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

Gaochao Song[1] Qian Sun[2*] Luo Zhang[3] Ran Su[1] Jianfeng Shi[2] Ying He[3]
[1]Tianjin University [2]Nanjing University of Information Science and Technology
[3]Nanyang Technological University
gaochaosong_21@tju.edu.cn sunqian@nuist.edu.cn

Abstract

Arbitrary-scale image super-resolution (SR) is often tackled using the implicit neural representation (INR) approach, which relies on a position encoding scheme to improve its representation ability. In this paper, we introduce orthogonal position encoding (OPE), an extension of position encoding, and an OPE-Upscale module to replace the INR-based upsampling module for arbitrary-scale image super-resolution. Our OPE-Upscale module takes 2D coordinates and latent code as inputs, just like INR, but does not require any training parameters. This parameter-free feature allows the OPE-Upscale module to directly perform linear combination operations, resulting in continuous image reconstruction and achieving arbitrary-scale image reconstruction. As a concise SR framework, our method is computationally efficient and consumes less memory than state-of-the-art methods, as confirmed by extensive experiments and evaluations. In addition, our method achieves comparable results with state-of-the-art methods in arbitrary-scale image super-resolution. Lastly, we show that OPE corresponds to a set of orthogonal basis, validating our design principle. [1]

1. Introduction

Photographs are composed of discrete pixels of varying precision due to the limitations of sampling frequency, which breaks the continuous visual world into discrete parts. The single image super-resolution (SISR) task aims to restore the original continuous world in the image as much as possible. In an arbitrary-scale SR task, one often reconstructs the continuous representation of a low-resolution image and then adjusts the resolution of the target image as needed. The recent rise of implicit neural representation (INR) in 3D vision has enabled the representation of complex 3D objects and scenes in a continuous manner [14,19,41,42,44,45,47,49,57,58], which also opens up possibilities for continuous image and arbitrary-scale image super-resolution [5,18,32,72].

Existing methods for arbitrary-scale SR typically use a post-upsampling framework [70]. In this approach, low-resolution (LR) images first pass through a deep CNN network (encoder) without improving the resolution, and then pass through an INR-based upsampling module (decoder) with a specified target resolution to reconstruct high-resolution (HR) images. The decoder establishes a mapping from feature maps (the output of encoder) to target image pixels using a pre-assigned grid partitioning and achieves arbitrary-scale with the density of the grid in Cartesian coordinate system. However, the INR approach has a defect of learning low-frequency information, also known as spectral bias [50]. To address this issue, sinusoidal positional encoding is introduced to embed input coordinates to higher dimensions and enable the network to learn high-frequency details. This inspired recent works on arbitrary-scale SR to further improve the representation ability [32,72].

Despite its effectiveness in arbitrary-scale SR, the INR-based upsampling module increases the complexity of the entire SR framework as two different networks are jointly trained. Additionally, as a black-box model, it represents a continuous image with a strong dependency on both the feature map and the decoder (e.g., MLP). However, its representation ability decreases after flipping the feature map, a phenomenon known as flipping consistency decline. As shown in Fig. 1, flipping the feature map horizontally before the upsampling module of LIIF results in a blurred target image that does not have the expected flip transformation. This decline could be due to limitations of the MLP in learning the symmetry feature of the image.

MLP is a universal function approximator [17], which tries to fit a mapping function from feature map to the continuous image, therefore, it is reasonable to assume that such process could be solved by an analytical solution. In this paper, we re-examine position encoding from the per-

*Corresponding author.
[1]Project page: https://github.com/gaochao-s/ope-sr

spective of orthogonal basis and propose orthogonal position encoding (OPE) for continuous image representation. The linear combination of 1D latent code and OPE can directly reconstruct continuous image patch without using implicit neural function [5]. To prove OPE's rationality, we analyse it both from functional analysis and 2D-Fourier transform. We further embed it into a parameter-free upsampling module, called OPE-Upscale Module, to replace INR-based upsampling module in deep SR framework, then currently deep SR framework can be greatly simplified.

Unlike the state-of-the-art method by Lee et al. [32], which enhances MLP with position encoding, we explore the possibility of extending position encoding without MLP. By providing a more concise SR framework, our method achieves high computing efficiency and consumes less memory than the state-of-the-art, while also achieving comparable image performance in arbitrary-scale SR tasks.

Our contributions are as follows:

- We propose a novel position encoding, called orthogonal position encoding (OPE), which takes the form of a 2D-Fourier series and corresponds to a set of orthogonal basis. Building on OPE, we introduce the OPE-Upscale Module, a parameter-free upsampling module for arbitrary-scale image super-resolution.
- Our method significantly reduces the consumption of computing resources, resulting in high computing efficiency for arbitrary-scale SR tasks.
- The OPE-Upscale Module is interpretable, parameter-free and does not require training, resulting in a concise SR framework that elegantly solves the flipping consistency problem.
- Extensive experiments demonstrate that our method achieves comparable results with the state-of-the-art. Furthermore, our method enables super-resolution up to a large scale of $\times 30$.

2. Related Work

2.1. Sinusoidal Positional Encoding

Sinusoidal positional encoding is widely used to counteract the negative effects of token order and sequence length in sequence models [65], or to guide image generation as the spatial inductive bias in CNNs [7, 23, 36]. In implicit neural representations, it plays a critical role in solving spectral bias [50]. By embedding input coordinates into a higher dimensional space, position encoding greatly improves high-frequency representation of implicit 3D scenes [42, 56] and subsequent works take it as the default operation to improve representation quality [37, 43, 53, 78]. Inspired by these works, positional encoding has been preliminarily explored in representing images in arbitrary-scale SR [32, 72].

