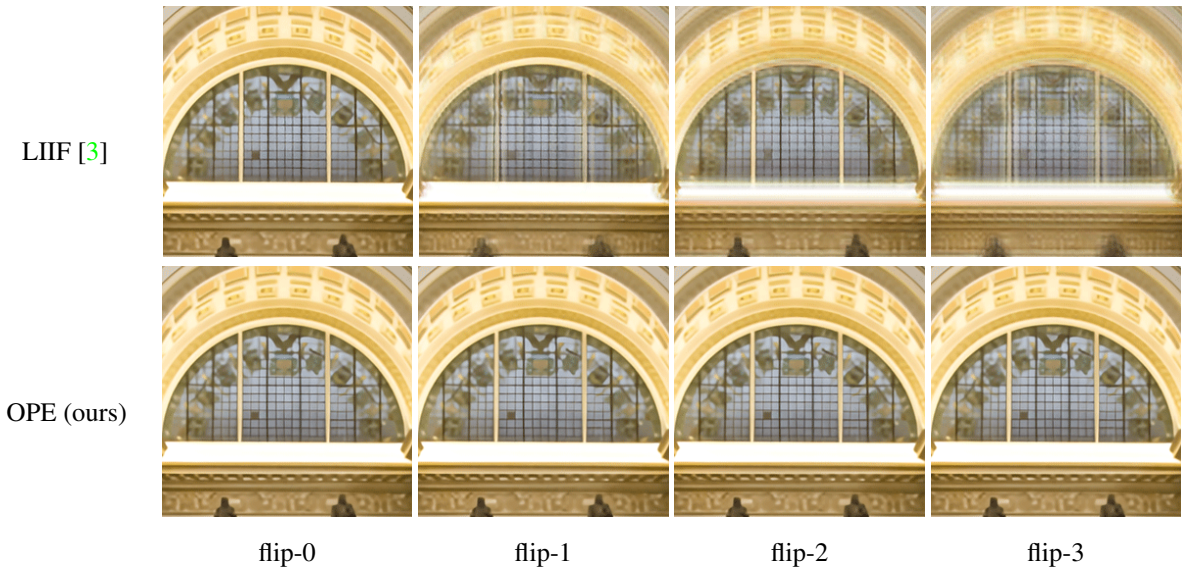

Figure 12. **Flipping consistency** on div2k 0884.png. flip-0: do not flip; flip-1: flip horizontal; flip-2: flip vertical; flip-3: flip horizontal and vertical.

Figure 13. **Flipping consistency** on div2k 0891.png. flip-0: do not flip; flip-1: flip horizontal; flip-2: flip vertical; flip-3: flip horizontal and vertical.