

# Best of Both Worlds: Learning Arbitrary-scale Blind Super-Resolution via Dual Degradation Representations and Cycle-Consistency

## *Supplementary Materials*

Shao-Yu Weng<sup>1</sup> Hsuan Yuan<sup>1</sup> Yu-Syuan Xu<sup>2</sup> Ching-Chun Huang<sup>1</sup> Wei-Chen Chiu<sup>1</sup>  
<sup>1</sup> National Yang Ming Chiao Tung University      <sup>2</sup> MediaTek Inc.

Here, we provide the detailed structure of our implicit degradation predictor, explicit kernel estimator, adaptive arbitrary-scale SR module, as well as the training details of these aforementioned modules/sub-networks. Then, more quantitative and qualitative experimental results are provided. Finally with an ablation study shows the effectiveness of wavelet transform. Our source code and model are available here.

## A. Details of Networks

### A.1. Implicit Degradation Predictor

The implicit degradation predictor  $E$  is composed of six stacked and repeated blocks, in which each block is sequentially composed of convolution, batch normalization, and leaky Relu layer. The kernel size of each convolution layer is  $3 \times 3$ , and the channel size increases from 9 (which is the concatenation of the high-frequency subbands) to 256. While for the predictor head, which is the same as [2], is composed of only one aforementioned block and followed by a linear layer.

### A.2. Explicit Kernel Estimator

The explicit kernel estimator is made up of two fully connected layers with channel size from 256 (which is the dimension of the implicit degradation representation) to 64, and the four separately fully connected layers with output sizes of 121, 49, 25, and 1. Due to having small numerical values in the kernels, we also use a weighted mask, which applies on  $\hat{k}_l$  and  $k_l$ , to aid the convergence of the model. The initial mask values are set to 100 but are doubled at locations  $(i, j)$  where the corresponding values in  $k_l$  are greater than zero, as given by the following equation:

$$mask^{(i,j)} = \begin{cases} 200, & \text{if } k_l^{(i,j)} > 0 \\ 100, & \text{otherwise} \end{cases} \quad (1)$$

### A.3. Adaptive Arbitrary-scale SR Module

The image feature extractor is EDSR [6] without the up-sampling layer, which is composed of 16 Resblocks. And

the INR network consists of three consecutive compositions, each of which consists of an MLP layer with a relu layer behind it. Here we further provide a detailed process of how we derived the input coordinates and cell size when querying the SR with HR and LR size: Following LIIF [3] and LTE [5], the coordinates are first normalized to  $[-1, 1]$ , and the cell size  $c$  is defined as  $\frac{2}{shape}$ , where  $shape$  indicates the size of the queried image (for simplicity, in the following explanation, we assume the image has the same height and width). Hence, the coordinates and cell size are what we control to get the SR with HR or LR size. For the SR in LR size which is in the shape of  $n \times n$ , the general formula to get the input coordinate is defined as

$$-1 + c \times \left(\frac{1}{2} + m\right) \quad (2)$$

where  $m$  belongs to the integer between  $[0, n]$ , and the cell size  $c$  here is therefore  $\frac{2}{n}$ . For the SR image with HR size, which is  $s$  times larger than LR, with shape  $sn \times sn$ , the cell size  $c$  becomes  $\frac{2}{sn}$ . And the coordinate is queried with the Equation 2 with  $m$  in the range of  $[0, sn]$ .

### A.4. Training Details

In stage one (i.e., the degradation representation training stage), we empirically set the batch size to 128 and the model is trained for 1500 epochs with learning rate 0.03. In stage two (i.e. arbitrary-scale super-resolution training stage), 1000 epochs with learning rate 0.001 and batch size 16 are set for training. Both are optimized with Adam of having  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . Since the training data is online generated, it limits the diversity of the generated data (because the size of input images have to be the same in a batch) and impedes the arbitrary-scale blind-SR training scheme. Motivated by [8], we maintain a data queue to put images inside after data processing at each iteration. And the index is shuffled to select the data for training. The queue size is empirically set to be 2048 for stage one and 320 for stage two.