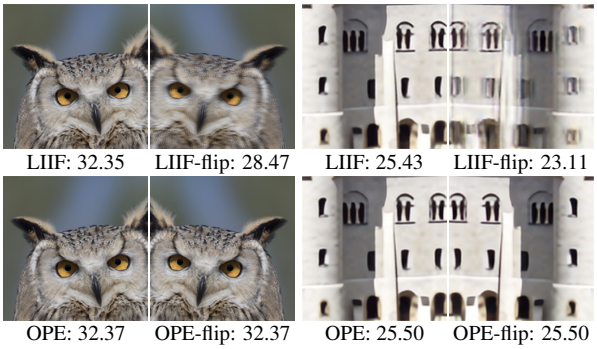


Figure 1. **Flipping consistency decline (PSNR (dB)).** LIIF-flip: Flipping the input of the LIIF [5] decoder yields blurred results in the symmetric outputs. OPE-flip: Our method does not exhibit such artifacts. Additional results can be found in the supplementary material.

2.2. Orthogonal Basis Decomposition

In functional analysis, an orthogonal basis decomposition is a way to represent a vector or function as a linear combination of orthogonal basis functions. Wavelet transform [2, 15, 20, 38] and 2D-Fourier transform [6, 12, 13, 22, 74] are commonly used decomposition techniques for images and videos. In DSGAN [12], the input image is explicitly decomposed into low and high frequencies using high-pass and low-pass filters. Other methods use frequency domain losses to decompose images, either in a supervised manner [22] or in an unsupervised manner [13]. To address with resolution discrepancy of reconstructed images and input images, Rippel et al. [52] employ spectral pooling to decrease resolution by truncating in the Fourier domain, while Zhou et al. [81] explore an up-sampling method in the Fourier domain to increase the resolution. Image moments, which decompose images into two-dimensional orthogonal polynomials [27, 82], are widely used in invariant pattern recognition [30, 77]. Image sparse representation inherits this decomposition idea and performs well in traditional computer vision tasks [39, 71, 73]. In 3D domains, spherical harmonics serve as an orthogonal basis in space to represent view dependence [3, 51, 59] and have recently been proposed as a replacement for MLPs [11] for representing neural radiance fields [42].

2.3. Deep Learning-based SR

Based on the upsampling operations and their location in the model, deep learning-based SR frameworks can be classified into four categories (see [70] for a comprehensive survey): pre-upsampling [8, 24, 25, 55, 61, 62], post-upsampling [9, 31, 35, 64, 79], progressive-upsampling [28, 29, 68], and iterative up-and-down sampling [16, 33, 69]. With pre-upsampling, the LR image is first upsampled by traditional interpolation and then fed into a deep CNN to reconstruct

high-quality details. While it was one of the most popular frameworks for arbitrary-scale factors, it has side effects like enlarged noise by interpolation and high time complexity and space consumption. The progressive-upsampling and iterative up-and-down sampling frameworks pose challenges in terms of complicated model designing and unclear design criteria, as noted in [70]. For post-upsampling, the LR image is directly fed as input to a deep CNN, and then a trainable upsampling module (e.g., deconvolution [9], sub-pixel [54], and interpolation convolution [10]) increases the resolution at the end. Since feature extraction process, which is computationally intensive, only occurs in low-dimensional space, it has become one of the mainstream frameworks [32, 34, 67].

2.4. Arbitrary-scale SR

In the field of arbitrary-scale SR, most existing works are based on the post-upsampling framework and replace the traditional upsampling module with an INR-based one, such as a coordinate-based MLP. Meta-SR [18] was the first arbitrary-scale SR method based on CNN. ArbSR [66] adopts a general plug-in module to solve the scaling problem of different horizontal and vertical scales. SRWarp [60] transforms LR images into HR images with arbitrary shapes via a differential adaptive warping layer. SphereSR [75] explores arbitrary-scale on 360° images. LIIF [5] uses coordinates and conditional latent code into an MLP to directly predict target pixel color with an intuitive network structure. LIIF-related follow-up works focus on predicting high-frequency information with position encoding [32, 72].

3. Method

3.1. OPE-based Image Representation

Given an LR image I_{LR} with resolution of $h \times w$, we devide its 2D domain into $h \times w$ grids, where each grid represents a pixel in the LR image and corresponds to a patch of the high resolution image of size $r_h \times r_w$. The output image I_{SR} has a resolution of $H \times W$, where $H = r_h \cdot h$ and $W = r_w \cdot w$. Denote by $I \in \mathbb{R}^{r_h \times r_w \times 1}$ the high resolution image patch with size $r_h \times r_w$ for a specific color channel. We view each pixel in I as a sample of a continuous bivariate function $f(x, y) : [-1, 1] \times [-1, 1] \to \mathbb{R}$. We embed the coordinates x and y using sinusoidal positional encoding

$$\begin{aligned} X &= \gamma(x) = \sqrt{2} \cdot [\frac{1}{\sqrt{2}}, \cos(\pi x), \sin(\pi x), \cos(2\pi x), \\ &\quad \sin(2\pi x), ..., \cos(n\pi x), \sin(n\pi x)] \end{aligned} \tag{1}$$

$$\begin{aligned} Y &= \gamma(y) = \sqrt{2} \cdot [\frac{1}{\sqrt{2}}, \cos(\pi y), \sin(\pi y), \cos(2\pi y), \\ &\quad \sin(2\pi y), ..., \cos(n\pi y), \sin(n\pi y)], \end{aligned} \tag{2}$$

where $\gamma(\cdot) : \mathbb{R} \to \mathbb{R}^{1\times(2n+1)}$ is a univariate function for position encoding with a predefined maximum frequency $n \in \mathbb{N}$. We flatten the matrix $X^T Y$ into a row vector $P \in \mathbb{R}^{1\times(2n+1)^2}$ as

$$P = \text{flat}(X^T Y), \tag{3}$$

where flat : $\mathbb{R}^{(2n+1)\times(2n+1)} \to \mathbb{R}^{1\times(2n+1)^2}$ is the flattening operation. Denote by $e_{i,j}$ the element on the i-th row and the j-th column of matrix $X^T Y$. For example. $e_{4,5} = 2\cos(2\pi x)\sin(2\pi y)$. It is easy to verify that

$$\langle e_{i_1,j_1}, e_{i_2,j_2} \rangle = \begin{cases} 0, & (i_1, j_1) \neq (i_2, j_2) \\ 1, & (i_1, j_1) = (i_2, j_2) \end{cases} \tag{4}$$

where $\langle , \rangle$ is the inner product in function space, i.e.,

$$\langle g, h \rangle = \frac{1}{4} \int_{-1}^{1} \int_{-1}^{1} g(x, y) h(x, y) \mathrm{d}x \mathrm{d}y.$$

