DIIN: Diffusion Iterative Implicit Network for Arbitrary-scale Super-resolution

Tao Dai[1] , **Song Wang**[1] , **Hang Guo**[2, *] , **Jianping Wang**[1] , **Zexuan Zhu**[3,4]
[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University
[3]School of Artificial Intelligence, Shenzhen University
[4]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University
{daitao.edu, cshguo}@gmail.com, {wangsong2022,wangjianping2022}@email.szu.edu.cn,
zhuzx@szu.edu.cn

Abstract

Implicit neural representation (INR) aims to represent continuous domain signals via implicit neural functions and has achieved great success in arbitrary-scale image super-resolution (SR). However, most existing INR-based SR methods focus on learning implicit features from independent coordinate, while neglecting interactions of neighborhood coordinates, thus resulting in limited contextual awareness. In this paper, we rethink the forward process of implicit neural functions as a signal diffusion process, we propose a novel Diffusion Iterative Implicit Network (DIIN) for arbitrary-scale SR to promote global signal flow with neighborhood interactions. The DIIN framework mainly consists of stacked Diffusion Iteration Layers with dictionary cross-attention block to enrich the iterative update process with supplementary information. Besides, we develop the Position-Aware Embedding Block to strengthen spatial dependencies between consecutive input samples. Extensive experiments on public datasets demonstrate that our method achieves state-of-the-art or competitive performance, highlighting its effectiveness and efficiency for arbitrary-scale SR. Our code is available at https://github.com/Song-1205/DIIN.

1 Introduction

Single Image Super-Resolution (SISR), which aims to reconstruct high-resolution (HR) images from corresponding low-resolution (LR) observations, has recently gained great progress. Most exciting SISR methods [Dai *et al.*, 2019; Liang *et al.*, 2021; Dai *et al.*, 2024; Dong *et al.*, 2015; Zhang *et al.*, 2018; Zhou *et al.*, 2023] focus on fixed-scale setups, where models are optimized predefined up-sampling factors (e.g.g. $2\times$, $4\times$) during training. However, such fixed-scale setup hinders the applications in real-world scenarios, where continuous arbitrary-scale factors are desired.

To upsample continuous arbitrary-scale factors, arbitrary-scale super-resolution (ASSR) has recently received much attention. In particular, implicit neural representation (INR) based SR methods [Chen *et al.*, 2021; Lee and Jin, 2022; Chen *et al.*, 2023] have achieved impressive performance by modeling image signals as continuous functions. Specifically, this type of method fits desired coordinates into the implicit neural function parameterized by a neural network to query image pixels at any position. In this way, the corresponding pixel signal can be generated directly at any coordinate, eliminating the constraints of pre-defined scales. However, most existing ASSR methods rely on point-wise multi-layer perceptions (MLPs) to parameterize the implicit neural function. However, such a design inherently treats different query positions independently, thus limiting the flow of contextual information between pixels. As illustrated in Figure 1, LIIF [Chen *et al.*, 2021] and SRNO [Wei and Zhang, 2023] operate with restricted receptive fields, relying on limited contextual cues. In contrast, our proposed method expands the receptive field and adaptively incorporates relevant information from surrounding pixels.

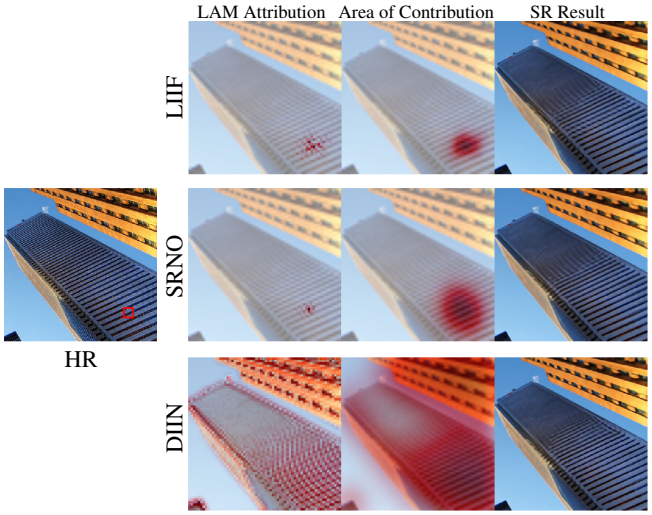


Figure 1: Comparison of LAM [Gu and Dong, 2021]. DIIN benefits from a significantly larger receptive field compared to LIIF and SRNO, allowing it to incorporate broader contextual cues. As a result, it produces more accurate and visually consistent reconstructions, particularly in regions with complex structures or boundaries.

