Local Texture Estimator for Implicit Representation Function

Jaewon Lee   Kyong Hwan Jin[*]

Daegu Gyeongbuk Institute of Science and Technology (DGIST), Korea

{ljw3136, kyong.jin}@dgist.ac.kr

https://github.com/jaewon-lee-b/lte

Abstract

Recent works with an implicit neural function shed light on representing images in arbitrary resolution. However, a standalone multi-layer perceptron shows limited performance in learning high-frequency components. In this paper, we propose a Local Texture Estimator (LTE), a dominant-frequency estimator for natural images, enabling an implicit function to capture fine details while reconstructing images in a continuous manner. When jointly trained with a deep super-resolution (SR) architecture, LTE is capable of characterizing image textures in 2D Fourier space. We show that an LTE-based neural function achieves favorable performance against existing deep SR methods within an arbitrary-scale factor. Furthermore, we demonstrate that our implementation takes the shortest running time compared to previous works.

1. Introduction

Single image super-resolution (SISR) is one of the most fundamental problems in computer vision and graphics. SISR aims to reconstruct high-resolution (HR) images from its degraded low-resolution (LR) counterpart. Dominant approaches [3, 5, 14, 15, 18, 24, 34, 35] are to extract feature maps using a deep vision architecture and then upsample to HR images at the end of a network. However, we need to train and store several models for each scale factor when an upsampler is implemented by sub-pixel convolution [24]. In contrast, arbitrary-scale SR methods [4, 9] are promising since such ideas pave the way to restore images in a continuous manner with only a single network.

Recently, implicit neural functions parameterized by a multi-layer perceptron (MLP) achieved remarkable performance in representing continuous-domain signals, such as images [4], occupancy [19], signed distance [21], shape representation [11], and view synthesis [20, 26]. Such MLPs take coordinates as inputs and are trained in a framework of gradient descent optimization and machine learning. In-
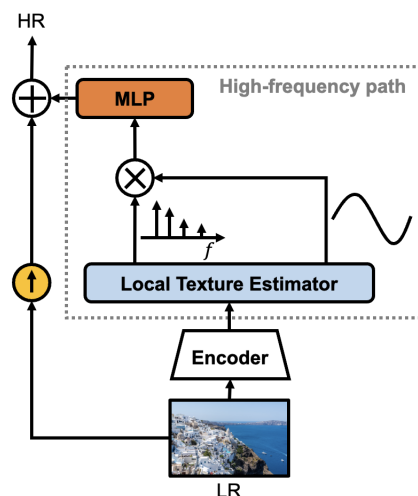

Figure 1. **Overview of Local Texture Estimator (LTE).** Our LTE estimates dominant frequencies and corresponding Fourier coefficients for natural images. Then, an MLP makes use of the estimated essential Fourier information to reconstruct an HR image in arbitrary resolution. We provide low-frequency information by adding an upscaled LR image to the output of MLP.

spired by recent progress in implicit representation, LIIF [4] replaced sub-pixel convolution with MLPs to accomplish arbitrary-scale SR, even at substantial scale factors.

One limitation of implicit neural representations is that a standalone MLP is biased towards learning low-frequency components [23] and fails to capture fine details [29]. Such phenomenon is referred to as spectral bias, and recent lines of research in resolving this problem are projecting input coordinates into a high-dimensional Fourier feature space [20, 29] or substituting a ReLU with a sinusoidal activation [25]. Motivated from previous works, we study arbitrary-scale SISR problems through the lens of Fourier analysis.

In this paper, we propose a Local Texture Estimator (LTE), a dominant-frequency estimator for natural images, to allow an implicit function to learn fine details while restoring images in arbitrary resolution. We assume that an implicit function prioritizes learning image textures when utilizing dominant frequencies of images, as described in

---

[*]Corresponding author.

Fig. 1. Let us take an example from an image with vertical textures. Intuitively, the dominant frequencies of such an image are located on an x-axis in 2D Fourier space. We observe that LTE is capable of extracting such dominant frequencies, characterizing image textures in 2D Fourier space, when jointly trained with a deep SR architecture, such as EDSR [15], RDN [35], and SwinIR [14]. In addition to extracting dominant frequencies, we show that estimating Fourier coefficients is also essential in improving a representational power of an implicit function in Sec. 5.3.

Short computation time is essential in SR applications. In addition, restoring larger than 2K-sized images is a memory-consuming task. Hence, we study the computation time of our LTE at various memory conditions and demonstrate that our approach is faster compared to previous works regardless of memory setting in Sec. 6.

In summary, our main contributions are as follows:

- We point out that a deep SR network followed by LTE is capable of estimating dominant frequencies and corresponding Fourier coefficients for natural images.

- We show that an implicit representation function for arbitrary resolution prioritizes learning high-frequency details when essential Fourier information is estimated by LTE.

- We investigate the computation time of our LTE at several memory settings and demonstrate that our approach is faster compared to previous works.

2. Related Work

Implicit neural representation Based on the fact that neural networks are universal function approximators [8], coordinate-based MLP is widely applied to represent continuous-domain signals in computer vision and graphics. Such MLPs are a memory-efficient framework for HR data since the amount of memory to store MLPs is independent of data resolution. However, pre-trained MLPs show limited performance in representing unseen data, requiring training per each signal. To overcome this generalization issue, [25, 26] trained representation with a hypernetwork, which maps latent variables to weights of an MLP as in [7]. In [19, 21], an MLP takes not only coordinates but also latent variables as inputs to enable representation to be a function of data. Recently, meta-learned initialization [28] has been proposed, which provides a strong prior for representing signals, leading to fast convergence. Our work is mainly related to local implicit neural representation [4, 11]. Such approaches assume that latent variables are evenly distributed over space, allowing an implicit function to focus on learning local features. Inspired by this, we hypothesize that the local region in HR representation shares the same image textures.