Therefore the elements $\{e_{i,j}\}$ form a set of orthogonal basis, allowing us to approximate f as a linear combination. Consider a pixel (x, y) in the upsampled image patch I, which corresponds to a region $[x - 1/r_h, x + 1/r_h] \times [y - 1/r_w, y + 1/r_w]$. We use $f(x, y)$ as a representative for the entire region and compute $I_{(x,y)}$ as

$$I_{(x,y)} \triangleq f(x, y) \approx ZP^T, \tag{5}$$

where $Z \in \mathbb{R}^{1\times(2n+1)^2}$ represents the projection. Due to the orthogonal property, we call the resulting vector P as orthogonal position encoding (OPE). Fig. 3 illustrates the concept of OPE-based patch representation.

Remark 1. Our OPE basis can be seen as the real form version of the 2D-Fourier basis, which eliminates the complex exponential term based on conjugate symmetry when representing real signals. See the supplementary material for details.

Remark 2. OPE differs from the commonly used positional encoding formulation, such as that in [42], in that it includes a constant term and takes the product of each coordinate embedding as a new term. These seemingly minor changes indeed have a deep impact. Conventional positional encoding processes the coordinates separately, thereby encoding frequencies in the horizontal and vertical directions only. In contrast, OPE includes frequencies covering in all directions of the plane due to the product terms, resulting in a better expression capability.

Remark 3. Unlike LIIF [5] that uses an MLP to approximate the image function f, our method completely eliminates the need of MLP.

3.2. OPE-Upscale Module

We project the latent code onto the OPE basis. OPE with a sufficiently long latent code could represent an image directly in a continuous manner. However, it suffers from long embedding time and is unstable for representing high-frequency details locally, similar to the limitations

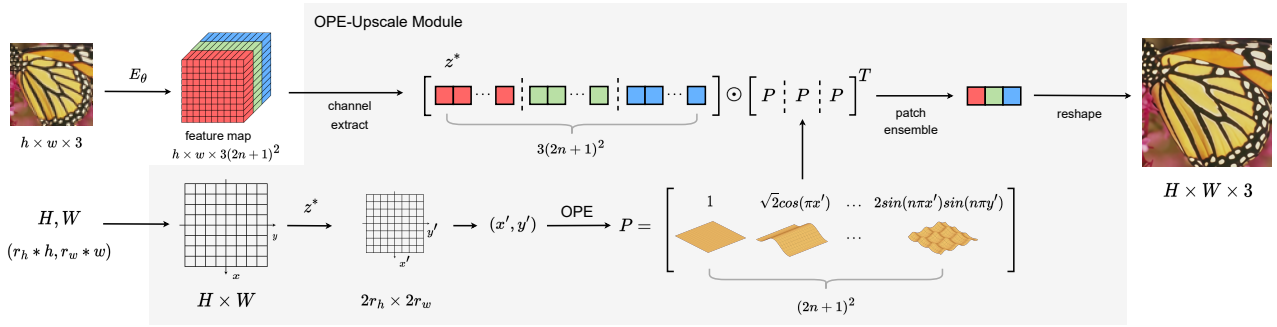


Figure 2. **OPE-Upscale Module for arbitrary-scale SR.** The encoder E_θ is the only trainable part. With a pre-defined maximum frequency n of OPE, the OPE-Upscale Module (shaded in grey) takes the feature map from E_θ and the target resolution H, W as input, and renders the pixels of the target SR image in parallel. $\odot$ is the matmul function that returns the product of $z^* \in \mathbb{R}^{1\times 3(2n+1)^2}$ and OPEs $\in \mathbb{R}^{3(2n+1)^2 \times 1}$ per color channel.

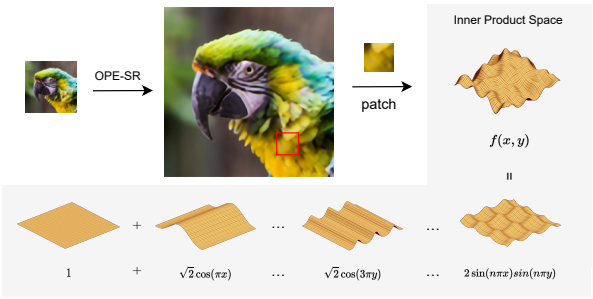


Figure 3. **OPE-based patch representation.** An image patch can be represented as a linear combination of basic plane waves in a continuous manner. The representation is a 2D extension to [46, Chap. 3.3.1]. Refer to the supplementary material for more details.

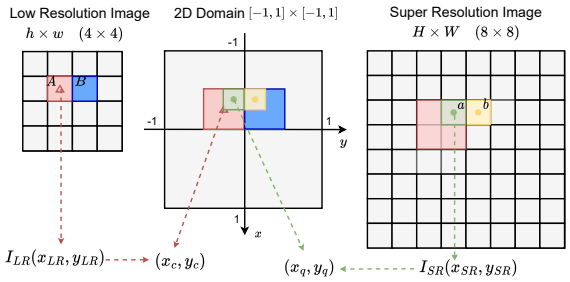


Figure 4. **Illustration of the mapping from the input LR image to the output SR image.** LR image, feature map and SR image are all divided into the same domain $[-1, 1] \times [-1, 1]$. Each LR pixel $I_{LR}(x_{LR}, y_{LR})$ corresponds to a latent code with coordinates (x_c, y_c), while each SR pixel $I_{SR}(x_{SR}, y_{SR})$ corresponds to a latent code with coordinates (x_q, y_q).

of Fourier transform to describe local information. To address this issue, we represent the input image as the seamless stitching of local patches, whose latent codes are extracted from a feature map over the channel dimension[2]. As shown in Fig. 2, the OPE-upscale module takes both the target resolution $H = r_h \cdot h$, $W = r_w \cdot w$ and the feature map $\in \mathbb{R}^{h\times w\times 3(2n+1)^2}$ from the deep encoder E_θ as inputs and computes target pixels in parallel.

