Exploring the Low-Pass Filtering Behavior in Image Super-Resolution

Haoyu Deng [1] **Zijing Xu** [1] **Yule Duan** [1] **Xiao Wu** [1] **Wenjie Shu** [1] **Liang-Jian Deng** [1]

Abstract

Deep neural networks for image super-resolution (ISR) have shown significant advantages over traditional approaches like the interpolation. However, they are often criticized as 'black boxes' compared to traditional approaches with solid mathematical foundations. In this paper, we attempt to interpret the behavior of deep neural networks in ISR using theories from the field of signal processing. First, we report an intriguing phenomenon, referred to as 'the sinc phenomenon.' It occurs when an impulse input is fed to a neural network. Then, building on this observation, we propose a method named Hybrid Response Analysis (HyRA) to analyze the behavior of neural networks in ISR tasks. Specifically, HyRA decomposes a neural network into a parallel connection of a linear system and a non-linear system and demonstrates that the linear system functions as a low-pass filter while the non-linear system injects high-frequency information. Finally, to quantify the injected high-frequency information, we introduce a metric for image-to-image tasks called Frequency Spectrum Distribution Similarity (FSDS). FSDS reflects the distribution similarity of different frequency components and can capture nuances that traditional metrics may overlook. Code, videos and raw experimental results for this paper can be found in: https://github.com/RisingEntropy/LPFInISR.

Please refer to Appx. A for notation conventions.

1. Introduction

The goal of image super-resolution (ISR) is to reconstruct low-resolution (LR) images into high-resolution (HR) images through various techniques. In recent years, with advances in deep learning, growing ISR methods using neural networks are proposed, bringing the development of ISR into a new level. While impressive results persistently arise, the mechanism under ISR networks remain largely unexplored, leading to criticism that they are considered black boxes. In comparison, traditional methods, such as interpolation or filtering, have strong interpretability. Despite the principles of traditional methods and neural networks are different, we can still attempt to explain the behavior of ISR networks using theories from traditional methods. In this paper, following this line of thought, we successfully utilize theories from the field of signal processing techniques to explain the performance of neural networks in the ISR task.

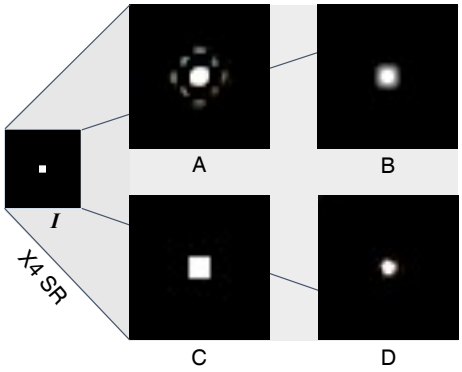

Figure 1. I is an image in which only the central pixel is 1 and the other pixels are 0. What would the result look like if image I is super-resolved using a neural network, A, B, C, or D? Surprisingly, the answer is A. We name this phenomenon as **the sinc phenomenon**. In this paper, we give a possible explanation for this phenomenon.

The target of the ISR task is to upsample a two-dimensional signal. In traditional signal processing theory (Oppenheim & Schafer, 2009; Oppenheim et al., 1996), a feasible method for upsample involves restoring a discrete low-sampling-rate signal to a continuous signal using a low-pass filter, and then sampling the continuous signal at a higher rate to obtain a high-sampling-rate signal. An intriguing aspect of this process is that when we try to upsample a Dirac δ signal, we will finally get a sinc signal since the sinc signal is the time-domain waveform of a low-pass filter, (for

[1]University of Electronic Science and Technology of China. Correspondence to: Liang-Jian Deng <liangjian.deng@uestc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

details about this, please refer to Sec. 3). Given this, we can conjecture: if neural networks exhibit similar behavior, then when attempt to super-resolve a Dirac δ signal, the resultant outcome would also be a sinc signal. As shown in Fig. 1, we indeed observe this phenomenon, and we name it as '*the sinc phenomenon*'. This phenomenon establishes a connection between traditional signal processing theory and the interpretability of neural networks, thus helping us form a deeper understanding of ISR networks.

Building upon the sinc phenomenon, we further propose a method named HyRA[1], which stands for Hybrid Response Analysis. HyRA considers the neural network as a parallel combination of a linear system and a non-linear system with a zero impulse response. It further indicates that this linear system functions as a low-pass filter, while the non-linear system utilizes the learned prior knowledge to inject high-frequency information. By employing HyRA, we can analyze performance bottlenecks in neural networks, discerning whether the issue lies in inadequate preservation of low-frequency components or insufficient injection of high-frequency components. This analysis facilitates the proposal of targeted improvements for enhanced adaptability.

Given that the non-linear component is injecting high-frequency information, there is a pressing need for a metric to quantitatively describe the extent of the injected high frequencies. Previous metrics, like PSNR, SSIM (Wang et al., 2004) and LPIPS (Zhang et al., 2018a), have not approached the evaluation of images from a frequency perspective. Therefore, we propose the frequency spectrum distribution similarity (FSDS), a metric that evaluates image quality based on the power distribution in the frequency spectrum.

In summary, our contribution can be concluded as:

- We report an intriguing phenomenon: the impulse responses of image super-resolution (ISR) networks are sinc functions, representing the temporal waveform of a low-pass filter. We name it the 'sinc phenomenon'. This observation helps to establish a connection between signal processing theory and neural networks. Moreover, we find that for a network, the more similar the impulse response is to the sinc function, the better performance it produces.

- In order to further explain the performance of neural networks in the ISR task through this phenomenon, we introduce HyRA. HyRA considers the neural network as a parallel combination of a linear system and a non-linear system with a zero impulse response. It points out that the linear system operates as a low-pass filter, while the non-linear system injects high-frequency information.

- To quantitatively describe the injection of high frequencies, we introduce the FSDS metric. FSDS measures image quality using frequency spectrum produced by FFT and can reflect high-frequency distortions that previous metrics fail to capture.

2. Related works

2.1. Super Resolution Using Neural Networks

Recent review articles in ISR include fixed-scale super-resolution (Yang et al., 2019) and arbitrary-scale super-resolution review (Liu et al., 2023). There are various architectures of mainstream ISR backbone networks, including CNN-style backbones (Ahn et al., 2018; Hui et al., 2019; Lim et al., 2017; Zhang et al., 2018b;c), transformer-style backbones (Liang et al., 2021; Wang et al., 2023) and GAN-style backbone networks (Wang et al., 2018), etc. Based on these backbones, researchers have proposed quantitative modules with various functions. For example, ArbSR (Wang et al., 2021) can expand a fixed-scale super-resolution network to an arbitrary-scale ISR network, LTE (Lee & Jin, 2022) can enhance local textures, etc. What worth mentioning is that LIIF (Chen et al., 2021) introduces implicit neural representation into ISR for the first time, bringing a new approach for ISR. This paper mainly focuses on approaches that utilize CNN-style or transformer-style backbones. Except for network architectures, numerous datasets have been proposed to facilitate further research. Commonly used datasets for ISR includes Set5 (Bevilacqua et al., 2012), Urban100 (Huang et al., 2015), Flickr2K (Young et al., 2014), SCI1K (Yang et al., 2021), DIV2K (Agustsson & Timofte, 2017), etc. We evaluate the effectiveness of our proposed FSDS metric on DIV2K dataset. The large size of the DIV2K dataset contributes to increased reliability in our conclusions.

2.2. Explaining the Behavior of Neural Networks

Despite neural networks are often criticized as 'black boxes,' predecessors have made remarkable efforts to mitigate this situation. Various previous researches have proposed plenty of methods to analyze the behavior of neural networks. Since Sundararajan et al. (Sundararajan et al., 2017) introduce the integrated gradients (IG) for attribution in classification tasks, numerous researchers have expanded this method to various domains, broadening the scope of attribution beyond classification tasks. Based on IG, Gu & Dong (Gu & Dong, 2021) propose LAM to analyze the impact of the local patch on the entire ISR outcome. However, such a method requires manually determined hyper-parameters and baselines, thus introducing subjectivity. Several notable analysis methods utilizing the Fourier transform have been explored

[1]Pronounce as [haɪ'rɑː]

in the literature (Xu, 2018; 2020; Xu et al., 2019; Zhang et al., 2019). Notably, Xu et al. (Xu et al., 2019) propose the Frequency-Principle, claiming its relevance to both convolutional neural networks (CNNs) and fully-connected deep neural networks. According to their proposition, these networks inherently adhere to the Frequency-Principle, wherein training data is systematically acquired in a sequential manner, progressing from low to high frequency. Unlike previous approaches these approaches, HyRA distinguishes itself by employing impulse response to probe the potential mechanisms of deep neural networks in the context of the ISR task.

3. Preliminaries

Appx. B.1-Appx. B.3 provide a brief overview of signal processing concepts for readers who are not familiar with it. Appx. B.1 introduces the concepts of signals and systems, along with the computation of responses in Linear Time-Invariant (LTI) systems. Appx. B.2 covers the processes of signal sampling and reconstruction. Appx. B.3 delves into the phenomenon of spectrum aliasing, a factor contributing to the ill-posed nature of the ISR task. And we will introduce applying low-pass filter for ISR here.

We can employ signal recovery methods to achieve image super-resolution (ISR). Initially, we conceptualize an image as a series of impulse trains in a two-dimensional continuous space, with varying densities representing different resolutions. Then, for the low-resolution image, we begin by implementing low-pass filtering, following the procedure outlined in Appx. B.2, to obtain the continuous image I^{cont}, This process can be mathematically described as:

$$I^{\text{cont}}_{x,y} = sinc^{\omega}_{x,y} * I^{\text{LR}}_{x,y}, \tag{1}$$

where $*$ denotes convolution, $I^{LR}_{x,y}$ is the low resolution image with variant x, y and $I^{\text{cont}}_{x,y}$ is the continuous signal. $sinc^{\omega}_{x,y}$ is a two-dimensional sinc function with parameter ω[2], whose frequency spectrum is an ideal low-pass filter with a passband of $0 \sim \omega$. Subsequently, we sample the 'conceptually continuous signal' at an elevated sampling rate to acquire a more densely populated two-dimensional sequence of impulse trains, i.e., an image with higher resolution denoted as I^{SR}:

$$I^{SR}_{x,y} = I^{\text{cont}}_{x,y} \cdot s^{\Delta X, \Delta Y}_{x,y}. \tag{2}$$

In the equation, $s^{\Delta X, \Delta Y}_{x,y}$ denotes the two-dimensional impulse trains with intervals of ΔX in x axis and ΔY in y axis.

In fact, commonly used interpolation kernels for ISR, such as nearest-neighbor interpolation, linear interpolation, cubic interpolation, etc., can be seen as approximations of the sinc function considering a balance between computational complexity and effectiveness, as illustrated in Fig. 2. Taking into account the similarity of these interpolation kernels, in this paper, we collectively refer to these parameter-free methods as low-pass filter-based super-resolution methods.

[2]Please refer to Tab. 3 for Fourier transform pairs

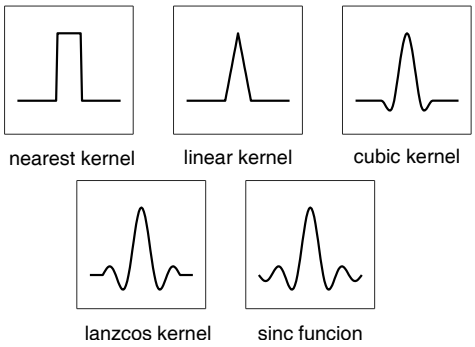

Figure 2. Various interpolation kernels for ISR. They can all be seen as an approximation of sinc function.