Spectral bias Recent works [20, 23, 25, 29] have shown that a standard MLP with ReLUs shows limited performance in representing high-frequency textures. Such a phenomenon is referred to as spectral bias, and various methods have been proposed to alleviate this problem. Recently, SIREN [25] used the sine layer as a non-linear activation instead of a ReLU, resulting in fast convergence and high data fidelity. Other approaches [20, 29] are to map input coordinates into high dimensional Fourier space by using position encoding or Fourier feature mapping before passing an MLP. Frequencies are fixed to the power of two [20] or randomly sampled from Gaussian distribution [29]. Unlike previous works, dominant frequencies from our LTE are data-driven and characterize high-frequency textures in 2D Fourier space.

Deep SR architecture After ESPCN [24] has proposed an efficient learnable upsampling module using pixel-shuffling, numerous CNN-based approaches [5, 15, 18, 34, 35] have been studied to improve the representational power of models. Such approaches have exploited more complicated neural network architecture designs, such as residual block [15], densely connected residual block [35], channel attention [5, 34], or non-local neural networks [18]. Inspired by the success of the self-attention mechanism in high-level vision tasks [6, 16], general-purpose image processing transformers, such as IPT [3] or SwinIR [14], have been proposed. Even though transformer-based SR architectures using a large dataset surpass CNN-based architectures in performance, these approaches nevertheless need to utilize a specific upscale module for each upsampling rate.

Arbitrary-scale SR Our work is highly related to SR tasks within an arbitrary-scale factor [4, 9, 27, 31], which is convenient and efficient for practical benefits. Training and storing models for each specific scale factor [3, 14, 15, 24, 34, 35] is unfeasible when considering limited memory resources. Since MetaSR [9] first proposed an arbitrary-scale SR method with a single model, various approaches have been explored. Recently, ArbSR [31] has been proposed as a general plug-in module using conditional convolutions. ArbSR conducts an SR with different scales along horizontal and vertical axes, respectively. More recently, SRWarp [27] successfully transformed an LR image into any shape in HR representation using a differentiable adaptive warping layer. Most related to ours is the model of [4]. Inspired by advancements in implicit neural representation, LIIF replaced sub-pixel convolution with an MLP, taking continuous coordinates and latent variables as inputs. Even though such a method outperformed previous works at large upsampling rates, structural distortion occurs at extreme scales. Instead of concatenating coordinates and latent variables, our LTE-based architecture transforms input coordinates into the Fourier domain using dominant frequencies extracted from latent variables.
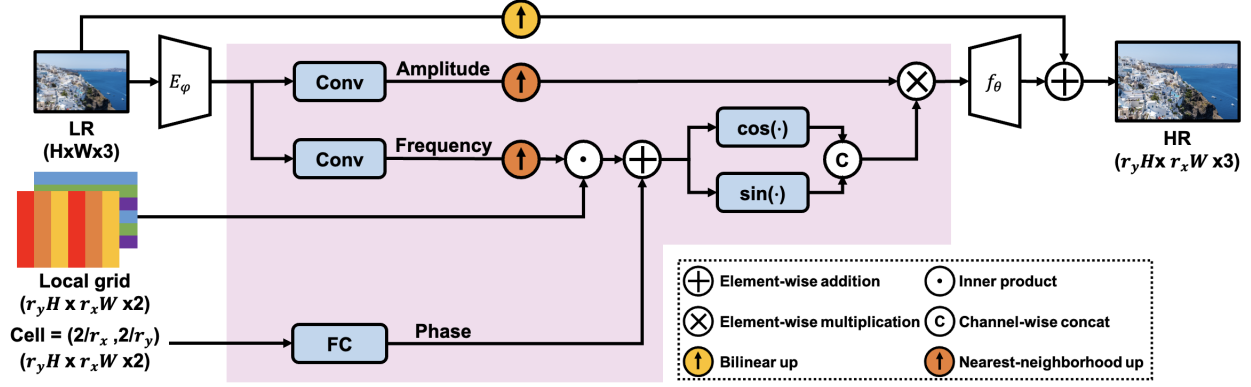
Figure 2. **Arbitrary-scale SR with our proposed Local Texture Estimator (LTE).** LTE-based arbitrary-scale SR architecture consists of an encoder (E_φ), LTE (a pink shaded region), a decoder (f_θ), and an LR skip connection. Inputs of LTE are as follows: feature map from the encoder, local grid, and cell. LTE transforms input coordinates into the Fourier domain using extracted amplitude, frequency, phase information. We add a bilinear upscaled LR image to the output of a decoder.

3. Problem Formulation

In this section, we aim to represent $\mathbf{I^{HR}} \in \mathbb{R}^{r_\mathbf{y} H \times r_\mathbf{x} W \times 3}$ from $\mathbf{I^{LR}} \in \mathbb{R}^{H \times W \times 3}$ given any fraction number $r_\mathbf{x}, r_\mathbf{y}$. We first review a continuous representation of an RGB image with a local implicit neural representation [4, 11]. Even though such approaches showed outstanding performance in representing continuous-domain signals, a standalone MLP fails to capture high-frequency details [23]. To overcome this spectral bias problem, we formulate a Local Texture Estimator (LTE), a dominant frequency estimator for natural images. Unlike prior arts [20, 29], estimated frequencies are data-driven and strongly correlated to image textures. Additionally, we introduce scale-dependent phase estimation and LR skip connection, which aid LTE in learning high-frequency textures.