Feature map rendering. As shown in Fig. 4, to render a target image I_{SR} with size $H \times W$ from a LR image I_{LR} with size $h \times w$, OPE-Upscale Module firstly divide a 2D domain $[-1, 1]\times[-1, 1]$ into $H\times W$ regions with equal size, so that every pixel in I_{SR} will be associated with an absolute central coordinates (x_q, y_q) in corresponding region. Secondly, the latent codes in the feature map (same dimension with I_{LR}) also possess corresponding central coordinates $(x_c, y_c) \in [-1, 1] \times [-1, 1]$ by dividing same 2D domain into $h \times w$ regions, therefore, given a target image pixel with (x_q, y_q), a specific latent code $z^* \in \mathbb{R}^{1\times 3(2n+1)^2}$ with coordinates (x_c, y_c), which has the smallest distance from (x_q, y_q) could be found. As shown in Eq.(6) and Eq.(7), a render function $\mathcal{R}$ takes two parts of inputs: z^* and (x'_q, y'_q), to calculate final target pixel value as following:

$$I_{SR}(x_q, y_q) = \mathcal{R}(z^*, (x'_q, y'_q)) \tag{6}$$

$$x'_q = (x_q - x_c) \cdot h, \quad y'_q = (y_q - y_c) \cdot w \tag{7}$$

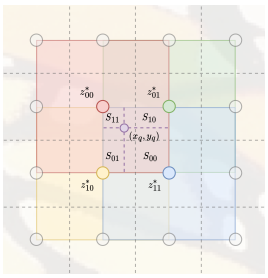
where z^* is the nearest latent code we found, and (x'_q, y'_q) are relative coordinates, which are calculated based on Eq.(7) to rescale the absolute coordinates (in domain $[-1, 1] \times [-1, 1]$) by times h and w, which is taken as input by function $\mathcal{R}$ to render target pixel. $\mathcal{R}$ has the similar calculation as Eq.(5) while the difference is it repeats OPE three times to adapt z^* and calculate the linear combination per color channel. In this way, our OPE-upscale module successfully deals with arbitrary size I_{SR} by processing each pixel by $\mathcal{R}$, in which feature map rendering process is parameter-free with high computing efficiency and less memory consumption (which has been confirmed by the experiments in Sec. 4.3).

[2]To ensure compatibility with color images, we adjust the output channel of the encoder to $3(2n + 1)^2$, where n is the pre-defined maximum frequency n of OPE.

Patch ensemble. As shown in Fig. 4, when moving (x_q, y_q) from location a to b, the pixel value in the target image I_{SR} may change abruptly, as well as the nearest latent code z^*. To address this discontinuity issue, we propose a patch ensemble, which is a local ensemble styled interpolation using relative coordinates. Instead of finding a single nearest latent code for (x_q, y_q), we select the nearest *four* neighbouring latent codes z_t^* with corresponding central coordinates (x_t, y_t), where $t \in \{00, 01, 10, 11\}$. Then we calculate the relative coordinates x'_q and y'_q as

$$x'_q = \frac{(x_q - x_t) \cdot h}{2}, \quad y'_q = \frac{(y_q - y_t) \cdot w}{2}, \tag{8}$$

which can guarantee $x'_q, y'_q \in [-1, 1]$. As the inset shows, for the four adjacent pixels from the low resolution image I_{LR} (i.e. related to latent codes z^*_{00}, z^*_{01}, z^*_{10} and z^*_{11}), their corresponding patches in super resolution image I_{SR} are colored in red, green, yellow and blue, respectively. The pixel color in I_{SR} is not solely dependent on the nearest latent code but considers the four neighboring latent codes. Specifically, using the rendering function $\mathcal{R}$ and the diagonal rectangle areas s_t as weights, we compute the pixel value as a weighted sum



$$I_{SR}(x_q, y_q) = \sum_{t \in \{00,01,10,11\}} \frac{s_t}{S} \cdot \mathcal{R}(z_t^*, (x'_q, y'_q)), \tag{9}$$

where $S = \sum_t s_t$ is the sum of areas. Considering the contribution of each latent code allows us to integrate the adjacent latent codes with different significance, thereby providing a seamless stitching of adjacent patches. We call Eq.(9) local ensemble styled interpolation since it takes a similar form of the local ensemble in LIIF [5].

3.3. Maximum Frequency n

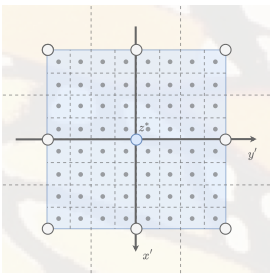
Selecting a proper maximum frequency n plays an important role in designing and implementing the OPE-upsampling module since it directly determines the network architecture and also has effects on the performance of different SR scales. Given a high resolution image I_{HR} with size $H \times W$, n and r, we aim to obtain a feature map with size $\frac{H}{r} \times \frac{W}{r}$, then we re-render the obtained feature map with the selected n. By the comparison of the rendered I_{SR} , we present the performance of $n \in \{1, 2, \cdots, 8\}$ under different r values (SR scale), as show in Tab. 1, and select the n with the best performance (the details would be discussed Sec. 4.1). To be specific, we use Eq.(10) as the basic theory and use Eq.(11) to infer the feature map. First, similar to calculate the projection of a normal vector on orthogonal basis, we can calculate projections (or so-called latent code) $Z \in \mathbb{R}^{1 \times (2n+1)^2}$ of $f(x, y)$ in Eq.(5) as follows:

$$Z[i] = \frac{1}{4} \int_{-1}^{1} \int_{-1}^{1} f(x, y) P[i](x, y) \mathrm{d}x \mathrm{d}y \tag{10}$$

where $P[i](x, y)$ is a bivariate function taken from the i-th position of OPE and $Z[i]$ is the corresponding projection. Based on Eq.(10) and taking both the discreteness of an image and the design of OPE-Upscale Module into consideration, we calculate the feature map of an image I_{HR} with down-sampling scale r as follows:

$$z^*[i] = \frac{1}{4} \sum_{x'}^{2r} \sum_{y'}^{2r} I_{HR}(x', y') P[i](x', y') \tag{11}$$