## B. More Quantitative Results

Here, we provide more results in different scales in Table 1, and more quantitative experiment on fixed kernel as well as RealSR dataset.

### B.1. Fixed Kernel Comparison

To experiment with degradation under various degradation conditions, we compare our method with other ASSR models. We use six different degradations (with two different blur kernels and three different downsampling operations, which are labeled as 'kernel' and 'down' respectively in Table 2) for evaluation. All methods here are trained under unknown degradations. Table 2 presents the performance on Set14 [11] under  $\times 4$  up-sampling scale with three evaluation metrics. Quantitative results demonstrate that our proposed method outperforms other ASSR methods in all kinds of degradations.

### B.2. RealSR

To justify the robustness under real-world images, the quantitative results on the RealSR version-3 dataset [1] are provided. All methods are trained the same as mentioned in Section 4.1 except for the noise level being set to 10, followed by the way that [10] adopts for the real-world images. In Table 3, our method is on average superior to the others on all scales; especially on scale two, we are better than others by more than 1db on PSNR.

## C. More Qualitative Results

More qualitative results are provided here: we generate the real-world SR images from the RealSR version-3 dataset [1] (cf. Figure 1) and the historical images (cf. Figure 2), while the continuous scale SR images are from Set14 [11] (cf. Figure 5). We observe that the SR images generated by the baselines produce more artifacts than ours. Also, they suffer from distortion (for example, the second row in Figure 1 demonstrates the distortion of the hole, where they should be in the shape of a circle), while our method better preserves the structure.

## D. Effectiveness of Wavelet transform

We demonstrate the benefit of learning degradation representation from the high-frequency subbands of input LR images in the wavelet space (noting again that our implicit degradation predictor takes the high-frequency subbands as input). Firstly, we adopt t-SNE [7] to visualize the distributions of implicit degradation representations learnt in wavelet space or the original RGB space. As there are four parameters (i.e.  $\{\lambda_1, \lambda_2, \theta\}$  to determine the degradation kernel and the downsampling scale  $s$ ) which would make differences upon the degradation representations, three different experimental scenarios are considered here: 1) fixing

the scale  $s$  while randomizing the other parameters; 2) fixing the kernel parameters  $\{\lambda_1, \lambda_2, \theta\}$  and randomizing the scale  $s$ ; and 3) randomizing all four parameters. The randomization upon scale parameter  $s$  is  $\sim \mathcal{U}(1, 10)$ ,  $\{\lambda_1, \lambda_2\}$  is  $\sim \mathcal{U}(0.2, 6)$ , and  $\theta$  is  $\sim \mathcal{U}(0, \pi)$ . Please note that, in order to have better visualization, for each of the experimental scenarios, there are only 6 distinct combinations of  $\{\lambda_1, \lambda_2, \theta, s\}$ . In Figure 3 the first row shows the distributions for the implicit degradation representations (produced by the implicit degradation predictor) learnt in the wavelet space, while the second row shows the ones learnt in the RGB space. Moreover, three columns sequentially correspond to the aforementioned three experimental scenarios. The implicit representation stemmed from different parameter settings are colorized differently. It is observed that, with adopting wavelet transform, learnt implicit representations are more discriminative (or distinguishable) with respect to be different degradations and scaling factors, in comparison to the ones learnt in the RGB space.

Secondly, we would like to show the (indirect) impact of using wavelet transform upon the learning of explicit kernel estimation. In Figure 4, it demonstrates the kernel  $\widehat{k}_l$  in LR space estimated by our method. All of the training procedures and hyper-parameters are set to be the same. It shows that with using the original RGB space instead of the wavelet one for learning the implicit representations, all explicit kernels degenerate to a cross-like shape with few variations in magnitudes. We argue that, since the explicit kernel estimator is implicit-degradation-dependent, the worse discriminativeness in the learnt implicit representations thereby negatively affects the explicit kernel estimator. In contrast, our proposed method can estimate the explicit kernel with various tendencies and magnitudes in consistent with the corresponding groundtruth kernel  $k_l$ .