Local implicit neural representation In local implicit neural representation [4, 11], a decoding function f_θ is shared by all images and is parameterized by an MLP with trainable weights θ. A decoder f_θ maps both latent tensors and local coordinates into RGB values; $f_\theta(\mathbf{z}, \mathbf{x})$: $(\mathcal{Z}, \mathcal{X}) \mapsto \mathcal{S}$. $\mathbf{z} \in \mathcal{Z}$ is a latent tensor from an encoder E_φ, $\mathbf{x} \in \mathcal{X}$ is a 2D coordinate in the continuous image domain, $\mathcal{S}$ is a space of predicted values from f_θ. For simplicity, we assume that a latent tensor $\mathbf{z} \in \mathbb{R}^{H \times W \times C}$ has the same width and height as $\mathbf{I^{LR}}$. Then, predicted RGB values ($\mathbf{s} \in \mathbb{R}^3$) at a coordinate $\mathbf{x} \in \mathbb{R}^2$ are estimated as

$$\mathbf{s}(\mathbf{x}, \mathbf{I^{LR}}; \Theta) = \sum_{j \in \mathcal{J}} w_j f_\theta(\mathbf{z}_j, \mathbf{x} - \mathbf{x}_j), \quad (1)$$

$$\text{where } \mathbf{z} = E_\varphi(\mathbf{I^{LR}}). \quad (2)$$

$\Theta = [\vec{\theta}; \vec{\varphi}]$, $\mathcal{J} \in \mathbb{Z}^4$ is a set of indices for four nearest (Euclidean distance) latent codes around $\mathbf{x}$, w_j is the bilinear interpolation weight corresponding to the latent code j

(referred to as the local ensemble weight [4], $\sum_j w_j = 1$), $\mathbf{z}_j \in \mathbb{R}^C$ is the $j-$th nearest latent feature vector from $\mathbf{x}$, and $\mathbf{x}_j \in \mathbb{R}^2$ is the coordinate of the latent code j. Given a series of M data points from N images such as $(\mathbf{x}_m, \mathbf{I}_n^{HR}(\mathbf{x}_m))$, $m = 1, \dots, M$ and $n = 1, \dots, N$, the learning problem is defined as follows:

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{m,n}^{M,N} \|\mathbf{I}_n^{HR}(\mathbf{x}_m) - \mathbf{s}(\mathbf{x}_m, \mathbf{I}_n^{LR}; \Theta)\|_1. \quad (3)$$

In practice, $\mathcal{X}$ spans $[-H, H]$ and $[-W, W]$ for two dimensions. Note that a step size (*Cell* in Fig. 2) of an output grid ($\mathbb{R}^{r_\mathbf{y} H \times r_\mathbf{x} W}$) and an input grid ($\mathbb{R}^{H \times W}$) is different. In [4, 11], their decoding function (f_θ) predicts continuous representation with a relative coordinate: $\mathbf{x} - \mathbf{x}_j$ ($|\mathbf{x} - \mathbf{x}_j| \leq \mathbf{1}$) known as *Local grid*. The same coordinate (*Local grid*) with [4, 11] is used for our work in order to represent a $r_\mathbf{x} \times r_\mathbf{y}$ local area in HR representation.

Learning dominant frequency component Recent works have shown that an MLP with ReLUs is biased towards learning low-frequency content [23]. To resolve this spectral bias problem of an implicit neural function, we propose a Local Texture Estimator (LTE), an essential Fourier information estimator for natural images. Inspired by position encoding [20] and Fourier feature mapping [29], LTE transforms input coordinates into the Fourier domain before passing an MLP. However, unlike [20, 29], estimated Fourier information is data-driven and reflects image textures in 2D Fourier space. The local implicit neural representation in Eq. (1) can be modified as follows:

$$\mathbf{s}(\mathbf{x}, \mathbf{I^{LR}}; \Theta, \psi) = \sum_{j \in \mathcal{J}} w_j f_\theta(h_\psi(\mathbf{z}_j, \mathbf{x} - \mathbf{x}_j)) \quad (4)$$

where $h_\psi(\cdot, \cdot)$ denotes the LTE, which is shift-invariant. LTE ($h_\psi(\cdot, \cdot)$) consists of three elements;(1) an amplitude

estimator ($h_a(\cdot) : \mathbb{R}^C \mapsto \mathbb{R}^{2K}$), (2) a frequency estimator ($h_f(\cdot) : \mathbb{R}^C \mapsto \mathbb{R}^{K \times 2}$), (3) a phase estimator ($h_p(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^K$). Thus, given a local-grid coordinate $\boldsymbol{\delta}(= \mathbf{x} - \mathbf{x}_j) \in \mathbb{R}^2$, the estimating function $h_\psi(\cdot, \cdot) : (\mathbb{R}^C, \mathbb{R}^2) \mapsto \mathbb{R}^{2K}$ is defined as

$$h_\psi(\mathbf{z}_j, \boldsymbol{\delta}) = \mathbf{A}_j \odot \begin{bmatrix} \cos(\pi \mathbf{F}_j \boldsymbol{\delta}) \\ \sin(\pi \mathbf{F}_j \boldsymbol{\delta}) \end{bmatrix}, \qquad (5)$$

$$\text{where } \mathbf{A}_j = h_a(\mathbf{z}_j), \ \mathbf{F}_j = h_f(\mathbf{z}_j). \qquad (6)$$

$\mathbf{A}_j \in \mathbb{R}^{2K}$ is an amplitude vector for a latent code j, $\mathbf{F}_j \in \mathbb{R}^{K \times 2}$ denotes a frequency matrix for a latent code j, and $\odot$ represents element-wise multiplication. We believe that the amplitude vector and the frequency matrix are extracted from the latent code j so as to represent $\mathbf{s}(\mathbf{x})$ as close as possible to original signals $\mathbf{I}^{\mathbf{HR}}(\mathbf{x})$. In this perspective, we understand that by observing pixels inside a receptive field (RF), LTE with the encoder ($h_\psi \circ E_\varphi$) estimates dominant frequencies and corresponding Fourier coefficients accurately. Here, the size of RF is decided by the encoder (E_φ). We visually demonstrate estimated frequencies and corresponding Fourier coefficients in Sec. 5.4.