It can be considered as the inverse operation of Eq.(9). Take the right inset as an example. We choose the HR image as the ground truth (e.g. HR in Fig. 5), when $r = 4$, every latent code z^* corresponds to a 8×8 patch of HR (gray points) in relative coordinate domain (blue region). To calculate the i-th position of z^*, we multiply every HR pixel value $I_{HR}(x', y')$ and basis value $P[i](x', y')$ together and finally sum them. After getting the feature map, we render it to the same size of I_{HR} via OPE-Upscale Module and calculate their Peak Signal-to-Noise Ratio (PSNR).



4. Experiments

4.1. Parameter Setting

We evaluated the performance of different maximum frequencies n on 50 images from the DIV2K validation set [1] under different scales r. Since our method is a local representation, we do not use a large n. As shown in Tab. 1, the optimal sampling frequency for a given r_i is always $r_i - 1$. This observation can also be explained by the Nyquist–Shannon sampling theorem. For example, when $r_i = 4$, there are 8×8 sampling points for every latent code to "fit", so the maximum frequency that can be recovered from these sampling points should be less than 4. We also tested larger frequency with $r_i \leq n \leq 2 \times r_i$, reaching the upper limit that equals the number of sampling points. We further visualize the reconstructed images for different n. As shown in Fig. 5, with scale factor $\times 4$, the larger frequency ($n > 3$) brings redundant high-frequency information that sharpens the resulting images.

Based on the above analysis, we decide to choose $n = 3$ as the maximum frequency for our OPE-Upscale Module. This is because existing arbitrary-scale SR methods, such as LIIF [5] and LTE [32], are trained with random scale factors up to 4, and the frequency $n = 4 - 1 = 3$ is sufficient to fully capture the ground truth information. Al-

n	×2	×3	×4	×5	×6	×7	×8
1	**31.1951**	28.6083	26.4424	25.0485	24.1114	23.3423	22.7898
2	30.7472	**33.6586**	31.2091	28.8022	27.3701	26.1913	25.3838
3	22.1871	33.6585	**35.1983**	32.4011	30.6135	28.8964	27.8159
4	12.1230	28.6083	34.9631	**34.6294**	34.0979	31.4462	30.2865
5	-	22.8465	29.9512	34.6293	**37.3704**	33.7285	32.8190
6	-	22.8465	24.3122	32.4011	37.1506	**35.3046**	35.8250
7	-	-	19.1593	28.8022	33.3039	35.3046	**39.1160**
8	-	-	12.0863	25.0485	29.0966	33.7286	38.9863

Table 1. **Representation performance (PSNR (dB)).** The best value for each upsampling factor is bolded.

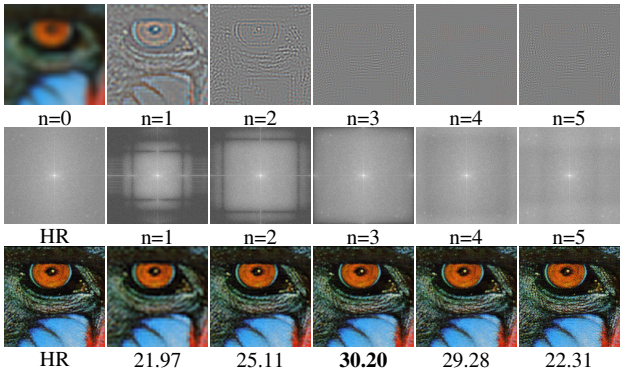


Figure 5. **Qualitative comparation of different OPE frequency n under scale factor** $\times 4$ **(PSNR (dB)).** 1-th row: residuals from $n = 0$ in image time domain. 2-th row: fourier frequency domain of HR and rendered image with n. 3-th row: HR image and rendered image with n.

though a larger n could represent more detailed patches of the target SR image, it would also introduce redundant high-frequency information and potentially increase computation time and memory consumption during training. Therefore, we choose $n = 3$ as the maximum frequency for our OPE-Upscale Module to balance performance and efficiency.

4.2. Training

Datasets. Similar to [5, 32], we use the DIV2K dataset [1] of the NTIRE 2017 Challenge [63] for training. For testing, we use the DIV2K validation set [1] with 100 images and four benchmark datasets: Set5 [4], Set14 [76], B100 [40], and Urban100 [21]. We use PSNR as the quality measure.

Implementation details. We mainly follow the prior implementation [5, 32] for arbitrary-scale SR training after replacing their upsampling module with OPE-Upscale. We use EDSR-baseline [35] and RDN [80] without their upsampling modules as the encoder, which is the only trainable part of our network. We use 48×48 patches cropped from training set as inputs, L1 loss and Adam [26] optimizer for optimization. The network was trained for 1000 epochs with batch size 16, while the initial learning rate is 1e-4 and decayed by factor 0.5 every 200 epochs. More implementation details are presented in supplementary material.

4.3. Evaluation

Quantitative results. Tab. 2 and Tab. 3 report quantitative results of OPE and the SOTA arbitrary-scale SR methods on the DIV2K validation set and the benchmark datasets. It is worth noting that we focus on finding an alternative of MLP with position encoding, rather than enhancing it like LTE [32]. We observe that our method achieves comparable results (less than 0.1dB on DIV2K and less than 0.15dB on benchmark), which indicates that our method is a feasible analytical solution with good performance and efficient parameter-free module. As shown in Tab. 2, EDSR [35] and RDN [80] are our selected encoders, and we achieve the highest efficiency (i.e. the shortest inference time in red number) comparing to all the other baselines with both encoders. The higher the scale factor, the better result we achieve. Specifically, in out-scale SR ($\times 6$ to $\times 30$), our method outperforms most baselines and just has a small gap with LTE (less than 0.1dB). Such results demonstrate that our method has rich representation capability. We also compared with the benchmark dataset, as shown in Tab. 3, we keep comparable results to baselines (the gap is less than 0.15dB). However, as a nonlinear representation method, MLP still has advantages over our linear representation with low scale factors. See Sec. 5 for discussion on this issue.