In Table 4, it reports the quantitative performance with respect to whether the implicit degradation representation is learnt upon the original RGB or wavelet spaces. It can be concluded that the wavelet transform does help to improve the learning of super-resolution. We can also see the importance of learning discriminative degradation representations towards the task of arbitrary-scale blind-SR.

## References

- [1] Jianrui Cai, Shuhang Gu, Radu Timofte, and Lei Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.
- [2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Table 1. Quantitative comparison on various datasets with using continuous upsampling scales. The best performance is in red while the second best is in blue. All models are trained with continuous scales randomly sampled from  $\mathcal{U}(1, 4)$ .

Dataset		Set5	Set14	BSD100	Urban100
Scale	Method	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
×2	Bicubic	27.9485 / 0.7667	25.7769 / 0.6757	25.5221 / 0.6294	22.9406 / 0.6234
	MetaSR [4]	31.0373 / 0.8715	27.9307 / 0.8042	<b>26.4721</b> / 0.7228	24.7597 / 0.7401
	LIIF [3]	<b>31.8991</b> / <b>0.8892</b>	28.1747 / 0.8137	26.4302 / 0.7272	24.9905 / 0.7519
	LTE [5]	31.7331 / 0.8865	28.2148 / 0.8144	26.4117 / 0.7251	<b>25.0141</b> / 0.7515
	SRNO [9]	31.5307 / 0.8851	<b>28.6815</b> / <b>0.8303</b>	26.3458 / <b>0.7344</b>	<b>25.0909</b> / <b>0.7622</b>
	Ours	<b>32.7075</b> / <b>0.8975</b>	<b>28.5204</b> / <b>0.8244</b>	<b>26.4582</b> / <b>0.7335</b>	<b>25.2237</b> / <b>0.7652</b>
×2.7	Bicubic	24.7017 / 0.6857	23.9357 / 0.6188	24.2446 / 0.5925	21.3123 / 0.5611
	MetaSR [4]	25.7880 / 0.7370	24.5656 / 0.6491	24.7521 / 0.6251	22.1059 / 0.6076
	LIIF [3]	<b>26.0169</b> / <b>0.7500</b>	<b>24.6822</b> / <b>0.6564</b>	<b>24.8227</b> / <b>0.6337</b>	<b>22.2511</b> / <b>0.6207</b>
	LTE [5]	25.7915 / 0.7398	24.5778 / 0.6510	24.7266 / 0.6270	22.1302 / 0.6127
	SRNO [9]	25.9713 / 0.7470	24.4804 / 0.6500	24.6761 / 0.6301	22.1040 / 0.6113
	Ours	<b>26.5096</b> / <b>0.7720</b>	<b>25.3506</b> / <b>0.6837</b>	<b>25.0788</b> / <b>0.6555</b>	<b>22.6549</b> / <b>0.6504</b>

Table 2. Quantitative comparison on Set14 with ×4 upsampling scale. The best performance is colored in red. All models are trained under unknown degradations. Please refer to the description in Section B.1 for details.



Method		MetaSR	LIIF	LTE	SRNO	Ours
kernel	down	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
	bicubic	25.7660 / 0.6930	26.0368 / 0.7038	26.0692 / 0.7026	26.0654 / 0.7113	<b>26.3796</b> / <b>0.7168</b>
	bilinear	25.4994 / 0.6801	25.8211 / 0.6938	25.7557 / 0.6904	25.9430 / 0.7005	<b>26.2525</b> / <b>0.7110</b>
	area	25.8433 / 0.6932	26.2315 / 0.7069	26.7001 / 0.7216	26.4831 / 0.7186	<b>26.7462</b> / <b>0.7239</b>
	bicubic	25.9936 / 0.7011	26.2683 / 0.7133	26.3721 / 0.7154	26.2579 / 0.7162	<b>26.5603</b> / <b>0.7247</b>
	bilinear	25.7851 / 0.6903	26.1042 / 0.7054	26.1754 / 0.7051	26.1238 / 0.7067	<b>26.4504</b> / <b>0.7188</b>
	area	26.1388 / 0.7029	26.5863 / 0.7188	26.1719 / 0.7036	26.6942 / 0.7240	<b>26.9884</b> / <b>0.7322</b>