4. Method

4.1. Hybrid Response Analysis (HyRA)

In this section, we describe the proposed Hybrid Response Analysis (HyRA), which treats the neural network as a combination of a linear system and a non-linear system. Through the impulse response, we can calculate a linear time invariant (LTI) system's output from any input using the convolution operation (see Appx. B.1). However, since neural networks are nonlinear systems, we cannot apply convolution to analyze them. To further explore the network features, we need to split it into a linear system and a non-linear system, i.e., HyRA. The core concept HyRA is illustrated in Fig. 3. We denote an ISR network as $N(I)$, where I is the input image. $N(I)$ is a non-linear system that can be expressed as the sum of a linear system and a non-linear system:

$$N(I) = H(I) + G(I). \tag{3}$$

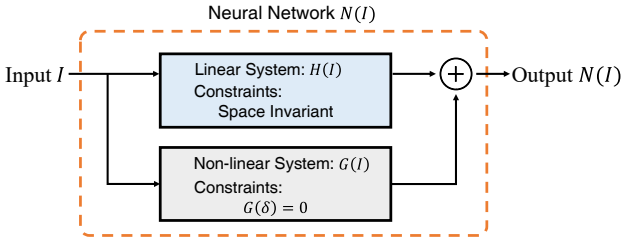

Figure 3. Conceptual diagram of HyRA's core idea.

In the equation, $H(I)$ represents a linear system, and $G(I)$ represents a non-linear system. Without constraints, such a representation is meaningless because $H(I)$ can be arbitrarily chosen, leading to an infinite variety of representations with the same form but different meanings. To give meaning to this representation, we introduce a constraint: the impulse response of $G(I)$ is zero. With this constraint, both $H(I)$ and $G(I)$ can be uniquely determined. Lemma 4.1 demonstrates that under this constraint, $N(I)$ can still be expressed in the form of Eq. 3. This straightforward method is the essence of HyRA.

For the ISR task, there is a distinctive property known as a 'spatially invariant system' (Miller et al., 1992) associated with it. Consider the definition of time-invariant systems as mentioned in Appx. B.1, we can naturally extend the concept of in-variance from one-dimensional to two-dimensional space and the definition of spatially invariant systems is: when the input is $I_{x,y}$, the output is $G(I_{x,y}) = O(x, y)$; when the input becomes $I' = I_{x-x_0,y-y_0}$, the output should be $G(I') = O(x - x_0, y - y_0)$. For convolution based architectures, we can easily prove its spatial invariance (see the proof below). For transformer-based architectures, we can still use experiments to prove the spatial invariance (see Fig. 15).

Proof. A convolution operation can be defined as:

$$Conv_{i,j} = \sum_{p,q} I_{i-p,j-1} K_{p,q}.$$

Then, the shifting operation can be defined as:

$$\mathrm{Sh}(i, j) \rightarrow (i + k, j + l).$$

Combine these two, we then have:

$$\begin{aligned} Conv_{\mathrm{Sh}(i,j)} &= Conv_{i+k,j+l} \\ &= \sum_{p,q} I_{i+k-p,j+l-q} K_{p,q} \\ &= \mathrm{Sh}(\sum I_{i-p,j-q} K_{p,q}) \\ &= \mathrm{Sh}(Conv_{i,j}). \end{aligned}$$

This is the invariance of a single convolution layer, and still holds for more layers. □

According to HyRA, when we input a Dirac δ signal to the neural network, we can get the impulse response of the linear system (please recall Appx. B.1), denoted as $H(\delta)$. For any input I, the response of the linear space invariant system can be obtained by convolving the input with the obtained impulse response, which can be expressed as:

$$H(I) = I * H(\delta), \tag{4}$$

where $*$ means the convolution operation. Although the response of the non-linear component cannot be directly computed, if we obtain the final output of the neural network, the non-linear part can be deduced by subtracting the response of the linear component from the final output, namely the non-linear response can be computed as:

$$\begin{aligned} G(I) &= N(I) - H(I) \\ &= N(I) - I * H(\delta). \end{aligned} \tag{5}$$

Lemma 4.1. *A neural network $N(I)$ can be expressed as a combination of a linear system $H(I)$ and a non-linear system with an impulse response of zero, i.e., $N(I) = H(I) + G(I)$, where $G(\delta) = 0$. Here, δ represents the Dirac delta function.*

Proof.
1) When $G(\delta) = 0$, the conclusion holds.

2) When $G(\delta) \neq 0$, Let $H_1(I) = H(I) + G(\delta) * I$ and $G_1(I) = G(I) - G(\delta) * I$. In this case, $H_1(I)$ remains a linear system and $G'(I)$ remains a non-linear system. The equation $N(I) = H_1(I) + G_1(I)$ holds, and it satisfies $G_1(\delta) = 0$. □

4.1.1. $H(I)$ IS A LOW-PASS FILTER

In Sec. 3, we mention that a simple low-pass filter achieves ISR functionality. Do neural networks possess low-pass filters internally? If this hypothesis is valid, according to the principle of HyRA, when we input a Dirac δ signal into the neural network $N(I)$, the output should be the impulse response of the low-pass filter, i.e., the sinc function (please recall Appx. B.1 and Tab. 3). In the experiment section (Sec. 5.2), we conduct tests on three mainstream ISR backbones and some derived methods. We find that their impulse responses are sinc functions[3]. Now, with both the impulse response and spatial invariance property, we can compute the response of the linear system $H(I)$ to any input through convolution:

$$\begin{aligned} H(I)_{x,y} &= I_{x,y} * H(\delta) \\ &= \iint_{(\tau,u)\in\mathbb{R}^2} I_{\tau,u} H(\delta)_{x-\tau,y-u} \mathrm{d}\tau \mathrm{d}u. \end{aligned} \tag{6}$$

In a practical scenario, when dealing with a two-dimensional impulse array represented by I, the integration process can be effectively substituted with summation, incorporating appropriate padding. Despite the convolution operator in PyTorch (Paszke et al., 2019) being inherently a correlation operator, the symmetric nature of the sinc function allows for its seamless utilization within such an operator. We

[3] Strictly speaking, it is a windowed sinc function. Regarding the windowing operation, please refer to Appx. E

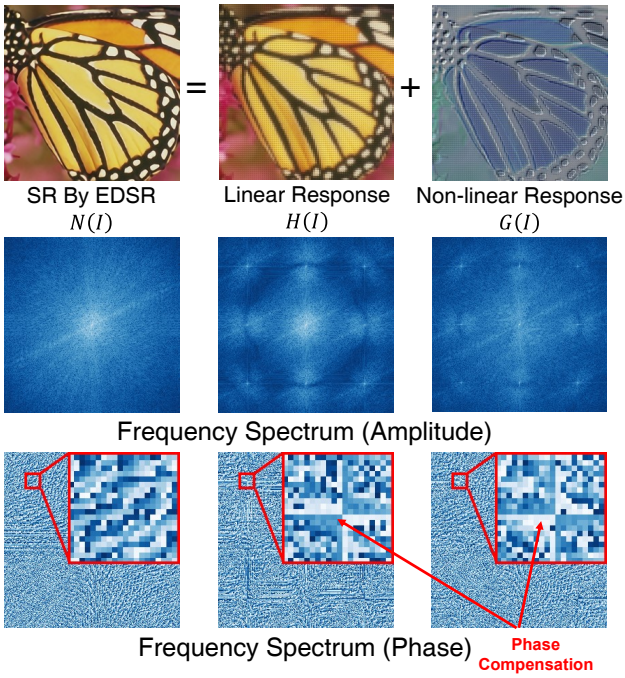

Figure 4. Top row: a super-resolved image by (Lim et al., 2017) can be viewed as the summation of a linear response obtained by convolving impulse response with the input and the non-linear response gained by subtracting linear-part from the ISR result. Second row: the corresponding frequency spectrum amplitude of the top row. Third row: the corresponding frequency spectrum phase of the top row. The phase compensation indicates that the non-linear part is compensating distortion.

present a toy example in Fig. 4 in which we compute the response of the linear component of the EDSR network (Lim et al., 2017) during ISR. Observing the experimental results, we notice that the linear function $H(I)$ essentially achieves super-resolution, but there are some issues: edge blurring and the presence of grid-like distortions.

The edge is blurred because the low-pass filter removes some high-frequency details. In the frequency spectrum, it is manifested as a relatively small range of diffusion of the central bright spot towards the surroundings. This implies that the image has more low-frequency components and fewer high-frequency components. Such an outcome is the inevitable consequence of applying the low-pass filter.

When computing the response of the linear system, we first perform zero-interpolation on the low-resolution image to achieve the target spatial size. This operation leads to periodic extension in the frequency spectrum[4]. Since this low-pass filter is not a complete ideal filter, but an ideal filter truncated by a certain window function, its filtering

[4]Please refer to Appx. F in the Appendix for details about the periodic extension.

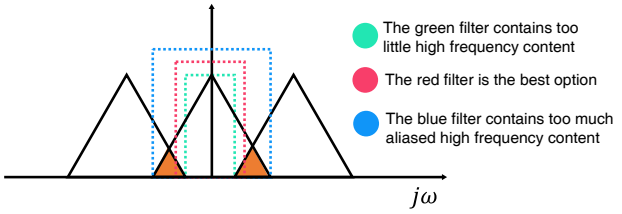

Figure 5. An illustration of how the passband width of a low-pass filter affects its ISR results. A too wide passband or a too narrow passband can result in a decline in performance.

performance is weakened by the window function. The weakened filter cannot completely eliminate the extended spectrum, meaning the attenuation in the stopband is insufficient, as referred to in signal processing, thus causing such gird-like distortions.

In summary, the linear system $H(I)$ (the low-pass filter approximated by the neural network) can achieve super-resolution functionality, but it is not perfect. On one hand, the low-pass filter determines that the image is blurred, lacking high frequencies. On the other hand, the filter is windowed, leading to a weakened filtering performance and resulting in grid-like distortions. These issues will be compensated for by the nonlinear system $G(I)$.

4.1.2. $G(I)$ Injects high-frequency information

Though a low-pass filter can achieve ISR (please refer to Sec. 3), its performance can never surpass a well-trained neural network. The outcome of a low-pass filter varies with respect to the passband width, as depicted in Fig. 5. However, information outside the passband will be completely wiped out, causing an observable detail loss in high-frequency components. On the contrary, the non-linear part of neural networks is able to inject information in high-frequency domain based on learned or structural priors. Moreover, it can compensate the grid-like distortions brought by the windowed low-pass filter. Together with the linear part, neural networks function as the superset of low-pass filter, retaining both high and low frequency information.

We compute the non-linear response and its frequency spectrum of the neural network using the proposed HyRA paradigm. In the toy example presented in Fig. 4, it can be noticed that the response of the non-linear component exhibits sharper edges. Compared with the frequency spectrum of the ISR results, the central bright spot in the response of $G(I)$ spreads to a larger range, indicating that more power is distributed into the high-frequency domain. Almost all the components of the high-frequency part in the final ISR result are contributed by the non-linear component.

As mentioned in Sec. 4.1.1, the non-linear component also plays a crucial role in compensating for the distortion intro-

duced by $H(I)$. Examining the response of $G(I)$, we note that it also exhibits grid-like distortions, matching those in the response of $H(I)$. This allows for the cancellation of the grid-like distortions, achieving the final goal of ISR. As shown in Fig. 4, upon observing the frequency spectrum, bright spots corresponding to the amplitude spectrum of $H(I)$ exist in all four corners of the amplitude spectrum of $G(I)$. However, the phase spectrum of $G(I)$ is in compensation of the phase spectrum of $H(I)$, indicating that the grid-like distortion is 'erased' here.

In summary, the non-linear component $G(I)$ serves to inject high-frequency details learned during training to compensate for the loss of high frequencies introduced by the low-pass filter. Simultaneously, it addresses distortions arising from the imperfect performance of the low-pass filter.

4.2. Frequency Spectrum Distribution Similarity (FSDS)

In this section, we introduce the FSDS metric to quantitatively describe the so called the 'injected high frequencies' as discussed in Sec. 4.1.2.