Qualitative results. Fig. 6 provides qualitative results with SOTA methods by using different scale factor. We show competitive visual quality against others, more results are provided in supplementary material. From the local perspective, LIIF and LTE only generate smooth patches, while our OPE with max frequency 3 is enough to achieve similar visual quality. We also notice LIIF [5] has artifact (vertical stripes) in the 1st row, this is a common drawback for implicit neural representation and is hard to be explained. However, with our image representation, there is no artifacts. In the 2nd row, we could observe a sprout (in red rectangle) in the GT, the same region of LIIF is vanished, and the boundary of our sprout is more obvious than LTE.

Computing efficiency. We measure computing efficiency with MACs (multiply-accumulate operations), FLOPs (floating point operations) and actual running time. In Tab. 4 column 2-3, judged by the time complexity measured by the number of operations, we save 2 orders of magnitude. In our upsampling module, there is only one matrix operation and essential position encoding between input and output. In Tab. 2 we show shortest inference time benefiting from our compact SR framework. To further demonstrate our time advantage on large size images, we take 256×256 as LR input of encoder and calculate time consumption of upsampling module with scale factor $\times 4$-$\times 30$ on NVIDIA RTX 3090. As shown in Tab. 5 , our upsampling module shows 26%-57% time advantage, this advantage keeps growing with larger scale factor. Notice We do not take advantage of GPU acceleration to design the upsampling

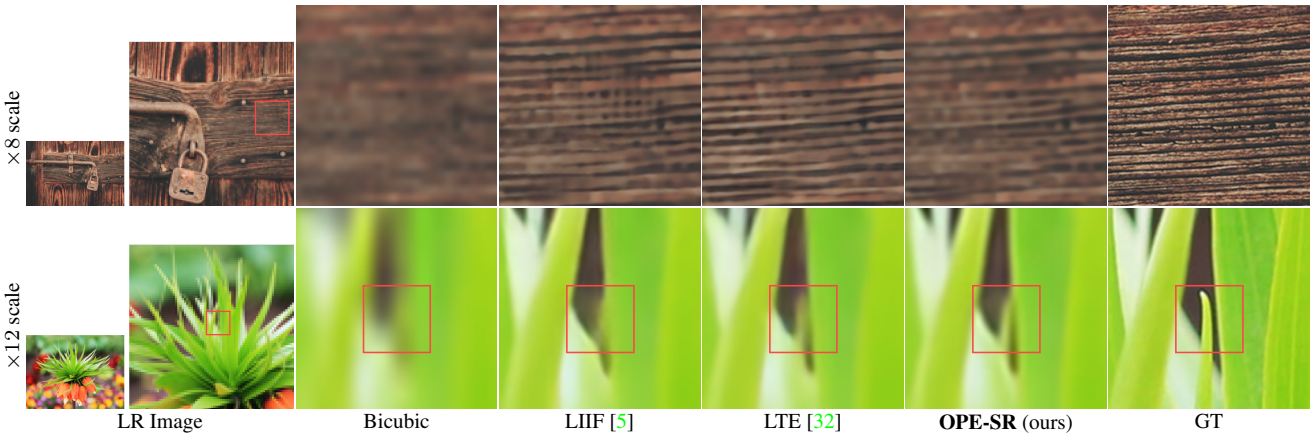


Figure 6. **Qualitative comparison** with SOTA methods for arbitrary-scale SR. RDN [80] is used as encoder for all methods.

Method	In-scale			Out-scale				
	×2	×3	×4	×6	×12	×18	×24	×30
Bicubic [35]	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline [35]	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-MetaSR♯ [5, 18]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF [5]	34.67 / 1702	30.96 / 1277	29.00 / 1144	26.75 / 1046	23.71 / 965	22.17 / 953	21.18 / 951	20.48 / 947
EDSR-baseline-LTE [32]	34.72 / 1158	31.02 / 1079	29.04 / 1045	26.81 / 1023	23.78 / 1007	22.23 / 1005	21.24 / 1003	20.53 / 1000
EDSR-baseline-OPE (ours)	34.34 / 476	30.94 / 395	29.02 / 364	26.77 / 348	23.74 / 322	22.21 / 318	21.21 / 314	20.52 / 311
RDN-MetaSR♯ [5, 18]	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF [5]	34.99 / 3107	31.26 / 2073	29.27 / 1513	26.99 / 1248	23.89 / 1025	22.34 / 994	21.31 / 991	20.59 / 972
RDN-LTE [32]	35.04 / 2549	31.32 / 1839	29.33 / 1420	27.04 / 1184	23.95 / 1049	22.40 / 1027	21.36 / 1025	20.64 / 1014
RDN-OPE (ours)	34.52 / 2277	31.17 / 1497	29.26 / 1039	26.98 / 813	23.91 / 663	22.36 / 623	21.34 / 596	20.63 / 590

Table 2. **Quantitative comparison** with the SOTA methods for arbitrary-scale SR on the DIV2K validation set (PSNR (dB) / running time (ms per image)). ♯ indicates implementation in LIIF [5]. With a parameter-free upsampling module, we narrow the gap between SOTA and ours in most results less than 0.1dB (blue) and obtain the shortest inference time (red).

module carefully, with hardware optimization, we believe our time advantage could be much larger thanks to fewer number of operations required.

Memory consumption. In Tab. 4 column 4-5 we compare GPU memory consumption of OPE-Upscale Module with LIIF [5] and LTE [32] under training mode and testing mode of Pytorch [48]. For training mode, we use a 48×48 patch as input and sample 2304 pixels as output following the default training strategy in arbitrary-scale SR works. For testing mode, we use 512×512 image as input with scale factor 4. As a interpretable image representation without network parameters, OPE-Upscale Module saves memory of intermediate data (e.g. gradients, hidden layer outputs), and this advantage is fully reflected in training mode.