In practice, to enrich the information in outputs of LTE, we apply the unfolding technique to $\mathbf{z}_j$ leading to a concatenation of the 3×3 nearest latent variables in $\hat{\mathbf{z}}_j$ [4]. This is implemented with trainable convolutional filters ($h_a(\cdot) : \mathbb{R}^{9C} \mapsto \mathbb{R}^{2K}, h_f(\cdot) : \mathbb{R}^{9C} \mapsto \mathbb{R}^{K \times 2}$).

Scale-dependent phase estimation Phase in Eq. (7) contains information about edge locations of features. For SR tasks, the location of edge changes within a small neighborhood in its HR domain when the scale factor changes. To address this issue, we redefine the estimating function as:

$$h_\psi(\hat{\mathbf{z}}_j, \boldsymbol{\delta}, \mathbf{c}) = \mathbf{A}_j \odot \begin{bmatrix} \cos(\pi(\mathbf{F}_j \boldsymbol{\delta} + h_p(\hat{\mathbf{c}}))) \\ \sin(\pi(\mathbf{F}_j \boldsymbol{\delta} + h_p(\hat{\mathbf{c}}))) \end{bmatrix} \qquad (7)$$

where $\mathbf{c}$ denotes the cell size. Inspired by the observation that MLPs with ReLUs are incapable of extrapolating unseen non-linear space [32], we use $\hat{\mathbf{c}} = \max(\mathbf{c}, \mathbf{c}_{tr})$, where $\mathbf{c}_{tr}$ denotes the minimum cell size during training.

LR skip connection A long skip connection in local implicit representation enriches high-frequency components in residuals and stabilizes convergence [12]. In the Fourier domain, LTE tends to predict frequencies located near a low-frequency region (DC). To prevent LTE from learning the DC only, we add upscaled LR. Thus, local implicit neural representation with the proposed LTE can be formulated as follows:

$$\hat{\mathbf{s}}(\mathbf{x}) = \mathbf{s}(\mathbf{x}, \mathbf{I}^{\mathbf{LR}}; \Theta, \psi) + \mathbf{I}_\uparrow^{\mathbf{LR}}(\mathbf{x}) \qquad (8)$$

4. Method

4.1. Network Detail

Our LTE-based arbitrary-scale SR network includes an encoder (E_φ), the LTE (a pink shaded area in Fig. 2), a

decoder (f_θ), and an LR skip connection. This section first describes a backbone structure (including encoder, decoder, and LR skip connection) and our LTE.

Backbone We use EDSR-baseline [15], RDN [35], and SwinIR [14] without their upsampling layers as an encoder (E_φ). Thus, an output of the encoder has the same width and height as an input LR image. We hypothesize that deep SISR networks [14, 15, 35] aid LTE in estimating crucial Fourier information by extracting features of natural images inside an RF. Our decoder (f_θ), shared by all images as [4], is a 4-layer MLP with a ReLU activation, and its hidden dimensions are 256. Lastly, we add a bilinear upscaled LR image to the decoder output as in Eq. (8). We expect that such a long skip connection provides DC offsets; thus, LTE is biased toward learning dominant frequencies and corresponding essential Fourier coefficients.

LTE Our LTE contains an amplitude estimator (h_a), a frequency estimator (h_f), a phase estimator (h_p), and sinusoidal activations. An amplitude and a frequency estimator are designed with 3x3 convolutional layers having 256 ($= 2K$) output channels, respectively, identical to a fully connected layer when feature maps are unfolded. The phase estimator is a single fully connected layer with hidden dimensions of 128. Note that an amplitude and a frequency estimator take the same feature map while the phase estimator takes cell as an input. Let us assume that a $r_\mathbf{x} \times r_\mathbf{y}$ local region in an HR domain shares amplitude and frequency information extracted from our LTE (h_ψ) as in Eq. (4). According to this, we upscale the extracted Fourier information using nearest-neighborhood interpolation. Then, a predicted phase is added to an inner product between the predicted frequency and local grid before passing it through the sinusoidal activation layer, as in Eq. (7). Finally, we multiply the predicted amplitude and sinusoidal activation output.

4.2. Training Strategy

We construct a minibatch with uniformly sampled scales from $\times 1 - \times 4$, dubbed *in-scale*, to teach the nature of bicubic degradation at various scales. Note that we evaluate our LTE for both *in-scale* and *out-of-scale*, which is an unseen scale (specifically $\times 6 - \times 30$), to verify the generalization ability of our network.