4.2.1. Motivation and Method

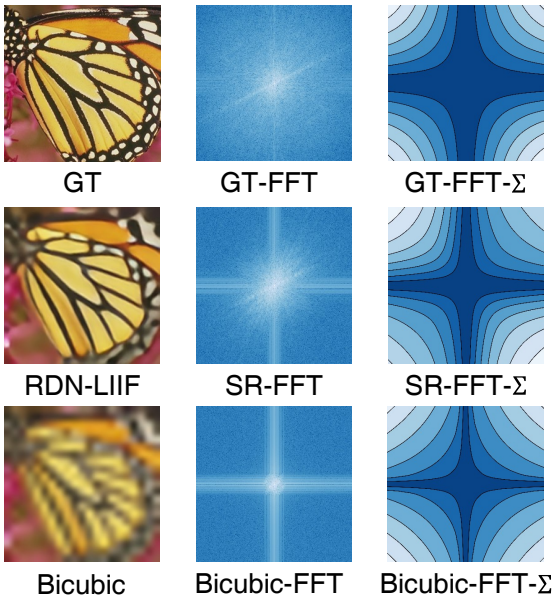

Figure 6. X-FFT-Σ denotes the integrated frequency spectrum, the integration path is from origin to infinty in every quadrant. Columns 1 and 2 in the figure respectively show that the differences in the results of different ISR methods can be reflected in the frequency spectrum. Column 3 presents the integral of the spectrum from low to high frequencies in a contour plot. The distribution of contours visually represents the distinct distribution of different frequency components.

Since we need to measure the components of injected high frequencies, we must delve into the issue from a frequency spectrum perspective. However, commonly used metrics such as PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018a) do not measure the quality of an image from a spectral perspective.

Additionally, we've noted that the frequency domain distribution in the ISR field can significantly impact downstream applications (Xu et al., 2020; Yu et al., 2023). Consequently, we propose that evaluating the ISR effectiveness of a network requires a thorough assessment of its performance in the frequency spectrum. This involves examining the similarity in frequency spectrum between the low-resolution image and the high-resolution image. The Frequency Spectrum Distribution Similarity (FSDS) metric integrates the power distribution maps of the spectrum for both images. The difference is then calculated to generate an error map, and the total sum of its absolute values is computed.

For an image $I^{\mathrm{HR}}_{x,y}$, to minimize the impact of the data input range on the results, we normalize the input data and then perform a two-dimensional Fourier transform to obtain $I^{\mathrm{HR}}_{j\omega_1,j\omega_2}$, which can be mathematically described as:

$$I^{\mathrm{HR}}_{j\omega_1,j\omega_2} = \mathcal{F}\left[\frac{I^{\mathrm{HR}} - E(I^{\mathrm{HR}})}{\sigma(I^{\mathrm{HR}})}\right], \tag{7}$$

where $E(I^{\mathrm{HR}})$ and $\sigma(I^{\mathrm{HR}})$ are the mean value and variance of I^{HR} respectively. Similarly, we perform a Fourier transform on the ISR image to obtain $I^{\mathrm{SR}}_{j\omega_1,j\omega_2}$. It is worth noting that unlike other metrics, such as PSNR and SSIM(Wang et al., 2004), which do not incorporate normalization, FSDS is specifically designed to accentuate numerical variations due to its emphasis on numerical changes rather than absolute numerical values. Then, the complex integration of the two spectrum is performed, providing the power distribution map D^{HR}, which is defined as:

$$D^{\mathrm{HR}} = \iint_{(\omega_1,\omega_2)\in\mathbb{R}^2} I^{\mathrm{HR}} \mathrm{d}\omega_1 \mathrm{d}\omega_2. \tag{8}$$

Similarly, we can obtain D^{SR}. Subsequently, the difference between D^{HR} and D^{SR} is calculated, providing a difference map D^{diff} of their power distribution:

$$D^{\mathrm{diff}} = D^{\mathrm{HR}} - D^{\mathrm{SR}}. \tag{9}$$

Finally, we define the frequency spectrum distribution similarity (FSDS) as:

$$\mathrm{FSDS} = -10\log_{10}\frac{\iint_{(\omega_1,\omega_2)\in\mathbb{R}^2} |D^{\mathrm{diff}}|^2 \mathrm{d}\omega_1 \mathrm{d}\omega_2}{\iint_{(\omega_1,\omega_2)\in\mathbb{R}^2} |D^{\mathrm{HR}}|^2 \mathrm{d}\omega_1 \mathrm{d}\omega_2}, \tag{10}$$

where $|\cdot|$ represents taking the magnitude of a complex number. Considering a more concise description of a larger dynamic range, logarithm is taken. A larger FSDS value indicates that the two images are closer, thereby suggesting better ISR results.

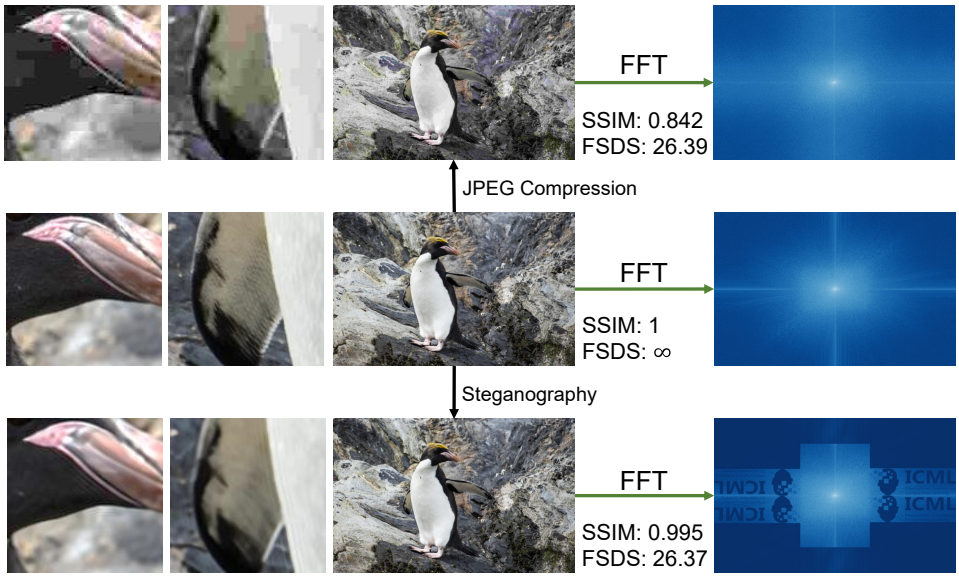

Figure 7. A comparison of SSIM and FSDS in JPEG compression and steganography. As can be seen, SSIM fails to reflect distortion brought by steganography, while FSDS captures both cases of distortion.

4.2.2. The merits of FSDS

Previous image evaluation metrics, such as PSNR, SSIM (Wang et al., 2004), have focused on statistical or structural features of images, but no work has evaluated images from the perspective of their frequency spectrum. The spectrum is the concentrated expression of components with different changing rates in a signal or image. It is crucial for capturing details, eliminating noise, and comprehensively understanding image features. In image processing, spectrum analysis provides a more accurate evaluation, particularly playing a key role in applications sensitive to details. Due to the nature of Fourier transformation, which involves every pixel of the image in the computation, it encompasses not only information such as signal-to-noise ratio and structural similarity but also the overall similarity of the entire image. Therefore, evaluating image quality from the perspective of the spectrum is highly reasonable and necessary. Our FSDS metric can reflect distribution differences by employing a paradigm of integrating first in the frequency spectrum and then comparing. In other words, FSDS not only reflects the signal-to-noise ratio captured by the PSNR metric and the structural similarity indicated by the SSIM metric, but also captures features that these two metrics cannot represent. In the next paragraph, we will use two toy examples to demonstrate the rationale and advantages of FSDS.

From Fig. 6, it can be observed that images obtained by different ISR methods have different proportions of high-frequency components (the center of the spectrum figure represents low frequency, while higher frequencies extend outward). After integration, this is reflected in the varying widths of the dark cross-shaped patterns in the center. A narrower width indicates a higher proportion of low-frequency components in the spectrum, and vice versa. Existing methods may not effectively capture the loss of high-frequency components with low power in the frequency spectrum. Performing information steganography in the frequency spectrum can effectively highlight this aspect. As shown in Fig. 7, we embed some content in the frequency spectrum of the image. Such steganography causes our FSDS metric to drop to 26.37dB while the SSIM metric remains in a high level of 0.995. we can observe that after applying specific steganography to the spectrum of an image, the image exhibits some blurring and oscillation. Such oscillations are actually the Gibbs phenomenon, a typical oscillation phenomenon caused by the loss of high-frequency information. Meanwhile, when we apply JEPG compression to the image[5], when FSDS drops to 26.39dB, SSIM together drops to 0.842. **This toy example demonstrates that there indeed exists some feature SSIM cannot reflect while that can be reflected by FSDS**.

In summary, previous methods may not effectively reflect the situation in the image frequency spectrum, while our proposed FSDS metric can sensitively detect distortions in the frequency spectrum.

5. Experiments

Due to the page limitation, we can only present three of the most crucial experiments in this section, namely: 1) the

[5]In this example, the compression quality is set to 10.

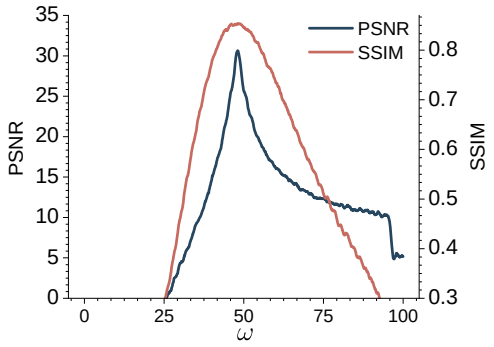

Figure 8. The ISR performance using a low-pass filter shows variations with the cutoff frequency ω. This figure illustrates the results obtained from the $\times 2$ ISR task conducted on the DIV2K dataset. To enhance the clarity of the visualization, the curve has been smoothed using a moving average with a window length of 10.

relationship between the low-pass filter passband width and ISR performance; 2) various network impulse responses; 3) a comparison of FSDS metrics with PSNR, SSIM and LPIPS (Zhang et al., 2018a) metrics on the DIV2K dataset. For more experiments, please refer to Appx. C.

5.1. Experiment on Low-pass Filtering Super-resolution Performance

In Sec. 4.1.2, we mention that a vanilla low-pass filter can achieve ISR, we now present an experiment on the relationship between the low-pass filter passband width and ISR performance. As shown in Fig. 8, we utilized various low-pass filters to perform $\times 2$ ISR on the validation set from of DIV2K dataset. Subsequently, we evaluated the ISR results using the PSNR and SSIM metrics. When $\omega = 48$, PSNR reaches its maximum value of 31.40. When $\omega = 45.8$, SSIM reaches its maximum value of 0.87. We assert that, in terms of neural network performance, for $\times 2$ ISR, the PSNR should not fall below 31.40, and the SSIM should not be lower than 0.87. Otherwise, it can be considered that the neural network may not effectively capture both low-frequency and high-frequency information.