Flipping consistency. As described in Sec. 1, the INR-based upsampling module like [5] is sensitive for the flipping of feature map. However, our method solves this problem completely and elegantly. The orthogonal basis of OPE is based on symmetric sinusoidal function, which leads to advantage of our method for keeping the flipping consistency. Also, more samples are provided in supplementary material for verifying other more flipping transforms.

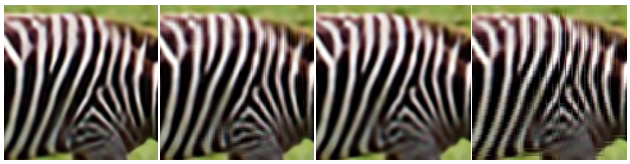


Figure 7. **Visual results of the ablation study on patch ensemble.** See the text for details.

4.4. Ablation Studies

In order to examine the effects of local ensemble styled interpolation (I)-Eq.(9) and extension of relative coordinate domain (E)-Eq.(8) in patch ensemble, we conducted experiments using four different settings with EDSR-baseline as the encoder. The four settings were: 1) OPE: OPE-Upscale module with I and E; 2) OPE-E: OPE-Upscale module without E but with I; 3) OPE-I: OPE-Upscale module without I but with E; and 4) OPE-IE: OPE-Upscale module with neither I nor E (that is, without patch ensemble).

Fig. 7 and Tab. 6 present the comparison results of different settings. In the OPE-E setting, only the nearest latent code to the query point can be rescaled to [-1,1] × [-1,1], while the relative coordinates of the other three latent codes

Method	Set5					Set14					B100					Urban100				
	In-scale			Out-scale		In-scale			Out-scale		In-scale			Out-scale		In-scale			Out-scale	
	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8
RDN [80]	38.24	34.71	32.47	-	-	34.01	30.57	28.81	-	-	32.34	29.26	27.72	-	-	32.89	28.80	26.61	-	-
RDN-MetaSR♯ [5, 18]	38.22	34.63	32.38	29.04	26.96	33.98	30.54	28.78	26.51	24.97	32.33	29.26	27.71	25.90	24.83	32.92	28.82	26.55	23.99	22.59
RDN-LIIF [5]	38.17	34.68	32.50	29.15	27.14	33.97	30.53	28.80	26.64	25.15	32.32	29.26	27.74	25.98	24.91	32.87	28.82	26.68	24.20	22.79
RDN-LTE [32]	38.23	34.72	32.61	29.32	27.26	34.09	30.58	28.88	26.71	25.16	32.36	29.30	27.77	26.01	24.95	33.04	28.97	26.81	24.28	22.88
RDN-OPE (ours)	37.60	34.59	32.47	29.17	27.22	33.39	30.49	28.80	26.65	25.17	32.05	29.19	27.72	25.96	24.91	31.78	28.63	26.53	24.06	22.70

Table 3. **Quantitative comparison** with SOTA methods for arbitrary-scale image SR on benchmark datasets (PSNR (dB)). ♯ indicates implementation in LIIF [5]. We narrow the gap between SOTA and ours in most results less than 0.15dB (blue number). For large scale factor, we keep comparable results to MetaSR [18] and LIIF [5]. The defect in low scale factor will be analysed in Sec. 5.

Method	Params	MACs	FLOPs	Mem (training)	Mem (Test)
LIIF	0.35 M	429 K	6.2 G	85.1 + 1.9 M	32 + 96 M
LTE	0.26 M	526 K	7.5 G	97.8 + 1.9 M	64 + 96 M
OPE (ours)	**0 M**	**6 K**	**85 M**	**0 + 1.9 M**	**0 + 96 M**

Table 4. **Parameter number, time complexity and memory consumption.** MACs: multiply-accumulate operations, FLOPs: floating point operations, Mem: intermediate data + essential output for GPU memory consumption. We use $n = 3$ as maximum frequency of OPE and test in training mode and test mode on Pytorch with tool: torch.cuda.memory_allocated(). Training mode: 48^2 to 2304 pixels, test mode: 512^2 to 2048^2.

Method	×4	×8	×12	×16	×20	×24	×30
LIIF	382	1521	3530	6004	10274	18350	27866
LTE	376	1490	3340	5922	10268	18340	27838
OPE	**277**	**1125**	**2495**	**3719**	**5673**	**8366**	**12012**
Percentage	28%	26%	30%	39%	45%	55%	57%

Table 5. **Rendering time of upsampling module** (ms per image) with an input resolution of 256×256. The last row shows the time saving percentage achieved by our method. We use $n = 3$ as the maximum frequency of OPE. Our method provides a time advantage, which increases as the rendering resolution increases. On average, we achieve a 40% reduction in rendering time.

	In-scale			Out-scale	
	×2	×3	×4	×6	×8
OPE	**33.29**	**30.29**	**28.65**	**26.46**	**24.98**
OPE-E	33.27	30.23	28.56	26.34	24.82
OPE-I	33.28	30.26	28.63	26.44	24.97
OPE-IE	33.20	30.09	28.44	26.25	24.70

Table 6. **Ablation studies on Set14.** EDSR-baseline [35] is used as encoder.

cannot be rescaled, resulting in periodic stripes in the resulting SR image. On the other hand, the OPE-I result shows no obvious discontinuity between patches since the extension plays a positive role, but this means that only a small region of the patch is presented in the target image. Lastly, the OPE-IE setting shows an obvious thick boundary between patches, indicating that both extension and interpolation are necessary for the best performance.