Let r ($= r_\mathbf{x} = r_\mathbf{y}$) be a scale factor randomly sampled from $\times 1 - \times 4$ and H, W be a height, a width of training patch, respectively. We first crop $rH \times rW$ patches from an HR image. When preparing training pairs, we randomly sample HW pixels from an HR patch for ground truth (GT) and downsample an HR patch by the scale factor r for an LR counterpart. When computing loss during training, we pick HW pixels from interpolation outputs to match the dimensions of prediction with GT.
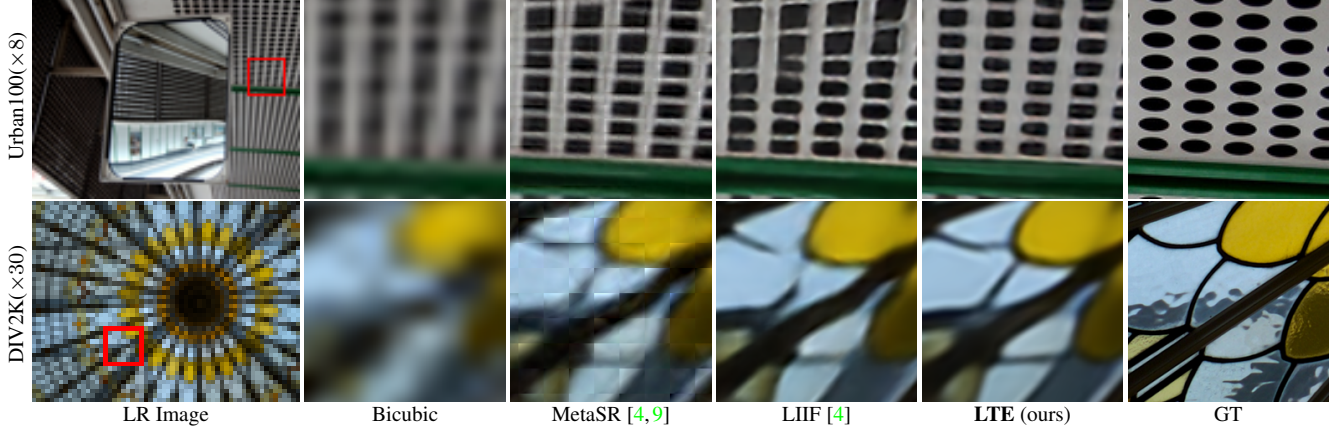
Figure 3. Qualitative comparison to other **arbitrary-scale SR**. RDN [35] is used as an encoder for all methods.

| Input | ×1.3 | ×1.7 | ×2.2 | ×2.8 | ×3.5 | ×4.3 |

Figure 4. Qualitative comparison with **non-integer scale factors** by LIIF [4] (top) and our LTE (bottom). RDN [35] is used as an encoder.

Figure 5. Qualitative comparison between various encoders with our LTE for ×**6 SR**.

Figure 6. Visual demonstration of SwinIR-LTE for ×**32 SR**.

5. Experiment

5.1. Training

Dataset We use a DIV2K dataset [1] of an NTIRE 2017 Challenge [30] for network training. For evaluation, we report peak signal-to-noise ratio (PSNR) results on the DIV2K validation set [1], Set5 [2], Set14 [33], B100 [17], and Urban100 [10].

Implementation detail We follow a prior implementation [15] and use 48×48 patches for inputs of our network.

For arbitrary-scale down-sampling during training time, we follow [4] and use bicubic resizing in Pytorch [22]. We use an L1 loss [15] and an Adam [13] method for optimization. When we train LTE with CNN-based encoders, such as EDSR-baseline [15] or RDN [35], networks are trained for 1000 epochs with batch size 16. The learning rate is initialized as 1e-4 and decayed by factor 0.5 at [200, 400, 600, 800]. For a transformer-based encoder, specifically SwinIR [14], a model is trained for 1000 epochs with batch size 32. The learning rate is initialized as 2e-4 and decayed by factor 0.5 at [500, 800, 900, 950].

Method	In-scale			Out-of-scale				
	×2	×3	×4	×6	×12	×18	×24	×30
Bicubic [15]	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR-baseline [15]	34.55	30.90	28.94	-	-	-	-	-
EDSR-baseline-MetaSR [4,9]	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
EDSR-baseline-LIIF [4]	34.67	30.96	29.00	26.75	23.71	22.17	21.18	20.48
EDSR-baseline-LTE (ours)	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
RDN-MetaSR [4,9]	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
RDN-LIIF [4]	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
RDN-LTE (ours)	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
SwinIR-MetaSR† [4,9]	35.15	31.40	29.33	26.94	23.80	22.26	21.26	20.54
SwinIR-LIIF† [4]	35.17	31.46	29.46	27.15	24.02	22.43	21.40	20.67
Swinir-LTE (ours)	35.24	31.50	29.51	27.20	24.09	22.50	21.47	20.73

Table 1. Quantitative comparison with state-of-the-art methods for **arbitrary-scale SR** on DIV2K validation set (PSNR (dB)). Red and blue colors indicate the best and the second-best performance, respectively. † indicates our implementation.

Method	Set5					Set14					B100					Urban100				
	In-scale			Out-of-scale		In-scale			Out-of-scale		In-scale			Out-of-scale		In-scale			Out-of-scale	
	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8	×2	×3	×4	×6	×8
RDN [35]	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 [4,9]	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 [4]	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 (ours)	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
SwinIR [14]	38.35	34.89	32.72	-	-	34.14	30.77	28.94	-	-	32.44	29.37	27.83	-	-	33.40	29.29	27.07	-	-
SwinIR-MetaSR† [4,9]	38.26	34.77	32.47	29.09	27.02	34.14	30.66	28.85	26.58	25.09	32.39	29.31	27.75	25.94	24.87	33.29	29.12	26.76	24.16	22.75
SwinIR-LIIF† [4]	38.28	34.87	32.73	29.46	27.36	34.14	30.75	28.98	26.82	25.34	32.39	29.34	27.84	26.07	25.01	33.36	29.33	27.15	24.59	23.14
SwinIR-LTE (ours)	38.33	34.89	32.81	29.50	27.35	34.25	30.80	29.06	26.86	25.42	32.44	29.39	27.86	26.09	25.03	33.50	29.41	27.24	24.62	23.17

Table 2. Quantitative comparison with state-of-the-art methods for **arbitrary-scale SR** on benchmark datasets (PSNR (dB)). Red and blue colors indicate the best and the second-best performance, respectively. † indicates our implementation.