5.2. Experiment on Impulse Response

We select several mainstream backbones and their derivatives commonly used for the ISR task (Chen et al., 2021; Hu et al., 2019; Lee & Jin, 2022; Liang et al., 2021; Lim et al., 2017; Song et al., 2023; Wei & Zhang, 2023; Zhang et al., 2018b;c) and conduct impulse response tests. The experimental results are compared with the sinc function and depicted in Fig. 9. The input image is an 11×11 image where only the pixel at position $(5, 5)$ is white (the values for all three channels at this position are 255, with indices starting from 0), and the rest of the image is black (with values of 0). According to Tab. 3, it can be observed that as the ISR factor increases, the central peak of the output sinc function becomes wider and more pronounced. To balance visual saliency and the maximum ISR factor achievable by certain networks, we opted for a 4x ISR factor. Observing the experimental results, we can notice that regardless of the neural network structure used for ISR, whether it's a CNN or a transformer, the impulse response exhibits some degree of similarity to the two-dimensional sinc function. This similarity is particularly pronounced in networks like RDN (Zhang et al., 2018c) and RCAN (Zhang et al., 2018b). Despite some distortion in comparison to the sinc function, EDSR (Lim et al., 2017), EQSR (Wang et al., 2023), and their derivatives still exhibit significant features of the sinc function, including the central bright spot and elongated bright patches in the cardinal directions. From Tab. 1, we observe that networks exhibiting superior performance tend to generate impulse responses that closely resemble the sinc function. **This observation suggests that preserving low-frequency information more effectively can also enhance performance. However, few previous works has focus on low-frequency, giving us a new idea for furture ISR networks.**

5.3. Experiment on FSDS Metric

We conducted tests on the validation set of the DIV2K dataset (Agustsson & Timofte, 2017) using several methods (Chen et al., 2021; Lee & Jin, 2022; Liang et al., 2021; Lim et al., 2017; Song et al., 2023; Wei & Zhang, 2023; Yang et al., 2021; Zhang et al., 2018c), depicted in Tab. 1. The evaluation metrics include PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018a), and our FSDS. All tests are performed using code and weights available in open-source official repositories. For all methods, we conduct experiment for $\times 2$ to $\times 4$. For methods that support arbitrary-scale ISR, we test for $\times 6$ and $\times 12$ as well. In the $\times 2$ to $\times 4$ range, GRLBase(Li et al., 2023) consistently achieves the best performance across PSNR, SSIM and LPIPS metrics, and FSDS shows that SwinIR (Liang et al., 2021) achieves the best performance. For $\times 6$ and $\times 12$, RDN-LTE (Lee & Jin, 2022) exhibits the best PSNR and SSIM metrics, while RDN-LIIF performs best on LPIPS and the FSDS metric.

From Sec. 4.2, we claim that previous metrics are not sensitive to high-frequency information, while FSDS does. This can be proven by Tab. 1. In the case of slight high-frequency loss, such as on scales $\times$ 2 to $\times$ 4, FSDS responds differently compared to previous metrics. In cases that suffer from severe high-frequency loss, such as on $\times$ 6 and $\times$ 12 scales, FSDS shows consistency with previous metrics. This is because when high-frequency loss is slight, previous metrics fail to reflect such high-frequency loss and while the loss becomes more severe, they start to capture such loss.

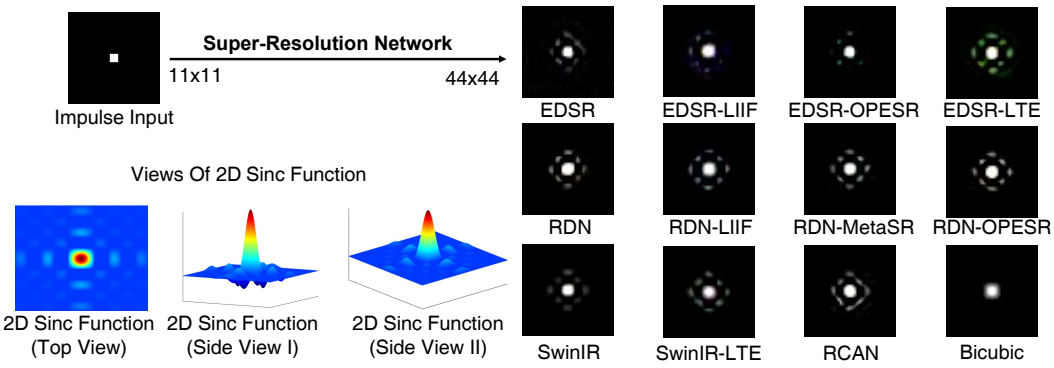

Figure 9. Comparison of impulse responses and the sinc function for several mainstream backbone networks and their derivatives. The impulse response of the bicubic interpolation result is presented as a reference.

Method	PSNR					SSIM					LPIPS					FSDS (Ours)				
	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12
EDSR(Lim et al., 2017)	34.63^{12}	30.95^{14}	28.87^{16}	-	-	0.937^{12}	0.874^{13}	0.816^{16}	-	-	0.042^{12}	0.101^{14}	0.155^{16}	-	-	39.21^{15}	34.15^{10}	31.38^{7}	-	-
EDSR-LIIF(Lim et al., 2017)	34.55^{14}	30.92^{15}	28.98^{15}	26.76^{4}	23.75^{4}	0.937^{15}	0.874^{14}	0.819^{14}	0.741^{4}	0.633^{4}	0.043^{15}	0.100^{13}	0.153^{13}	0.243^{4}	0.428^{2}	39.37^{13}	34.53^{6}	31.32^{9}	28.45^{4}	23.15^{3}
EDSR-OPESR(Lim et al., 2017)	34.34^{16}	30.96^{12}	29.04^{12}	-	-	0.936^{16}	0.875^{11}	0.820^{13}	-	-	0.043^{13}	0.100^{11}	0.153^{14}	-	-	39.80^{6}	34.63^{5}	31.29^{11}	-	-
EDSR-SRNO(Lim et al., 2017)	34.72^{9}	31.05^{10}	29.13^{10}	26.90^{3}	23.87^{3}	0.939^{9}	0.876^{10}	0.822^{11}	0.746^{3}	0.638^{3}	0.041^{8}	0.098^{10}	0.149^{10}	0.241^{3}	0.437^{4}	39.53^{11}	34.53^{7}	31.45^{6}	28.46^{3}	22.78^{4}
EDSR-LTE(Lim et al., 2017)	34.61^{13}	30.97^{11}	29.03^{14}	-	-	0.937^{14}	0.874^{12}	0.820^{12}	-	-	0.043^{14}	0.100^{12}	0.152^{12}	-	-	39.29^{14}	34.33^{8}	31.30^{10}	-	-
RDN(Zhang et al., 2018c)	34.69^{10}	30.58^{16}	29.12^{11}	-	-	0.938^{10}	0.867^{16}	0.823^{10}	-	-	0.041^{9}	0.106^{16}	0.150^{11}	-	-	40.02^{3}	32.95^{16}	31.65^{4}	-	-
RDN-LIIF(Zhang et al., 2018c)	34.86^{5}	31.21^{8}	29.26^{9}	26.99^{2}	23.93^{2}	0.939^{8}	0.879^{8}	0.826^{9}	0.749^{2}	0.639^{2}	0.041^{10}	0.096^{8}	0.147^{8}	0.231^{1}	0.406^{1}	39.69^{9}	34.83^{3}	31.83^{3}	28.89^{1}	23.78^{1}
RDN-OPESR(Zhang et al., 2018c)	34.52^{15}	31.19^{9}	29.28^{8}	-	-	0.938^{11}	0.879^{9}	0.826^{8}	-	-	0.042^{11}	0.096^{7}	0.148^{9}	-	-	40.19^{2}	34.96^{2}	31.48^{5}	-	-
RDN-LTE(Zhang et al., 2018c)	34.91^{7}	31.26^{7}	29.31^{7}	27.05^{1}	23.99^{1}	0.939^{7}	0.879^{7}	0.827^{7}	0.750^{1}	0.641^{1}	0.041^{7}	0.095^{6}	0.144^{6}	0.233^{2}	0.431^{3}	39.82^{5}	34.75^{4}	31.84^{2}	28.65^{2}	23.35^{2}
SwinIR-classical(Liang et al., 2021)	35.34^{5}	31.64^{5}	29.63^{4}	-	-	0.943^{5}	0.885^{5}	0.835^{5}	-	-	0.038^{5}	0.092^{4}	0.140^{5}	-	-	40.37^{1}	35.13^{1}	32.37^{1}	-	-
ITSRN(Yang et al., 2021)	32.67^{17}	30.49^{17}	28.73^{17}	26.64^{5}	23.72^{5}	0.922^{17}	0.866^{17}	0.813^{17}	0.736^{5}	0.630^{5}	0.052^{17}	0.113^{17}	0.167^{17}	0.271^{5}	0.469^{5}	31.25^{18}	26.18^{18}	25.88^{18}	25.62^{5}	21.57^{5}
HAT-S(Chen et al., 2023)	35.46^{2}	31.72^{3}	29.72^{3}	-	-	0.944^{2}	0.887^{3}	0.837^{3}	-	-	0.038^{2}	0.092^{3}	0.139^{3}	-	-	39.78^{7}	33.80^{13}	31.06^{14}	-	-
HAT(Chen et al., 2023)	35.46^{2}	31.77^{2}	29.75^{2}	-	-	0.944^{2}	0.887^{2}	0.837^{2}	-	-	0.038^{2}	0.090^{2}	0.138^{2}	-	-	39.78^{7}	33.91^{11}	31.20^{12}	-	-
HDSRNet(Tian et al., 2024)	34.64^{11}	30.95^{13}	29.04^{13}	-	-	0.937^{13}	0.873^{15}	0.819^{15}	-	-	0.043^{16}	0.103^{15}	0.154^{15}	-	-	39.46^{12}	34.20^{9}	31.33^{8}	-	-
GRLBase(Li et al., 2023)	35.66^{1}	31.93^{1}	29.91^{1}	-	-	0.945^{1}	0.889^{1}	0.841^{1}	-	-	0.037^{1}	0.089^{1}	0.135^{1}	-	-	39.99^{4}	33.84^{12}	31.12^{13}	-	-
GRLSmall(Li et al., 2023)	35.39^{4}	31.65^{4}	29.63^{5}	-	-	0.943^{4}	0.886^{4}	0.835^{4}	-	-	0.038^{4}	0.092^{5}	0.140^{4}	-	-	39.54^{10}	33.68^{14}	31.03^{15}	-	-
GRLTiny(Li et al., 2023)	35.17^{6}	31.41^{6}	29.40^{6}	-	-	0.942^{6}	0.882^{6}	0.830^{6}	-	-	0.039^{6}	0.096^{9}	0.146^{7}	-	-	39.20^{16}	33.20^{15}	30.56^{16}	-	-
Bicubic	31.04^{18}	28.25^{18}	26.69^{18}	24.87^{6}	22.34^{6}	0.893^{18}	0.813^{18}	0.752^{18}	0.675^{6}	0.587^{6}	0.096^{18}	0.191^{18}	0.291^{18}	0.439^{6}	0.613^{6}	32.79^{17}	28.90^{17}	26.57^{17}	23.45^{6}	18.74^{6}

Table 1. Comparison of PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018a) and FSDS metrics for different methods on the DIV2K dataset (Agustsson & Timofte, 2017). Items with the highest and the second-highest mean values are highlighted in red and blue, respectively. The gray superscripts are the order of each method.

This observation shows the necessity of applying the FSDS metrics to assess image quality objectively.

5.4. Some Exceptions to Impulse Responses

Figure 10. The impulse response of SwinIR-Real (Liang et al., 2021) and ESRGAN (Wang et al., 2018) is not an obvious sinc function.

We observe that not all impulse response of networks is 'sinc' function, as shown in Fig. 10. SwinIR-Real (Liang et al., 2021) and ESRGAN (Wang et al., 2018) are trained using adversarial loss, while methods in Fig. 9 uses loss like ℓ_1 or ℓ_2 loss. Therefore, we believe the 'sinc' impulse response is related to the loss function.

6. Conclusion

In this paper, we report an intriguing observation. i.e., the sinc phenomenon, which reveals that the impulse response of ISR networks act as low-pass filters. Building on this observation, we introduce a novel approach called Hybrid Response Analysis (HyRA) to explore the hidden behavior of ISR networks. HyRA treats a neural network as a combination of a linear system and a non-linear system with a zero impulse response. The linear system functions as a low-pass filter, while the non-linear system utilizes prior knowledge to inject high-frequency details. To assess the neural network's information recovery across the frequency spectrum, we propose the Frequency Spectrum Distribution Similarity (FSDS) metric. FSDS uncovers properties overlooked by previous metrics, and experiments validate the rationality and necessity of it.

Acknowledgements

We appreciate anonymous reviewers for their previous suggestions to help this paper better. Moreover, we would like to express our sincere gratitude to Ruijie Zhu (rzhu48@ucsc.edu) for his generous support in GPUs. Without his support, it is hard for us to do experiments using full-scale DIV2K dataset. This work is supported by NSFC (12271083).

Impact Statement

This paper presents work whose goal is to advance the interpretability of neural networks in Image super-resolution. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agustsson, E. and Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 126–135, 2017.

Ahn, N., Kang, B., and Sohn, K.-A. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 252–268, 2018.