5. Discussions

We observed that our quantitative results decreases at low scale factors, especially when the input size is small, such as with benchmark datasets. See Tab. 3 and Tab. 2. This is due to the fact that a smaller target size ($W \times H$) leads to a larger grid in the 2D domain ($[-1, 1] \times [-1, 1]$), where utilizing only one central point value to represent the entire larger grid would result in a loss of detailed information compared to a smaller grid. Since the 2D domain we use is continuous, the higher the resolution of the target image, the stronger the representation ability we can achieve. MLP-based representations, such as [5] and [32], can overcome this issue through nonlinear operations. In our method, this defect can be ignored for high SR scale factors where pixels are dense, but for low scale such as $\times 2$ or $\times 3$, our performance may slightly degrade. A possible way to address this issue is to sample more points for every grid region and calculate their mean value, with careful consideration of the time consumption trade-off.

6. Conclusion

In this paper, we proposed an interpretable method for continuous image representation without implicit neural networks. Our method leverages a novel position encoding technique called orthogonal position encoding, which takes the form a 2D-Fourier series and corresponds to 2D image coordinates. As a set of orthogonal basis in inner product space, OPE is both interpretable and rich in representation. Building on OPE, we introduced the OPE-Upscale Module, a parameter-free approach for arbitrary-scale image super-resolution that simplifies the existing deep SR framework, leading to high computing efficiency and less memory consumption. Our OPE-Upscale Module can be easily integrated into existing image super-resolution pipelines, and extensive experiments demonstrate that our method achieves competitive results with the state-of-the-art.

It is worth noting that the overall efficiency of SR framework depends on both the encoder and decoder. Since our work focuses on decoder design and efficiency, we leave the development of high-efficient encoder as future work.

Acknowledgement This project was partially supported by NSFC Grants (61702363, 62222311, 62201274) and the the Ministry of Education, Singapore, under its Academic Research Fund Grants (MOE-T2EP20220-0005 & RG20/20).

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. 5, 6

[2] Woong Bae, Jaejun Yoo, and Jong Chul Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 145–153, 2017. 2

[3] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003. 2

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6

[5] 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. 1, 2, 3, 5, 6, 7, 8

[6] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. *Advances in Neural Information Processing Systems*, 33:4479–4488, 2020. 2

[7] Jooyoung Choi, Jungbeom Lee, Yonghyun Jeong, and Sungroh Yoon. Toward spatially unbiased generative models. *arXiv preprint arXiv:2108.01285*, 2021. 2

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 2

[9] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016. 2, 3

[10] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 3

[11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[12] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3599–3608. IEEE, 2019. 2

[13] Dario Fuoli, Luc Van Gool, and Radu Timofte. Fourier space losses for efficient perceptual image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2360–2369, 2021. 2

[14] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 1

[15] Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga. Deep wavelet prediction for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 104–113, 2017. 2

[16] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. 2

[17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 1

[18] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1575–1584, 2019. 1, 3, 7, 8

[19] Binbin Huang, Xinhao Yan, Anpei Chen, Shenghua Gao, and Jingyi Yu. Pref: Phasorial embedding fields for compact neural representations. *arXiv preprint arXiv:2205.13524*, 2022. 1

[20] Huaibo Huang, Ran He, Zhenan Sun, and Tieniu Tan. Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1689–1697, 2017. 2

[21] 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. 6

[22] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13919–13929, 2021. 2

[23] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[24] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 2

[25] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[27] Tom Koornwinder. Two-variable analogues of the classical orthogonal polynomials. In *Theory and application of special functions*, pages 435–495. Elsevier, 1975. 2

[28] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 2

[29] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with

deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018. 2

[30] Seyed Mehdi Lajevardi and Zahir M Hussain. Higher order orthogonal moments for invariant facial expression recognition. *Digital Signal Processing*, 20(6):1771–1779, 2010. 2

[31] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[32] 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. 1, 2, 3, 5, 6, 7, 8

[33] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3867–3876, 2019. 2

[34] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 3

[35] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 2, 6, 7, 8

[36] Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, and Ming-Hsuan Yang. Infinitygan: Towards infinite-pixel image synthesis. *arXiv preprint arXiv:2104.03963*, 2021. 2

[37] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[38] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 2

[39] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007. 2

[40] 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. 6

[41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 1

[42] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[43] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[44] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 1

[45] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019. 1

[46] Alan V. Oppenheim and Alan S. Willsky. *Signals and Systems*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1997. 4

[47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7

[49] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 1

[50] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 1, 2

[51] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *JOSA A*, 18(10):2448–2459, 2001. 2

[52] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 2

[53] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 2

[54] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan

Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 3

[55] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018. 2

[56] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2

[57] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 1

[58] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[59] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002. 2

[60] Sanghyun Son and Kyoung Mu Lee. Srwarp: Generalized image super-resolution under arbitrary transformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7782–7791, 2021. 3

[61] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017. 2

[62] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017. 2

[63] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 6

[64] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE international conference on computer vision*, pages 4799–4807, 2017. 2

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[66] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4801–4810, 2021. 3

[67] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021. 3

[68] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 864–873, 2018. 2

[69] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 864–873, 2018. 2

[70] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020. 1, 2, 3

[71] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008. 2

[72] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021. 1, 2, 3

[73] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2

[74] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2

[75] Youngho Yoon, Inchul Chung, Lin Wang, and Kuk-Jin Yoon. Spheresr: 360deg image super-resolution with arbitrary projection via continuous spherical image representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5677–5686, 2022. 3

[76] 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. 6

[77] Hui Zhang, Huazhong Shu, Guoniu N Han, Gouenou Coatrieux, Limin Luo, and Jean Louis Coatrieux. Blurred image recognition by legendre moment invariants. *IEEE Transactions on Image Processing*, 19(3):596–611, 2009. 2

[78] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[79] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 2

[80] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 6, 7, 8

[81] Man Zhou, Hu Yu, Jie Huang, Feng Zhao, Jinwei Gu, Chen Change Loy, Deyu Meng, and Chongyi Li. Deep fourier up-sampling. *arXiv preprint arXiv:2210.05171*, 2022. 2

[82] Hongqing Zhu. Image representation using separable two-dimensional continuous and discrete orthogonal moments. *Pattern Recognition*, 45(4):1540–1558, 2012. 2