5.2. Evaluation

Quantitative result Tab. 1 demonstrates a quantitative comparison between our LTE and existing arbitrary-scale SR methods, MetaSR [9], LIIF [4], on the DIV2K validation set. The top, middle, and bottom rows show results when EDSR-baseline [15], RDN [35], and SwinIR [14] are used as encoders. We notice that, regardless of a choice of an encoder, LTE achieves the best performance for all the scale factors, which indicates the effectiveness of the local texture.

In Tab. 2, we compare our LTE and RDN [35], SwinIR [14], MetaSR [9], LIIF [4] on benchmark datasets. Note that RDN and SwinIR [14] are trained with a specific scale; thus, it has significant benefits for *in-scale* [4]. However, including RDN and SwinIR, our LTE shows remarkable performance compared to other methods. The maximum PSNR gain is 0.15dB on Urban100 for ×3 within RDN.

Qualitative result Qualitative comparisons to other arbitrary-scale SR methods are provided in Fig. 3. For a fair comparison, MetaSR [9], LIIF [4], and our LTE are trained with RDN [35]. Note that MetaSR [9] follows [4]'s implementation to reconstruct an HR image for large-scale factors (> ×4). We see that MetaSR suffers from blocky artifacts, and LIIF shows structural distortion. In contrast, our LTE captures high-frequency details without any discontinuities.

Fig. 4 compares LIIF [4] and our LTE for text images with non-integer scale factors. We observe that our LTE is capable of restoring more clear edges of printed texts for all scale factors (particularly, 'n', 't', 's' in the first row and 'u', 'i', 'n' in the second row).

In Fig. 5, we show a qualitative comparison for ×6 SR. We remark that SwinIR [14] followed by LTE reconstructs the most visually pleasing image, faithful to the GT. It implies that LTE precisely extracts dominant frequencies and corresponding essential Fourier coefficients when jointly trained with a robust encoder. An empirical explanation using Fourier analysis is provided in Fig. 8 and Sec. 5.4.

As shown in Fig. 6, we visually demonstrate our LTE at an extremely large scale factor, specifically ×32. For the demonstration, we trained our LTE with SwinIR [14], and the width of an input image is 64px. We note that our LTE interpolates images with more sharp and natural edges compared to the bicubic method.

5.3. Ablation Study

In this section, we demonstrate the effect of each component in LTE. Our LTE consists of an amplitude estimator, a frequency estimator, a phase estimator, and an LR skip connection. To support the significance of each component, we retrain the following models with EDSR-baseline [15]. (-A): LTE without an amplitude estimator. (-F): LTE with a frequency estimator that estimates only 128 frequencies (not 256). (-P): LTE without a phase estimator. (-L): LTE without an LR skip connection.

Fig. 10 and Tab. 3 show the contributions of each LTE component on visual quality and performance. To verify the importance of each estimated frequency, we compare LTE to LTE (-F). We find that an amplitude estimator emphasizes dominant frequencies compared between LTE and LTE (-A). By comparing LTE and LTE (-P), disregarding a
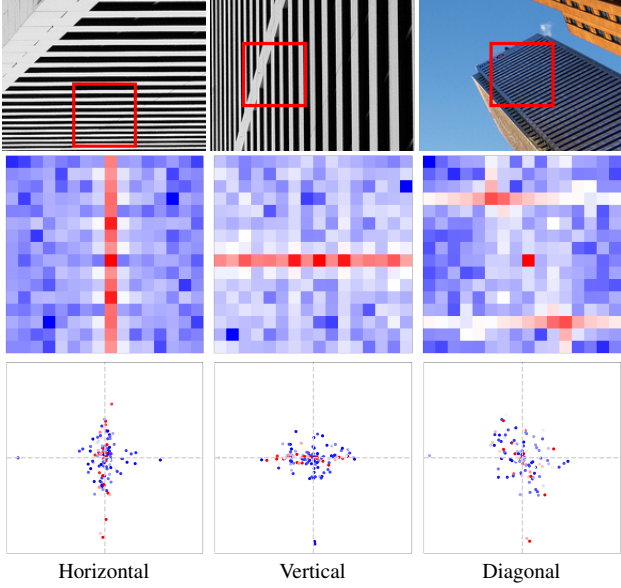
Figure 7. GT image (top), GT image spectrum (middle), and corresponding estimated Fourier space from LTE (bottom) of **various textures**. EDSR-baseline [15] is used as an encoder.

Horizontal   Vertical   Diagonal

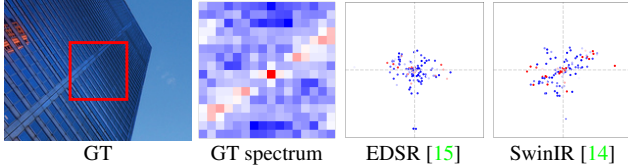GT   GT spectrum   EDSR [15]   SwinIR [14]

Figure 8. Estimated Fourier space with various encoders.

phase difference causes a significant performance drop. We see that an LR skip connection consistently enhances the quality of LTE when comparing LTE with LTE (-L).

5.4. Fourier Space

In this section (Figs. 7 to 9), we visualize extracted dominant frequencies with various textures. Furthermore, we investigate the contributions of each LTE component (specifically amplitude, frequency, phase, LR skip connection) through the lens of Fourier space.