Bevilacqua, M., Roumy, A., Guillemot, C., and Alberi-Morel, M. L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

Chen, X., Wang, X., Zhou, J., Qiao, Y., and Dong, C. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22367–22377, 2023.

Chen, Y., Liu, S., and Wang, X. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8628–8638, 2021.

Gu, J. and Dong, C. Interpreting super-resolution networks with local attribution maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9199–9208, 2021.

Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., and Sun, J. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1575–1584, 2019.

Huang, J.-B., Singh, A., and Ahuja, N. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5197–5206, 2015.

Hui, Z., Gao, X., Yang, Y., and Wang, X. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th acm international conference on multimedia*, pp. 2024–2032, 2019.

Lee, J. and Jin, K. H. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1929–1938, 2022.

Li, Y., Fan, Y., Xiang, X., Demandolx, D., Ranjan, R., Timofte, R., and Van Gool, L. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18278–18289, 2023.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

Liu, H., Li, Z., Shang, F., Liu, Y., Wan, L., Feng, W., and Timofte, R. Arbitrary-scale super-resolution via deep learning: A comprehensive survey. *Information Fusion*, pp. 102015, 2023.

Miller, J. W., Farison, J. B., and Shin, Y. Spatially invariant image sequences. *IEEE Transactions on Image Processing*, 1(2):148–161, 1992.

Nyquist, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 1928.

Oppenheim, A. V. and Schafer, R. W. *Discrete-Time Signal Processing*. Prentice Hall Press, USA, 2009.

Oppenheim, A. V., Willsky, A. S., and Nawab, S. H. *Signals & Systems (2nd Ed.)*. Prentice-Hall, Inc., USA, 1996.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Song, G., Sun, Q., Zhang, L., Su, R., Shi, J., and He, Y. Ope-sr: Orthogonal position encoding for designing a parameter-free upsampling module in arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pp. 10009–10020, 2023.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Tian, C., Zhang, X., Ren, J., Zuo, W., Zhang, Y., and Lin, C.-W. A heterogeneous dynamic convolutional neural network for image super-resolution. *arXiv preprint arXiv:2402.15704*, 2024.

Wang, L., Wang, Y., Lin, Z., Yang, J., An, W., and Guo, Y. Learning a single network for scale-arbitrary super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4801–4810, 2021.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Wang, X., Chen, X., Ni, B., Wang, H., Tong, Z., and Liu, Y. Deep arbitrary-scale image super-resolution via scale-equivariance pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1786–1795, 2023.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wei, M. and Zhang, X. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18247–18256, 2023.

Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.-K., and Ren, F. Learning in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1740–1749, 2020.

Xu, Z. J. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018.

Xu, Z.-Q. J. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020.

Xu, Z.-Q. J., Zhang, Y., and Xiao, Y. Training behavior of deep neural network in frequency domain. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26*, pp. 264–274. Springer, 2019.

Yang, J., Shen, S., Yue, H., and Li, K. Implicit transformer network for screen content image continuous super-resolution. *Advances in Neural Information Processing Systems*, 34:13304–13315, 2021.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21 (12):3106–3121, 2019.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Yu, Y., She, K., Liu, J., Cai, X., Shi, K., and Kwon, O. A super-resolution network for medical imaging via transformation analysis of wavelet multi-resolution. *Neural Networks*, 166:162–173, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018a.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 286–301, 2018b.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018c.

Zhang, Y., Xu, Z.-Q. J., Luo, T., and Ma, Z. Explicitizing an implicit bias of the frequency principle in two-layer neural networks. *arXiv preprint arXiv:1905.10264*, 2019.

A. Notation Conventions

Symbols	
j	Imaginary number unit
$*$	Convolution operator
$I_{x,y}^{\text{comment}}$	2-D signal with variant x, y
$I_{j\omega_1,j\omega_2}^{\text{comment}}$	Fourier transform of $I_{x,y}$
$x(t)$	1-D signal with variant t
$X(j\omega)$	Fourier transform of $x(t)$, $j\omega$ is a notation, ω is the variant
$x[n]$	Discrete signal with index n
$X[k]$	DFT of $x[n]$
$\mathcal{F}[x(t)]$	Fourier transform operator, $X(j\omega) = \mathcal{F}[x(t)]$
$\mathcal{F}^{-1}[X(j\omega)]$	Inverse Fourier transform, $x(t) = \mathcal{F}^{-1}[X(j\omega)]$
Signals	
$\delta(t)$	Dirac δ function
$sinc_\omega(t)$	The sinc funtion with parameter ω, $sinc_\omega(t) = \frac{sin(\omega t)}{\pi t}$. The sinc function is the time-domain waveform of an ideal low-pass filter.
$sinc_{x,y}^{\omega}$	2-D sinc function with parameter ω, $sinc_{x,y}^{\omega} = \frac{sin(\omega x)}{\pi x} \cdot \frac{sin(\omega y)}{\pi y}$
$s_{\Delta T}(t)$	1-D sample signal with a sample interval of ΔT, $s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$

Table 2. Notation Conventions

B. Signal Processing Theories

We briefly introduce some related concepts and methods used in this paper in this section.

B.1. System and Response

The word 'system' has many meanings and interpretations. This paper views a system as a process in which input signals are transformed by the system or cause the system to respond in some way, resulting in other signals as output (Oppenheim et al., 1996). Systems can be divided into linear systems and nonlinear systems according to their mathematical properties. A linear system refers to a system with such a property: the response of the system to the input $x_1(t)$, $x_2(t)$ is $y_1(t)$, $y_2(t)$ respectively, then when the input is $x_1(t) + x_2(t)$, the response of the system is $y_1(t) + y_2(t)$.

Systems can also be divided into time-variant ones and time-invariant ones according to their temporal properties. A time-invariant system refers to that the properties of the system do not change with time, that is, the system has the same impulse response at any time. It satisfies such a relationship: when the input is $x(t)$, the output is $y(t)$, and when the input is $x(t - t_0)$, the output is $y(t - t_0)$.

A system with both linear and time-invariant properties is a linear time-invariant (LTI) system. For an LTI system, we can use 'impulse response' to uniquely describe it: systems with the same impulse response are the same system, vice versa. The impulse response $h(t)$ is defined as the output of the system when the input signal is $\delta(t)$ (Dirac delta function). The response of a linear system to an arbitrary input signal can be computed through the convolution operation of its impulse response and the input signal, namely:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau)h(t - \tau)d\tau. \tag{11}$$

In the equation, $*$ is the convolution operator, $y(t)$ is the system output and $x(t)$ is the input signal. When we apply Fourier transform to the impulse response $h(t)$, then we can obtain the transfer function $H(j\omega)$ of the system. The transfer function describes the frequency domain waveform of the impulse response. According to the convolution theorem, the response of a linear system can also be obtained by multiplying the Fourier transform of the input signal by the transfer function of the system and then performing the inverse Fourier transform. In summary, given the impulse response of an LTI system, we can calculate the system's response to any output.

B.2. Signal Sampling and Recovery

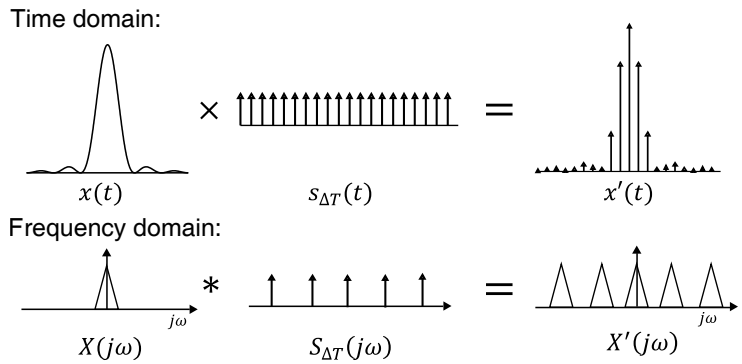

Figure 11. Time-domain to frequency-domain waveform variation of the continuous signal sampling process. The sampling function $s_{\delta T}$ is an impulse train sequence with an interval of T, and $S_{\delta T}(t)$ is its frequency domain waveform, which is also an impulse train sequence. Sampling a signal causes duplication in the frequency domain.

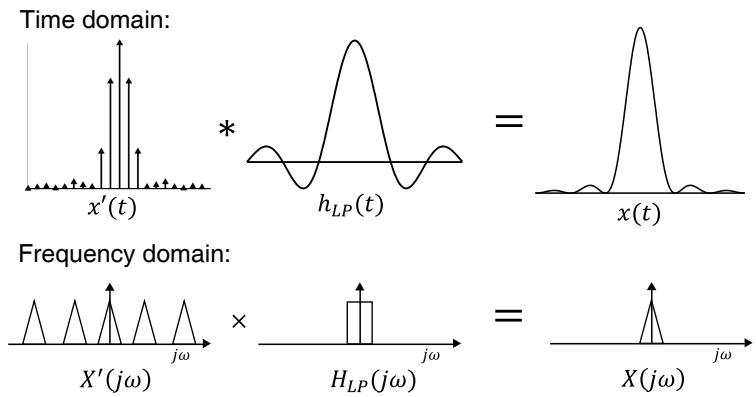

Figure 12. Time-domain to frequency-domain waveform variation in the process of sampling signal recovery. $h_{LP}(t)$ is the time-domain impulse response of a low-pass filter, and $H_{LP}(j\omega)$ is its frequency-domain waveform.

Signal sampling and sample recovery are very common operations, and in this section, we will briefly analyze this process from the perspective of both the time-domain and frequency-domain. The upper part of Fig. 11 shows the time domain waveform variation of signal sampling process, and the lower part shows the frequency domain waveform variation of signal sampling process. To sample a continuous signal, the sampling process can be regarded as the multiplication of the original signal $x(t)$and an impulse train signal $s_{\Delta T}(t)$. It can be described as:

$$\begin{cases} x'(t) = x(t) \cdot s_{\Delta T}(t), \\ s_{\Delta T}(t) = \sum\limits_{n=-\infty}^{\infty} \delta(t - nT), \end{cases} \tag{12}$$

where T denotes the sampling interval, $x'(t)$ denotes the sampled signal. According to the convolution theorem, the frequency domain change of the sampling process can be described in the following way:

$$\begin{aligned} X'(j\omega) &= X(j\omega) * S_{\delta T}(j\omega) \\ &= \sum_{k=-\infty}^{\infty} X[j(\omega - k\frac{2\pi}{T})]. \end{aligned} \tag{13}$$

That is, the sampling process is reflected in the frequency spectrum as a periodic extension of the frequency spectrum of the original signal $x(t)$.

Fig. 12 shows the time domain and frequency domain waveform variation during the recovery process. For sampling recovery, in order to restore the sampled signal $x'(t)$ to the original signal $x(t)$, from the perspective of frequency domain, a low-pass filter is all we needed, that is, convolving the sampled signal with a low-pass filter $h_{LP}(t)$. This process can be expressed as:

$$\begin{aligned} x(t) &= x'(t) * h_{LP}(t) \\ &= x'(t) * sinc_{\omega}(t), \end{aligned} \tag{14}$$

where in the equation, $h_{LP}(t) = sinc_{\omega_0}(t) = \frac{sin(\omega_0 t)}{\pi t}$ is the time domain response of the ideal low-pass filter, and its frequency-domain waveform $H_{LP}(j\omega)$ is a rectangular window.