The above observations motivate us to incorporate more

*Corresponding author: Hang Guo (cshguo@gmail.com)

contextual information in implicit neural representation. To this end, we rethink the modeling process of the implicit neural functions to inject neighborhood interaction and attempt to re-project the forward process of implicit neural functions as a signal diffusion process, where the signal transmission between different query locations adapts over time. Specifically, we define the diffusion rate to depict the mutual influence between any two nodes at a certain time step. The diffusion rate can measure the rate at which the signal flows from one node to another.

Inspired by the above analysis, we can derive the diffusion implicit function that describes the state updates of all nodes at each time step, and propose an efficient Diffusion Iterative Implicit Network (DIIN) with global signal flows for arbitrary-scale SR, which mainly consists of stacked diffusion iteration layers to help the sampled representations undergo iterative updates layer by layer. In this way, our method explicitly captures the dependencies between query locations and fully utilizes the contextual information in the image space. Furthermore, we introduce a set of adaptive token dictionaries, which interact with the latent code output from the encoder through a cross-attention module in an end-to-end manner to generate sparse representations in the image feature space. These token dictionaries provide auxiliary information support for the iterative updates of each query point, effectively addressing the problem of incomplete contextual information due to the random sampling of HR coordinates. Besides, we introduce a position-aware Embedding Block, which adaptively modulates the sampled features based on query coordinates, thereby enhancing the spatial dependencies between subsequent input samples.

The main contributions are summarized as follows:

- We rethink the forward process of implicit neural functions as a signal diffusion process, and propose a novel Diffusion Iterative Implicit Network (DIIN) with stacked diffusion iteration layers to promote global signal flow with neighborhood interactions.
- We further introduce a set of adaptive tokens through cross-attention modules, interacting with the latent codes output by the encoder to generate sparse image feature space representations. This design effectively supports the iterative update of query points, addressing the issue of incomplete contextual information.
- Extensive experiments demonstrate the effectiveness of our method over other state-of-the-art methods across various in-distribution and out-of-distribution upsampling factors.