Setup For visualization, we observe outputs of an amplitude estimator (h_a) and a frequency estimator (h_f). We first scatter dominant frequencies on 2D space and set color for each point with a magnitude. All scatter maps are defined on $[-1.5, 1.5]^2$, and the value range of each map is different from the other. In addition, 16-tap discrete Fourier transform (DFT) of GT images are provided to compare dominant frequencies of LTE and those of GT.

Image texture and Fourier space We choose three different textures: horizontal, vertical, diagonal textures, as illustrated in Fig. 7. Frequency maps from LTE in the bottom row are obtained from two-fold downsampled images. By comparing the middle row and the bottom row in Fig. 7, we observe that estimated dominant frequencies fol-
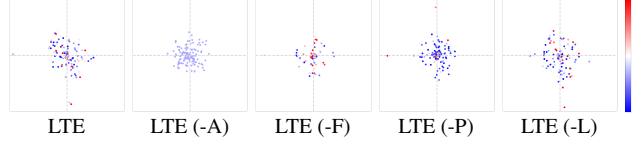

LTE   LTE (-A)   LTE (-F)   LTE (-P)   LTE (-L)

Figure 9. **Ablation study and corresponding Fourier space.** -F refers to reducing the number of estimated frequencies, and -A, -P, -L refers to removing amplitude estimator, phase estimator, LR skip connection, respectively. The diagonal texture in Fig. 7 is chosen for an ablation study, and EDSR-baseline [15] is used as an encoder.
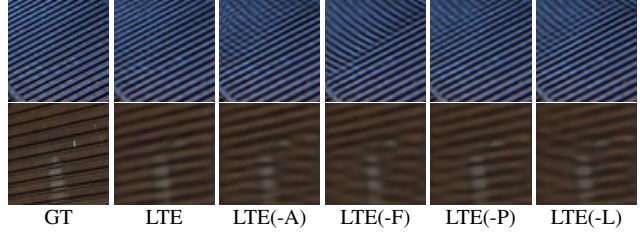

GT   LTE   LTE(-A)   LTE(-F)   LTE(-P)   LTE(-L)

Figure 10. **Qualitative ablation study of LTE on Urban100.** Definitions of -A, -F, -P, -L are shown in a caption of Fig. 9. EDSR-baseline [15] is used as an encoder.

	In-scale			Out-of-scale	
	×2	×3	×4	×6	×8
LTE	**33.72**	**30.37**	28.65	**26.50**	**24.99**
LTE (-A)	33.66	30.07	**28.66**	26.49	24.93
LTE (-F)	33.66	30.35	28.64	**26.50**	24.98
LTE (-P)	33.58	30.26	28.58	26.40	24.90
LTE (-L)	33.62	30.36	28.64	26.48	24.98

Table 3. **Quantitative ablation study of LTE on Set14.** Definitions of -A, -F, -P, -L are shown in a caption of Fig. 9. EDSR-baseline [15] is used as an encoder.

low the dominant frequencies of GT. It indicates that LTE obtains dominant frequencies and corresponding Fourier coefficients by observing pixels inside an RF. Note that the size of RF is determined by deep SR encoders (E_φ), such as EDSR-baseline [15], RDN [35], and SwinIR [14].

Encoder and Fourier space Tab. 2 and Fig. 5 demonstrate that LTE accomplishes better performance when SwinIR [14] is used as an encoder. Fig. 8 supports the observation by visualizing Fourier space. We remark that SwinIR-LTE captures dominant frequencies on a diagonal axis while EDSR-baseline-LTE estimates only low-frequency components. From this observation, LTE with a powerful encoder extracts precise dominant frequencies.

Ablation study on Fourier space Fig. 9 shows Fourier spaces of LTE when each component is missing. We choose a diagonal texture in Fig. 7 for an ablation study. Please see a caption of Fig. 9 for definitions of -A, -F, -P, -L. LTE (-A) considers coefficients of all frequencies as equal since Fourier coefficients are not given from LTE. Therefore, LTE (-A) focuses on learning low-frequency content. LTE (-P) is incapable of estimating frequencies positioned on a diagonal axis. We suppose that without a scale-dependent phase estimation, LTE (-P) detects only scale-independent

information: Image signal is compactly supported by low-frequency regions. We validate that the deficiency of dominant frequencies fails to learn high-frequency details by comparing LTE and LTE (-F). On comparing LTE and LTE (-L), LTE (-L) is poor in capturing dominant frequencies.

6. Discussion

Advantage of LTE over DFT In DFT, Fourier information is represented with a linear combination of image intensity. However, the DFT of an LR image with aliasing (middle) is limited in capturing dominant frequencies. In contrast, LTE with a deep neural encoder (bottom), which is a multi-chain of linear combinations and non-linear activations, is capable of estimating accurate Fourier information for an HR image (top).

Figure 11. DFT and LTE

Effect of aliasing Fig. 8 shows that SwinIR-LTE is capable of estimating dominant frequencies of natural images. In addition, the middle row in Fig. 12 demonstrate that SwinIR-LTE extracts essential Fourier information under mild aliasing. However, such capability of SwinIR-LTE is limited when an LR image has severe aliasing. From Fig. 12, we observe that dominant frequencies (bottom right) are inconsistent with a GT spectrum (top right) when harsh aliasing artifacts occur in an LR image (bottom left). We can resolve such limitations by extending the size of an encoder's RF, followed by increased computation and memory costs. Cost-effective architectures achieving robust performance even under severe aliasing will be investigated in future work.

Figure 12. Aliasing.

Gibbs phenomenon When representing continuous signals with a finite sum of Fourier basis, function overshoots at discontinuities. Such observation is referred to in the literature as the *Gibbs phenomenon* or ringing artifacts in 2D images. From Fig. 13, we notice that LTE might cause overshoot at large scale factors, e.g., $\times 12$. Further investigation of smoothing algorithms to alleviate such an issue is a promising direction for future work.