B.3. Spectrum Aliasing

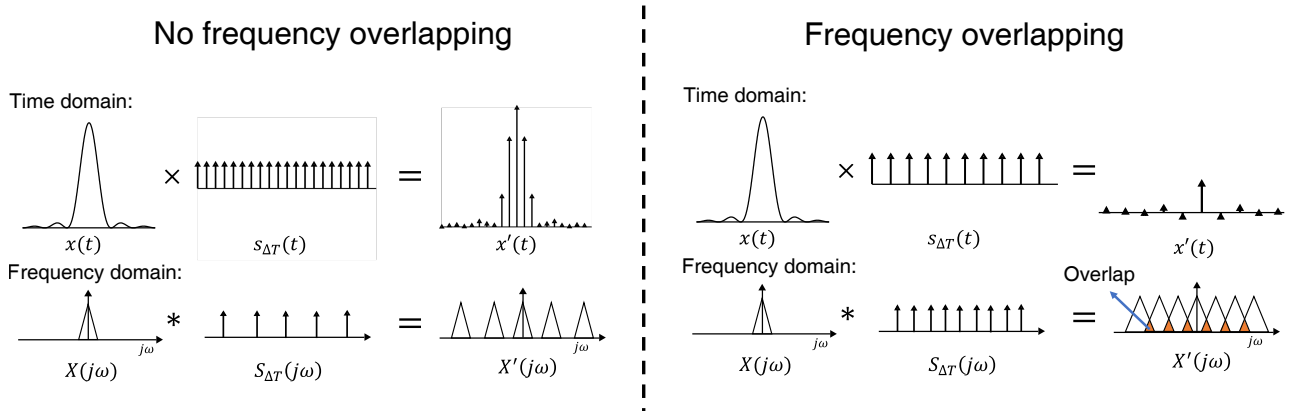

Figure 13. The illustration of spectrum aliasing. On the left, there is no aliasing as the sampling rate is sufficiently high. On the right, aliasing occurs due to an insufficient sampling rate.

Spectrum aliasing is a manifestation of information loss. Fig. 13 depicts the time-domain and frequency-domain scenarios of no frequency overlapping and frequency overlapping, respectively. When the sampling rate is lower than the Nyquist sampling rate[6] (Nyquist, 1928). When the sampling rate is below the Nyquist sampling rate, the approach mentioned in Appx. B.2 cannot completely restore the original signal $x(t)$. From Tab. 3, we can see that for the sample signal $s_{\Delta T}(t)$, the larger T is, the sparser its time domain impulse train gets, while in the frequency spectrum the impulse trains gets denser. When the impulse trains in the frequency domain become sufficiently dense, and the spectrum of the original signal is periodically extended, overlapping occurs, preventing the complete recovery of the original signal. In ISR tasks, spectrum aliasing is manifested when restoring a low-resolution image to a high-resolution image, resulting in the loss of high-frequency information such as details and textures.

C. Extra Experiments

C.1. Linear and Non-linear Responses

In Fig. 14, we present the linear and nonlinear responses of various ISR networks along with their corresponding spectrums. From the figure, it is evident that different networks exhibit varying filtering effects in their linear components. EDSR demonstrates a pronounced removal of high-frequency components, and compared to other methods, it exhibits the smallest area of brightness diffusion around the central bright spot in its spectrum. From the nonlinear responses, it can be observed that the nonlinear components of the networks are all involved in supplementing high-frequency information and correcting distortions.

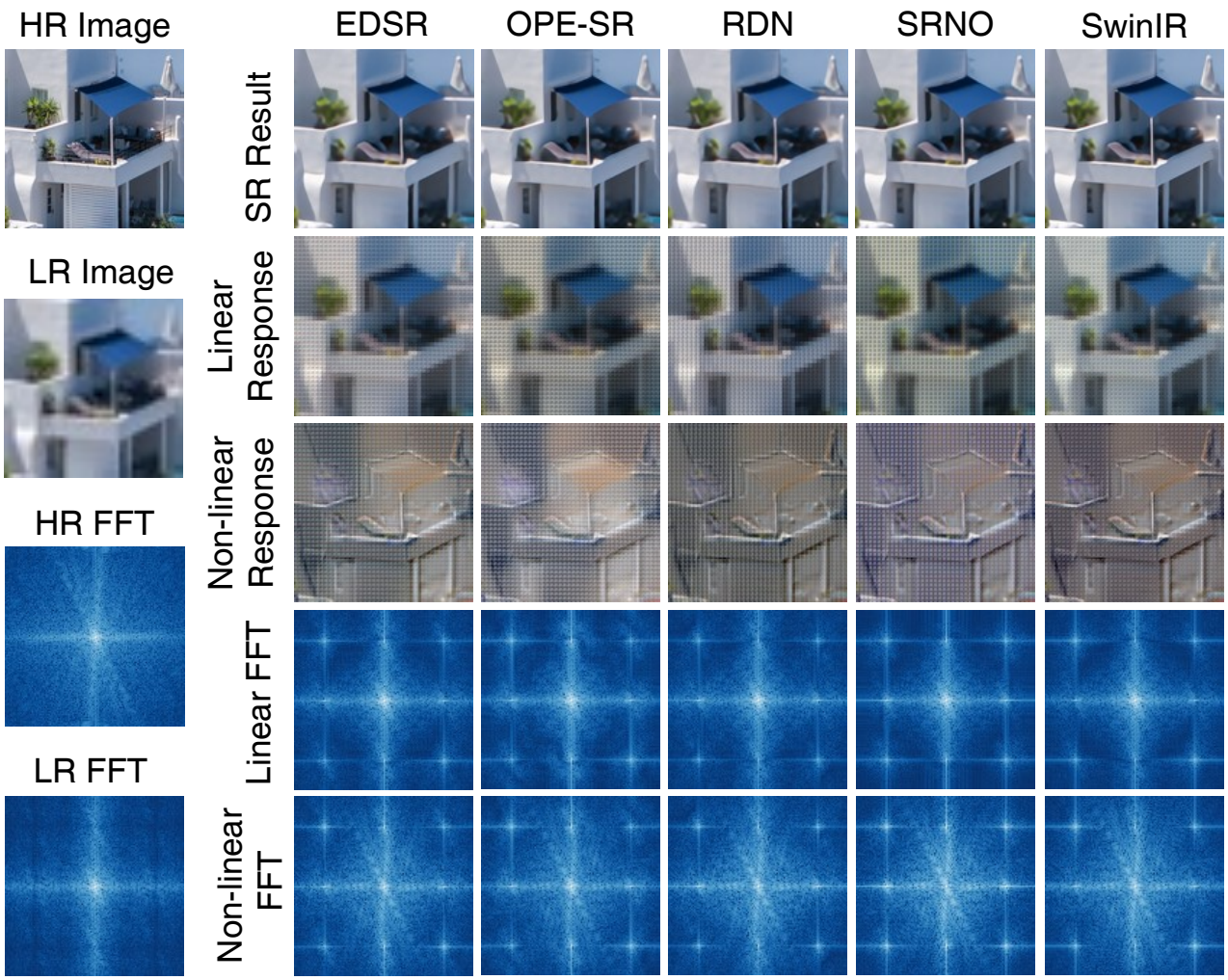

Figure 14. Linear and non-linear responses and their corresponding frequency spectrum of various ISR methods.

C.2. Space Invariance

We conducted spatial invariance testing on RDN (Zhang et al., 2018c) (For the concept of spatial invariance, please refer to Sec. 4.1). The input data consists of an image where only one pixel is white (the pixel value is 1), and all other pixels are black (the pixel value is 0). By shifting the position of this white pixel, we obtain $I(x - \Delta x, y - \Delta y)$. This shifted input $I(x - \Delta x, y - \Delta y)$ is then fed into the neural network, and we obtain its shifted impulse response, as illustrated in Fig. 15. Observing the experimental results, we find that the responses to different $I(x - \Delta x, y - \Delta y)$ are consistent, with the only difference being their position. This demonstrates that, for ISR networks, the linear component in HyRA exhibits spatial invariance.

C.3. Exploration of the Positional Origin of Sinc-like Patterns

As shown in Fig. 16, we visualize the output features of different components in the EDSR (Lim et al., 2017) network for analysis. We observe that the approximate shape of the sinc function begins to take form after the Upsampler module, and after a convolution, it essentially forms the shape of a sinc function. Interestingly, in the EDSR network, the Upsample

[6]The minimum sampling rate that can completely restore the origin of the sampled from sampled signal, which is twice the highest frequency of the original signal

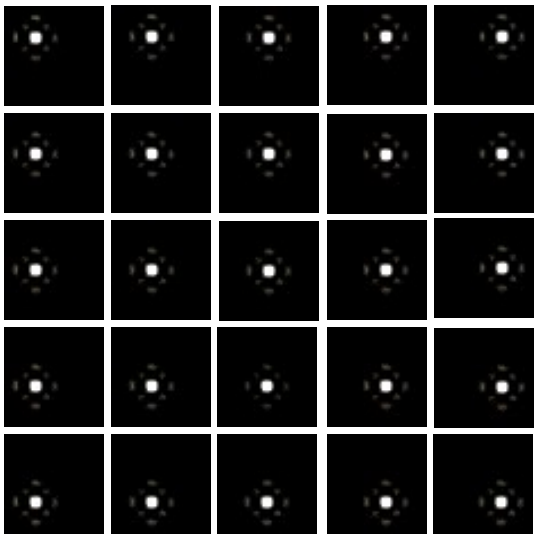

Figure 15. Spatial invariance experiment conducted on SwinIR (Liang et al., 2021). When we feed the SwinIR network with impulses at various positions, the ISR results demonstrate that the RDN exhibits spatial invariance.

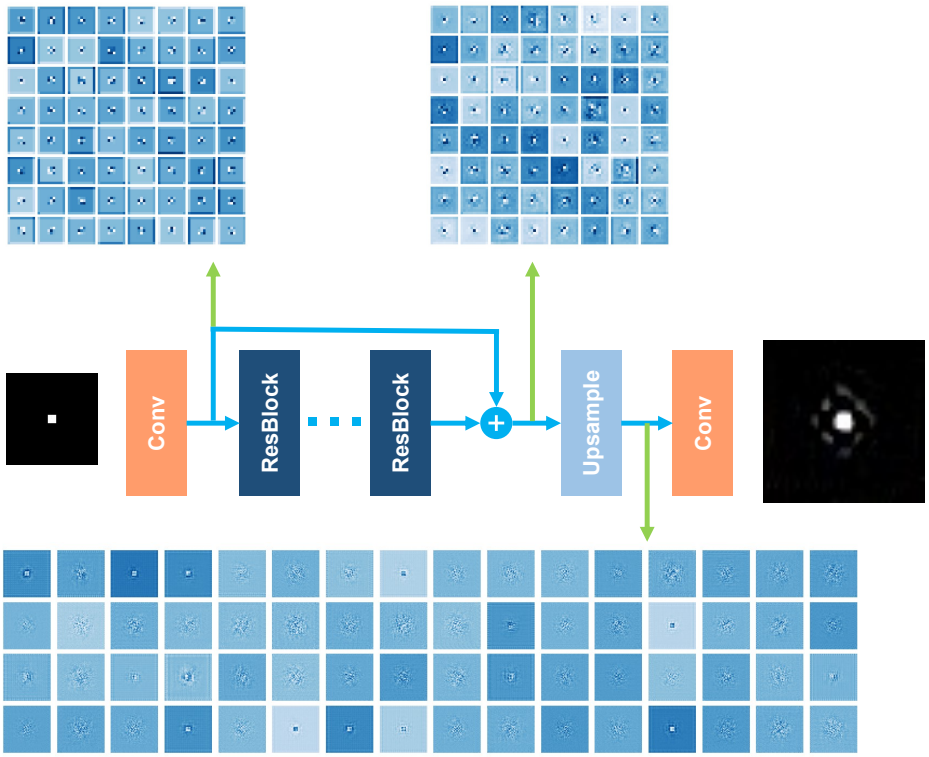

Figure 16. Visualization of feature maps in different EDSR (Lim et al., 2017) network layers. The sinc-like pattern start to take shapes after the sub-pixel convolution, before the last convolution layer.

module uses sub-pixel convolution (convolution + pixel shuffle) for upsampling without any interpolation. This indicates that the low-pass filter present in the network is learned by the network itself and not introduced by interpolation kernels.