2 Related Work

2.1 Implicit Neural Representation

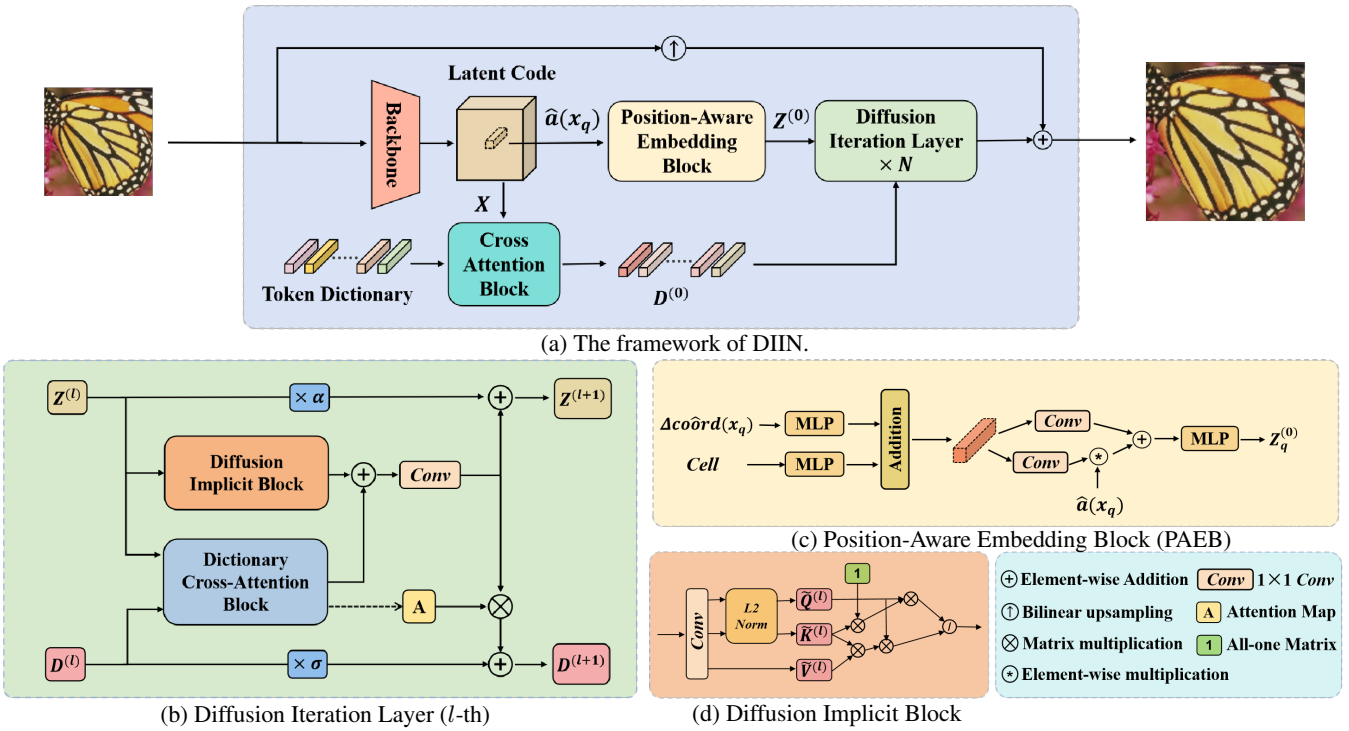
Implicit neural representations (INR) build the mapping between coordinates and their signal values using a neural network, enabling continuous and memory-efficient modeling for various signal types (e.g., 1D audio [Gao *et al.*, 2022], 2D images [Tancik *et al.*, 2020], 3D shapes [Park *et al.*, 2019; Mildenhall *et al.*, 2020]). In recent years, implicit neural representations have seen significant advancements in 2D applications, such as image representation [Sitzmann *et al.*, 2020; Xie *et al.*, 2023] and image super-resolution [Liang *et al.*, 2021; Lee and Jin, 2022]. For image super-resolution, [Chen *et al.*, 2021] proposed a method that uses implicit functions to model continuous image signals. This approach no longer relies on fixed-resolution storage of image signals, but can directly generate corresponding signal values at arbitrary coordinates, thereby overcoming the limitations of traditional methods that depend on predefined scales, offering greater flexibility and accuracy.

2.2 Single Image Super-Resolution

Single Image Super-Resolution (SISR) aims to reconstruct high-resolution (HR) images from low-resolution (LR) observations, with the goal of improving perceptual quality while accurately recovering spatial details. Since the introduction of SRCNN [Dong *et al.*, 2015], the first deep learning-based SISR model, various implementation architectures have emerged in the field with the development of new models. For example, CNN-based methods [Dai *et al.*, 2019; Lim *et al.*, 2017; Zhang *et al.*, 2018; Kim *et al.*, 2016], transformer-based methods [Liang *et al.*, 2021; Dai *et al.*, 2024; Zhang *et al.*, 2024; Guo *et al.*, 2024a], diffusion-based methods [Saharia *et al.*, 2021; Xia *et al.*, 2023; Wang *et al.*, 2024b; Guo *et al.*, 2024b; Wu *et al.*, 2024; Wang *et al.*, 2024a; Lin *et al.*, 2024], and SSM-based methods [Guo *et al.*, 2024d; Ren *et al.*, 2024; Guo *et al.*, 2024c] have driven the development of the field, each leveraging the unique characteristics of their respective architectures. However, these methods are limited to fixed scales, which makes them less suitable for real-world applications where continuous scaling adjustments are required.

2.3 Arbitrary-Scale Super-Resolution

To overcome the above limitation, Arbitrary-Scale Super-Resolution (ASSR) has become a prominent research focus, allowing image reconstruction at any up-sampling factor with a single model. MetaSR [Hu *et al.*, 2019] is among the first works to tackle this problem, introducing a meta-upscale module that predicts convolutional filter weights based on coordinates and scaling factors. Inspired by INR, LIIF [Chen *et al.*, 2021] learns implicit features from local regions to predict the RGB value at arbitrary coordinates. Subsequent works have focused on improving performance by incorporating more features into the MLP-parameterized continuous function. For instance, LTE [Lee and Jin, 2022] proposed a local texture estimator that maps coordinates to Fourier domain information, enhancing the expressive power of its local implicit function. CLIT [Chen *et al.*, 2023] combines local attention mechanisms with frequency encoding techniques to enhance the ability to capture fine details. Meanwhile, an accumulative training strategy and a cascaded framework are employed to optimize the training process. Although these INR-based methods achieve state-of-the-art results in arbitrary-scale super-resolution, they often parameterize the implicit neural function using point-wise MLPs, which results in the independence of different query positions, limiting the signal flow between pixels.