Figure 13. Gibbs phenomenon ($\times 12$).

Computation time In practice, SR applications require short computation time. Moreover, reconstructing high-quality images, such as DIV2K, consumes extensive memory during evaluation. Tab. 4 compares the computa-
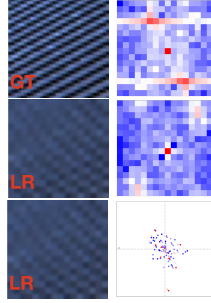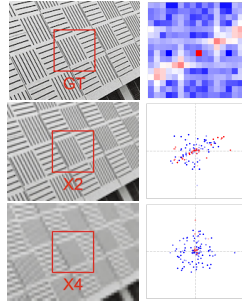
#Eval/Query	Method	# Params.	Mem. (GB)	Time (ms)
9216 (96 × 96)	MetaSR [9]	1.7M	1.9	3462
	LIIF [4]	1.6M	1.9	4559
	LTE (ours)	1.7M	2.3	2912
1.6M (1248 × 1248)	MetaSR [9]	1.7M	*OOM*	-
	LIIF [4]	1.6M	11.4	873
	LTE (ours)	1.7M	10.2	925
	LTE+ (ours)	1.7M	7.1	483

Table 4. Memory consumption (GB) and computation time (ms) comparison to other **arbitrary-scale SR** methods for an ×2 SR task. We use a 624×624 sized input and EDSR-baseline [15] as an encoder. *OOM* denotes an *out-of-memory*, and our GPU memory was deficient in evaluating MetaSR [9].

tion time of our LTE to other arbitrary-scale SR methods for both cases: memory-limited (top rows) and memory-consuming (bottom rows) on NVIDIA RTX 3090 24GB. To evaluate an HR image under a memory-limited condition, we compute 96×96 output pixels per query [4]. From the top rows of Tab. 4, we observe that our LTE takes the shortest computation time while increasing memory usage. LTE has 4-layer MLP for querying, two convolution layers for estimators, and LIIF has 5-layer MLP only for querying. When querying evaluation points only once, estimators become more dominant than querying, resulting in more computation time, as in the bottom rows of Tab. 4. To overcome such a limitation, we design an LTE+, which utilizes 1×1 convolution instead of a shared MLP for decoder implementation. Since 1×1 convolution has a GPU-friendly data structure, our LTE+ takes a shorter computation time and consumes less memory compared to previous works when all output pixels are queried at once.

7. Conclusion

In this paper, we proposed the Local Texture Estimator (LTE) to overcome the spectral bias problem of an implicit neural function. Our LTE-based arbitrary-scale SR method consists of three components: (1) Deep SR encoder (2) LTE (3) Implicit representation function. Firstly, a deep SR encoder extracts feature maps whose height and width are the same as an LR image. Then, LTE takes feature maps from the encoder and estimates dominant frequencies with corresponding Fourier coefficients for natural images. Scale-dependent phase and LR skip connection are further provided to allow LTE to be biased in learning high-frequency textures. Finally, the implicit function reconstructs an image in arbitrary resolution using estimated Fourier information. We showed that our LTE-based neural function outperforms other arbitrary-scale SR methods in performance and visual quality with the shortest computation time.

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 (CVPR) Workshops*, July 2017. 5

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012. 5

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-Trained Image Processing Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12299–12310, June 2021. 1, 2

[4] 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 (CVPR)*, pages 8628–8638, June 2021. 1, 2, 3, 4, 5, 6, 8

[5] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-Order Attention Network for Single Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[7] David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[8] K. Hornik, M. Stinchcombe, and H. White. Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.*, 2(5):359–366, July 1989. 2

[9] 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 (CVPR)*, June 2019. 1, 2, 5, 6, 8

[10] 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 (CVPR)*, June 2015. 5

[11] Chiyu "Max" Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Niessner, and Thomas Funkhouser. Local Implicit Grid Representations for 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3

[12] 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 (CVPR)*, June 2016. 4

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[14] 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 (ICCV) Workshops*, pages 1833–1844, October 2021. 1, 2, 4, 5, 6, 7

[15] 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 (CVPR) Workshops*, July 2017. 1, 2, 4, 5, 6, 7, 8

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021. 2

[17] D. Martin, C. Fowlkes, D. Tal, and J. 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 vol.2, 2001. 5

[18] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image Super-Resolution With Non-Local Sparse Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3517–3526, June 2021. 1, 2

[19] 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 (CVPR)*, June 2019. 1, 2

[20] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 1, 2, 3

[21] 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 (CVPR)*, June 2019. 1, 2

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In

H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 5

[23] 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 Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019. 1, 2, 3

[24] Wenzhe Shi, Jose Caballero, Ferenc Huszar, 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 (CVPR)*, June 2016. 1, 2

[25] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020. 1, 2

[26] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 2

[27] 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 (CVPR)*, pages 7782–7791, June 2021. 2

[28] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned Initializations for Optimizing Coordinate-Based Neural Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2846–2855, June 2021. 2

[29] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020. 1, 2, 3

[30] 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 (CVPR) Workshops*, July 2017. 5

[31] 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 (ICCV)*, pages 4801–4810, October 2021. 2

[32] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4

[33] Roman Zeyde, Michael Elad, and Matan Protter. On Single Image Scale-Up Using Sparse-Representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker, editors, *Curves and Surfaces*, pages 711–730, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 5

[34] 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)*, September 2018. 1, 2

[35] 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 (CVPR)*, June 2018. 1, 2, 4, 5, 6, 7