D. The Fourier Transform Pairs Involved in This Paper

Symbol/Name	Section(s)	Time domain	Frequency Domain
$s_{\Delta T}(t)$, ΔT is the samping interval	Appx. B.2, Appx. B.3	$\sum_{n=-\infty}^{\infty} \delta(t - nT)$	$\frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\omega - k\frac{2\pi}{T})$
Ideal Low-pass filter	Sec. 4.1.1	$x(t) = sinc_{\omega_0}(t) = \frac{sin(\omega_0 t)}{\pi t}$, ω_0 is called the cut-off frequency	$X(j\omega) = \begin{cases} 1, & \lvert\omega\rvert < \omega_0 \\ 0, & \lvert\omega\rvert > \omega_0 \end{cases}$
$\delta(t)$	Sec. 4.1.1	$\lim_{\tau \to 0} \int_{-\tau}^{+\tau} \delta(t) = 1$	1

Table 3. Fourier transform pairs

E. The Windowing Operation

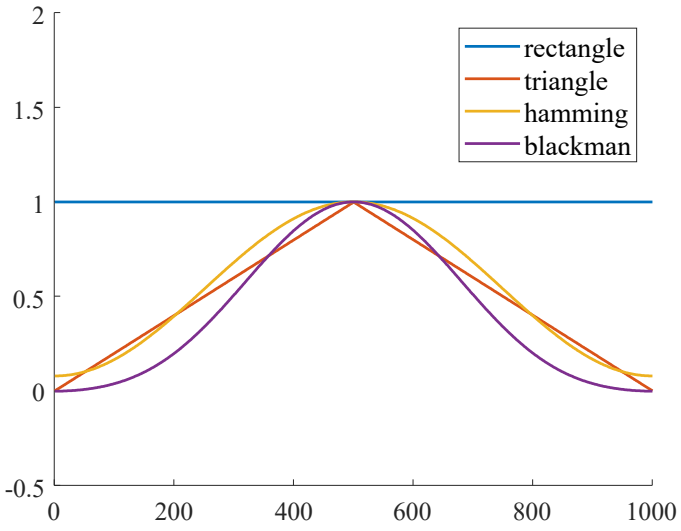

Figure 17. Various window functions.

The time-domain waveform of an ideal low-pass filter is a sinc function. The sinc function is defined over $[-\infty, \infty]$, and the number of zero crossings is countable. This implies that in reality, an ideal low-pass filter does not exist. In discrete-time signal processing, truncating a designed filter using a window function is common. There are many window functions, such as the rectangular window, Hanning window, Blackman window, and so on. Fig. 17 illustrates some commonly used window functions. Observing the experimental results and analyzing the relationship between the peak values of the main lobe and the first side lobe, we find that the impulse response of the neural network seems to undergo windowing. However, different networks appear to adopt different window functions.

F. Frequency Spectrum Period Extension Caused by Zero Padding

When considering integer factor ISR, our approach to computing the linear component response is as follows: first, upsample the low-resolution image to the high-resolution image through zero-padding, and then convolve it with the impulse response to obtain the response. During the zero-padding process, it leads to period extension in the frequency spectrum. For a signal

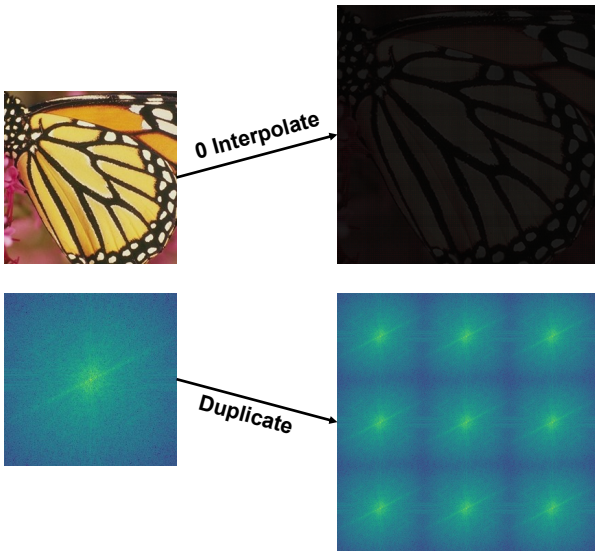

Figure 18. Performing zero-padding on an image to reach the target size will result in periodic extension in the frequency spectrum obtained through its Discrete Fourier Transform.

$x[n]$ of length N undergoing DFT to obtain $X[k]$, we have:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}. \tag{15}$$

Then, zero-padding is applied to $x[n]$, producing in a new signal $x_2[n]$ of length $3N$:

$$x_2[n] = \begin{cases} x[\frac{n}{3}], & n = 0, 3, \cdots, 3N-3 \\ 0, & otherwise. \end{cases} \tag{16}$$

Perform DFT to $x_2[n]$ to obtain $X'[k]$, then we have:

$$\begin{aligned} X'[k] &= \sum_{n=0}^{3N-1} x_2[n] e^{-j\frac{2\pi}{3N}k\cdot n} \\ &= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{3N}k\cdot 3n} \\ &= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \end{aligned} . \tag{17}$$

When $k < N$, there exists:

$$e^{-j\frac{2\pi}{N}kn} = e^{-j\frac{2\pi}{N}(k+N)n} = e^{-j\frac{2\pi}{N}(k+2N)n} = \cdots . \tag{18}$$

Therefore,

$$X'[k] = \begin{cases} X[k] & 0 \le k < N \\ X[k \; mod \; N] & N \le k < 3N-1 \end{cases} \tag{19}$$

Thus, zero-padding causes period extension in the frequency spectrum. Ideally, the extended spectrum would be filtered out by the low-pass filter used in the ISR process. However, due to the limited filtering capability of the potential filters within the neural network, the stopband attenuation is low, and the extended spectrum cannot be completely filtered out.

G. Comparison Between FSDS and ℓ_1, ℓ_2 norms

We compare our proposed FSDS metric with ℓ_1 norm, and ℓ_2 norm on both frequency domain and image domain as depicted in Tab. 4 and Tab. 5. From these two figures, we can observe that ℓ_1 norm and ℓ_2 norm produces similar ranking orders

when they are calculated on the same domain, indicating that ℓ_1 norm is equivalent to ℓ_2 norm when assessing image quality. However, the ranking orders produced by FSDS is distinctive to that of ℓ_1 and ℓ_2. This means our FSDS metric reflects image quality in a unique way.

Method	FSDS (Ours)					ℓ_1 Norm in Frequency Domain					ℓ_1 Norm				
	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12
EDSR(Lim et al., 2017)	39.210^{15}	34.148^{10}	31.380^{7}	-	-	35.190^{12}	51.676^{12}	62.478^{16}	-	-	0.012^{10}	0.019^{13}	0.023^{16}	-	-
EDSR-LIIF(Lim et al., 2017)	39.371^{13}	34.535^{6}	31.321^{9}	28.446^{4}	23.147^{3}	35.369^{14}	51.886^{14}	61.996^{15}	73.963^{4}	88.203^{5}	0.013^{15}	0.019^{15}	0.023^{15}	0.029^{4}	0.042^{4}
EDSR-OPESR(Lim et al., 2017)	39.798^{6}	34.630^{5}	31.286^{11}	-	-	36.796^{16}	51.948^{15}	61.711^{14}	-	-	0.013^{16}	0.018^{11}	0.023^{12}	-	-
EDSR-SRNO(Lim et al., 2017)	39.533^{11}	34.527^{7}	31.448^{6}	28.458^{3}	22.778^{4}	34.688^{10}	51.113^{10}	61.140^{11}	73.136^{3}	87.704^{3}	0.012^{11}	0.018^{10}	0.023^{10}	0.029^{3}	0.041^{3}
EDSR-LTE(Lim et al., 2017)	39.290^{14}	34.328^{8}	31.303^{10}	-	-	35.110^{11}	51.556^{11}	61.706^{12}	-	-	0.013^{14}	0.019^{14}	0.023^{14}	-	-
RDN(Zhang et al., 2018c)	40.022^{3}	32.946^{16}	31.646^{4}	-	-	34.595^{9}	53.431^{16}	61.069^{10}	-	-	0.013^{12}	0.019^{16}	0.023^{11}	-	-
RDN-LIIF(Zhang et al., 2018c)	39.690^{9}	34.831^{3}	31.832^{3}	28.894^{1}	23.778^{1}	34.224^{8}	50.434^{8}	60.563^{9}	72.753^{2}	87.488^{2}	0.012^{8}	0.018^{9}	0.022^{9}	0.029^{2}	0.041^{2}
RDN-OPESR(Zhang et al., 2018c)	40.188^{2}	34.959^{2}	31.475^{5}	-	-	36.126^{15}	50.792^{9}	60.388^{8}	-	-	0.013^{13}	0.018^{7}	0.022^{7}	-	-
RDN-LTE(Zhang et al., 2018c)	39.825^{5}	34.749^{4}	31.837^{2}	28.654^{2}	23.346^{2}	33.997^{7}	50.106^{7}	60.229^{7}	72.301^{1}	87.090^{1}	0.012^{7}	0.018^{8}	0.022^{8}	0.029^{1}	0.041^{1}
SwinIR-classical(Liang et al., 2021)	40.372^{1}	35.125^{1}	32.370^{1}	-	-	32.750^{5}	48.380^{4}	58.579^{4}	-	-	0.012^{5}	0.017^{5}	0.021^{5}	-	-
ITSRN(Yang et al., 2021)	31.254^{18}	26.178^{18}	25.876^{18}	25.619^{5}	21.566^{5}	41.928^{17}	53.459^{17}	62.927^{17}	74.260^{5}	88.160^{4}	0.015^{17}	0.020^{17}	0.024^{17}	0.030^{5}	0.042^{5}
Bicubic	32.790^{17}	28.896^{17}	26.568^{17}	23.450^{6}	18.736^{6}	50.571^{18}	64.949^{18}	73.412^{18}	82.567^{6}	93.012^{6}	0.018^{18}	0.024^{18}	0.029^{18}	0.036^{6}	0.050^{6}
HAT-S(Chen et al., 2023)	39.784^{7}	33.799^{13}	31.057^{14}	-	-	32.387^{2}	48.047^{3}	58.232^{3}	-	-	0.011^{2}	0.017^{3}	0.021^{3}	-	-
HAT(Chen et al., 2023)	39.784^{7}	33.913^{11}	31.196^{12}	-	-	32.387^{2}	47.815^{2}	58.052^{2}	-	-	0.011^{2}	0.017^{2}	0.021^{2}	-	-
HDSRNet(Tian et al., 2024)	39.458^{12}	34.203^{9}	31.325^{8}	-	-	35.245^{13}	51.690^{13}	61.709^{13}	-	-	0.012^{9}	0.019^{12}	0.023^{13}	-	-
GRLBase(Li et al., 2023)	39.990^{4}	33.840^{12}	31.121^{13}	-	-	31.807^{1}	47.263^{1}	57.296^{1}	-	-	0.011^{1}	0.017^{1}	0.021^{1}	-	-
GRLSmall(Li et al., 2023)	39.544^{10}	33.679^{14}	31.027^{15}	-	-	32.714^{4}	48.593^{5}	58.805^{5}	-	-	0.012^{4}	0.017^{4}	0.021^{4}	-	-
GRLTiny(Li et al., 2023)	39.205^{16}	33.197^{15}	30.556^{16}	-	-	33.394^{6}	49.550^{6}	59.835^{6}	-	-	0.012^{6}	0.018^{6}	0.022^{6}	-	-

Table 4. Comparison between our proposed FSDS metric and ℓ_1 norm in both frequency domain and image domain. Items with the highest mean values are highlighted in red and secondary mean values in blue. The gray superscripts denote the ranking order.