Table 3. Quantitative results on RealSR version-3 dataset

Scale	×2	×3	×4
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
MetaSR	29.664 / 0.869	27.973 / 0.822	26.881 / 0.786
LIIF	29.752 / <b>0.874</b>	28.014 / <b>0.826</b>	<b>27.004</b> / <b>0.793</b>
LTE	29.777 / 0.871	27.751 / 0.819	26.696 / 0.788
SRNO	<b>30.110</b> / 0.870	<b>28.068</b> / 0.815	26.841 / 0.784
Ours	<b>31.644</b> / <b>0.890</b>	<b>28.773</b> / <b>0.823</b>	<b>27.393</b> / <b>0.789</b>

Table 4. Ablation study upon the utilization of wavelet transform in learning implicit degradation representations. The experiments are based on validation set of DIV2K.

scale	w/o wavelet	w/ wavelet
	PSNR / SSIM	PSNR / SSIM
×2	30.094 / 0.821	30.538 / 0.837
×3	29.276 / 0.797	29.810 / 0.811
×4	28.478 / 0.766	28.796 / 0.777

[4] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *IEEE Conference on Com-*

puter Vision and Pattern Recognition (CVPR), 2019.

[5] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[7] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 2008.

[8] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[9] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] Mehmet Yamac, Baran Ataman, and Aakif Nawaz. Kernel-net: A blind super-resolution kernel estimation network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, 2012.

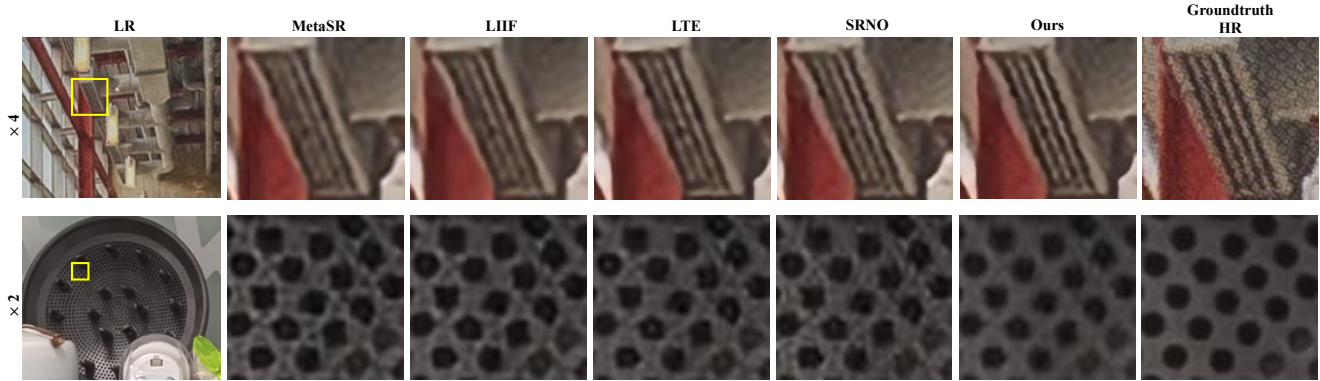


Figure 1. Qualitative results on RealSR version-3 dataset with  $\times 2$  and  $\times 4$  upsampling scales. The baseline methods produce more artifacts in the super-resolution results.