Method	FSDS (Ours)					ℓ_2 Norm in Frequency Domain					ℓ_2 Norm				
	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12	×2	×3	×4	×6	×12
EDSR(Lim et al., 2017)	39.210^{15}	34.148^{10}	31.380^{7}	-	-	4966.905^{13}	11163.148^{15}	17206.009^{16}	-	-	4966.906^{13}	11163.149^{15}	17206.010^{16}	-	-
EDSR-LIIF(Lim et al., 2017)	39.371^{13}	34.535^{6}	31.321^{9}	28.446^{4}	23.147^{3}	4967.727^{14}	11163.112^{14}	16947.235^{15}	27007.748^{4}	49280.883^{4}	4967.727^{14}	11163.113^{14}	16947.236^{15}	27007.749^{4}	49280.886^{4}
EDSR-OPESR(Lim et al., 2017)	39.798^{6}	34.630^{5}	31.286^{11}	-	-	5247.229^{16}	11117.694^{12}	16818.598^{14}	-	-	5247.229^{16}	11117.695^{12}	16818.599^{14}	-	-
EDSR-SRNO(Lim et al., 2017)	39.533^{11}	34.527^{7}	31.448^{6}	28.458^{3}	22.778^{4}	4794.903^{9}	10845.390^{10}	16453.701^{10}	26236.246^{3}	48097.246^{3}	4794.903^{9}	10845.391^{10}	16453.702^{10}	26236.247^{3}	48097.249^{3}
EDSR-LTE(Lim et al., 2017)	39.290^{14}	34.328^{8}	31.303^{10}	-	-	4907.855^{11}	11032.252^{11}	16781.621^{13}	-	-	4907.855^{11}	11032.252^{11}	16781.622^{13}	-	-
RDN(Zhang et al., 2018c)	40.022^{3}	32.946^{16}	31.646^{4}	-	-	4820.540^{10}	12055.745^{16}	16462.594^{11}	-	-	4820.541^{10}	12055.746^{16}	16462.595^{11}	-	-
RDN-LIIF(Zhang et al., 2018c)	39.690^{9}	34.831^{3}	31.832^{3}	28.894^{1}	23.778^{1}	4627.806^{8}	10477.303^{8}	15993.797^{8}	25720.766^{2}	47650.681^{2}	4627.806^{8}	10477.304^{8}	15993.798^{8}	25720.768^{2}	47650.683^{2}
RDN-OPESR(Zhang et al., 2018c)	40.188^{2}	34.959^{2}	31.475^{5}	-	-	5061.535^{15}	10609.168^{9}	16034.925^{9}	-	-	5061.535^{15}	10609.169^{9}	16034.926^{9}	-	-
RDN-LTE(Zhang et al., 2018c)	39.825^{5}	34.749^{4}	31.837^{2}	28.654^{2}	23.346^{2}	4576.652^{7}	10349.273^{7}	15808.890^{7}	25384.597^{1}	46955.729^{1}	4576.652^{7}	10349.274^{7}	15808.891^{7}	25384.598^{1}	46955.731^{1}
SwinIR-classical(Liang et al., 2021)	40.372^{1}	35.125^{1}	32.370^{1}	-	-	4222.226^{5}	9617.212^{5}	14804.533^{5}	-	-	4222.226^{5}	9617.212^{5}	14804.534^{5}	-	-
ITSRN(Yang et al., 2021)	31.254^{18}	26.178^{18}	25.876^{18}	25.619^{5}	21.566^{5}	7483.582^{17}	12244.897^{17}	17878.399^{17}	27664.054^{5}	49417.061^{5}	7483.582^{17}	12244.898^{17}	17878.400^{17}	27664.056^{5}	49417.064^{5}
Bicubic	32.790^{17}	28.896^{17}	26.568^{17}	23.450^{6}	18.736^{6}	10873.900^{18}	19564.473^{18}	26817.580^{18}	38541.675^{6}	64011.698^{6}	10873.901^{18}	19564.474^{18}	26817.581^{18}	38541.677^{6}	64011.702^{6}
HAT-S(Chen et al., 2023)	39.784^{7}	33.799^{13}	31.057^{14}	-	-	4107.382^{2}	9423.447^{3}	14498.509^{3}	-	-	4107.383^{2}	9423.448^{3}	14498.510^{3}	-	-
HAT(Chen et al., 2023)	39.784^{7}	33.913^{11}	31.196^{12}	-	-	4107.382^{2}	9326.613^{2}	14407.457^{2}	-	-	4107.383^{2}	9326.613^{2}	14407.458^{2}	-	-
HDSRNet(Tian et al., 2024)	39.458^{12}	34.203^{9}	31.325^{8}	-	-	4948.237^{12}	11121.912^{13}	16697.185^{12}	-	-	4948.238^{12}	11121.912^{13}	16697.186^{12}	-	-
GRLBase(Li et al., 2023)	39.990^{4}	33.840^{12}	31.121^{13}	-	-	3937.013^{1}	9073.913^{1}	13959.078^{1}	-	-	3937.013^{1}	9073.913^{1}	13959.079^{1}	-	-
GRLSmall(Li et al., 2023)	39.544^{10}	33.679^{14}	31.027^{15}	-	-	4170.853^{4}	9585.326^{4}	14742.787^{4}	-	-	4170.853^{4}	9585.327^{4}	14742.788^{4}	-	-
GRLTiny(Li et al., 2023)	39.205^{16}	33.197^{15}	30.556^{16}	-	-	4384.364^{6}	10079.017^{6}	15488.316^{6}	-	-	4384.364^{6}	10079.018^{6}	15488.317^{6}	-	-

Table 5. Comparison between our proposed FSDS metric and ℓ_2 norm in both frequency domain and image domain. Items with the highest mean values are highlighted in red and secondary mean values in blue. The gray superscripts denote the ranking order. To demonstrate Parseval's theorem, we omit the mean operation when calculating ℓ_2 norm.

H. How G(I) is Trained and How G(I) and H(I) Vary Dependently

To better explain how $G(I)$ is trained and how $G(I)$ and $H(I)$ vary dependently, we conduct a new experiment on observing their varying progress during training. We train a vanilla RDN (Zhang et al., 2018c) network from scratch and obtain the impulse response, $H(I)$, $G(I)$, and $N(I)$ from each epoch (see Fig. 19). As shown in the figure, in the second row, the $sinc$ phenomenon becomes clearer along with the training process, this indicates that the network is gradually learning the low-pass filter. The same conclusion can be further supported by observing the variation in $H(I)$ as shown in the third row. In this row, $H(I)$ illustrates the phenomenon of the grid-like distortion vanishing while low-frequency areas getting smoother. The fourth row depict the plot of $G(I)$. This row demonstrates that $G(I)$ is capturing more and more high-frequency information, such as edges. This is very significant especially when comparing the result from Epoch 1 and Epoch 50. Such increase in high-frequncy information can also be found in the frequency spectrum. During the entire training process, the network fails to recover the words on the wall (please see the magnified area, there are words on the wall next to the blue awning). Observing $G(I)$, we can find no sign of words as well. This indicates that the network treats it as low-frequency information and ignores it, pointing the way for future network improvements.

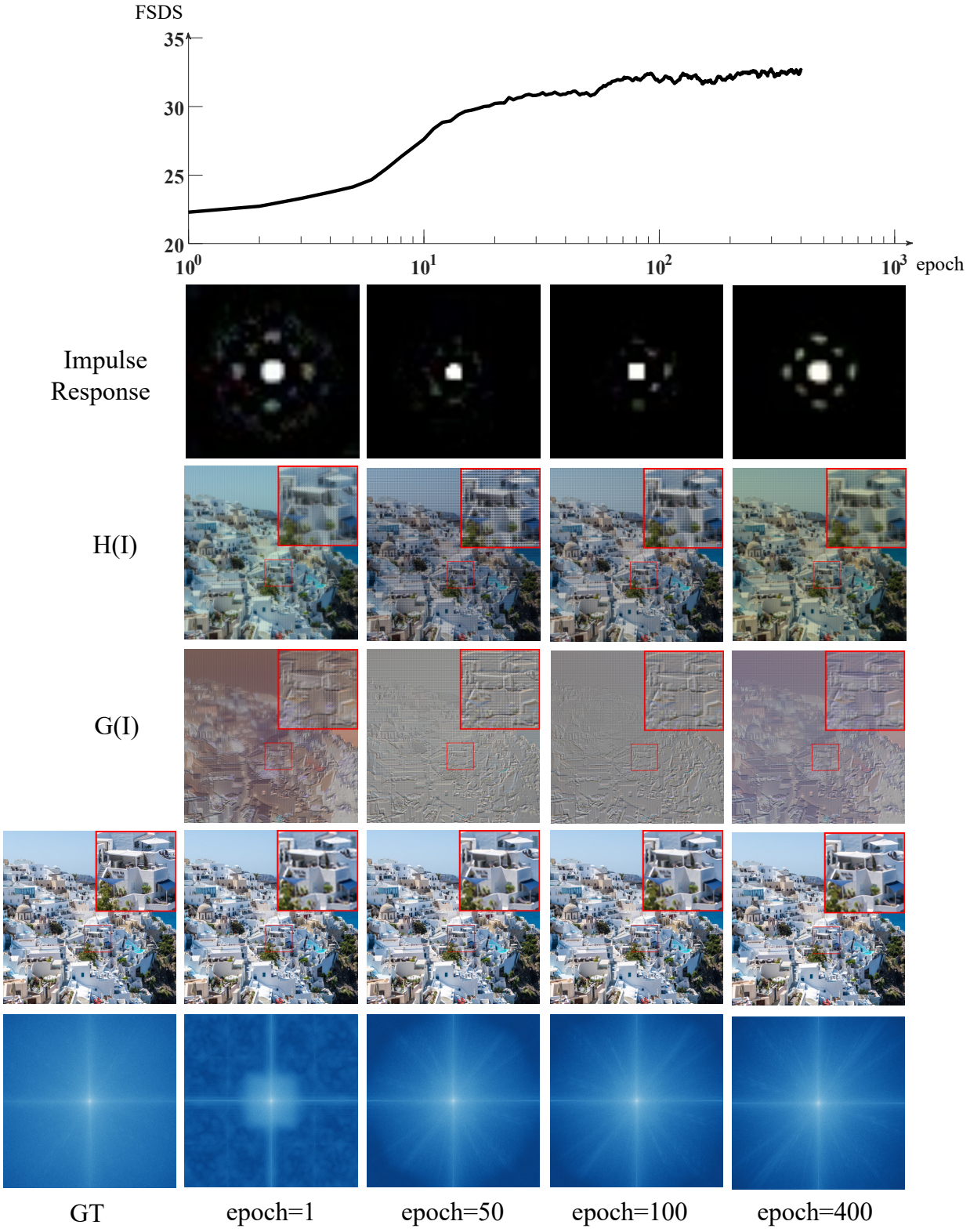

Figure 19. $N(I)$, $H(I)$ and $G(I)$ from different epoches during training.

I. Code Repositories

Abbreviate	Title	Publication	Year	Code Link
EDSR (Lim et al., 2017)	Enhanced Deep Residual Networks for Single Image Super-Resolution	CVPRW	2017	Github
LIIF(Chen et al., 2021)	Learning Continuous Image Representation with Local Implicit Image Function	CVPR	2021	Github
OPE-SR(Song et al., 2023)	OPE-SR: Orthogonal Position Encoding for Designing a Parameter-Free Upsampling Module in Arbitrary-Scale Image Super-Resolution	CVPR	2023	Github
SRNO(Wei & Zhang, 2023)	Super-Resolution Neural Operator	CVPR	2023	Github
LTE (Lee & Jin, 2022)	Local Texture Estimator for Implicit Representation Function	CVPR	2022	Github
RDN (Zhang et al., 2018c)	Residual Dense Network for Image Super-Resolution	CVPR	2018	Github
SwinIR (Liang et al., 2021)	SwinIR: Image Restoration Using Swin Transformer	ICCV	2021	Github
ITSRN (Yang et al., 2021)	Implicit Transformer Network for Screen Content Image Continuous Super-Resolution	NeurIPS	2021	Github
RCAN (Zhang et al., 2018b)	Image Super-Resolution Using Very Deep Residual Channel Attention Networks	ECCV	2018	Github
HAT (Chen et al., 2023)	Activating More Pixels in Image Super-Resolution Transformer	CVPR	2023	Github
HDSRNet (Tian et al., 2024)	Heterogeneous Dynamic Convolutional Network in Image Super-Resolution	Arxiv	2024	Github
GRL (Li et al., 2023)	Efficient and Explicit Modelling of Image Hierarchies for Image Restoration	CVPR	2023	Github

Table 6. The papers and repository links used in this paper.