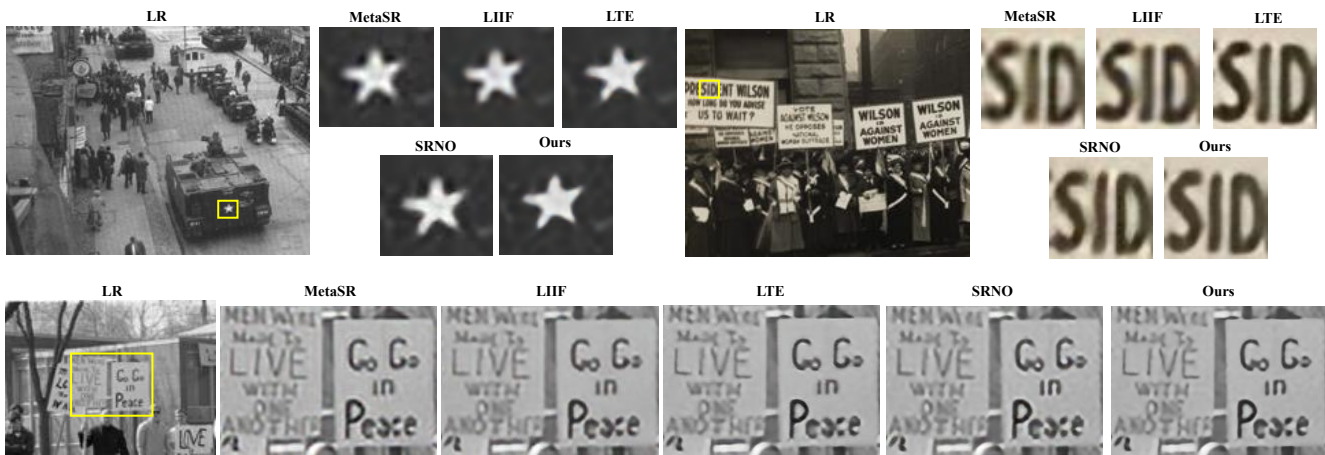


Figure 2. Qualitative results on historical images with  $\times 4$  upsampling scale.

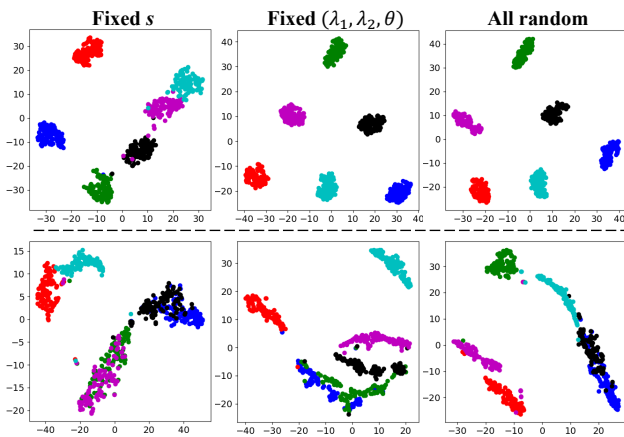


Figure 3. The top row shows degradation representation training with wavelet transform. The bottom row demonstrates the results without wavelet transform. There are three schemes provided: the downsampling scale is fixed at  $\text{scale}=4$  and hyper-parameters are randomly sampled (label as Fixed  $s$ ), the blur kernel  $(\lambda_1, \lambda_2, \theta)$  is fixed at  $(1.1, 2.5, 65)$  (label as Fixed  $(\lambda_1, \lambda_2, \theta)$ ). The rest of column set all of hyper-parameters randomly (label as All random).

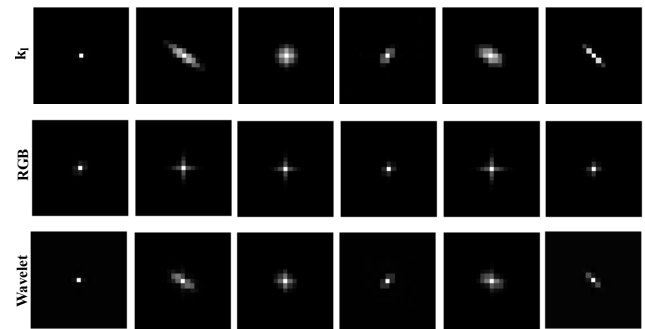


Figure 4. The top row shows the ground truth  $k_l$ . The middle and the bottom rows show the estimation results of explicit kernels, respectively with adopting the RGB or wavelet spaces for learning implicit degradation representations.



Figure 5. Qualitative results on Set14 with various continuous upsampling scales. We can observe that our method restores clear edges for all scale factors, particularly in the letter "w".