(a) The framework of DIIN.

(b) Diffusion Iteration Layer (l-th)

(c) Position-Aware Embedding Block (PAEB)

(d) Diffusion Implicit Block

Figure 2: The overall architecture of our Diffusion Iteration Image Network (DIIN) comprises N Diffusion Iteration Layers, each implementing a Diffusion Iterative Implicit Function. Each layer consists of a Dictionary Cross-Attention Block (DCAB) and a Diffusion Implicit Block (DIB).

3 Method

3.1 Signal Diffusion Perspective

We propose the Diffusion Iterative Implicit Network (DIIN) to address the limitations of existing implicit neural representation methods, which either lack contextual interaction or suffer from high computational complexity. Our method rethinks the forward computation of implicit neural functions as a dynamic signal diffusion process, enabling efficient and adaptive information propagation between query positions while maintaining linear computational complexity.

Inspired by the analogy between neural network computation and heat conduction in thermodynamics [Wu *et al.*, 2023], we model the signal flow between query positions as a dynamic diffusion process. This process is governed by a diffusion rate $\mathbf{S}_{ij}(\mathbf{Z}(t), t)$, which measures the influence of node. The diffusion process is expressed as:

$$\frac{\partial \mathbf{z}_i(t)}{\partial t} = \sum_{j=1}^{N} \mathbf{S}_{ij}(\mathbf{Z}(t), t)(\mathbf{z}_j(t) - \mathbf{z}_i(t)) \tag{1}$$

Using a numerical finite difference method, we discretize the above equation into an iterative update form:

$$\mathbf{z}_i^{(k+1)} = \alpha \mathbf{z}_i^{(k)} + (1-\alpha) \sum_{j=1}^{N} \mathbf{S}_{ij}^{(k)} \mathbf{z}_j^{(k)} \tag{2}$$

where α is a constant and $\mathbf{S}_{ij}^{(k)}$ is the diffusion rate matrix. By making different assumptions, this framework can generalize existing models, such as:

- **MLP:** When $\mathbf{S}_{ij}^{(k)}$ is an identity matrix, each query position is computed independently.
- **Attention:** When $\mathbf{S}_{ij}^{(k)}$ allows interactions between all positions, the model captures global dependencies but incurs quadratic complexity.

3.2 Diffusion Iterative Implicit Function (DIIF)

DIIF combines the strengths of MLPs and attention mechanisms by efficiently modeling contextual interactions with linear complexity. It represents the image signal as a continuous function, taking the latent code $\mathbf{M}$ extracted by an encoder and query coordinates x_q as input to predict pixel values at specified coordinates:

$$\mathbf{I}(x_q) = f_\theta(\mathbf{M}, x_q) \tag{3}$$

For a set of query coordinates x_q, the corresponding latent codes $\mathbf{X}$ are retrieved from $\mathbf{M}$ and passed through a neural network with layers $f^{(l)}$:

$$\begin{aligned} \mathbf{Z}^{(0)} &= \mathbf{X} \\ \mathbf{Z}^{(l+1)} &= f^{(l)}(\mathbf{Z}^{(l)}), \quad l = 0, 1, \ldots, L-1, \\ \hat{\mathbf{Y}} &= f^{(L)}(\mathbf{Z}^{(L)}) \end{aligned} \tag{4}$$

where $\mathbf{X}$ represents the corresponding latent codes, and $\hat{\mathbf{Y}}$ is the output prediction. To model dependencies between query coordinates, DIIF adopts the iterative update function from

Eqn. 2. The diffusion rate $\hat{\mathbf{S}}_{ij}^{(k)}$ is computed as:

$$\hat{\mathbf{S}}_{ij}^{(k)} = \frac{f(\|\mathbf{z}_i^{(k)} - \mathbf{z}_j^{(k)}\|_2^2)}{\sum_{l=1}^{N} f(\|\mathbf{z}_i^{(k)} - \mathbf{z}_l^{(k)}\|_2^2)}, \quad 1 \le i, j \le N \tag{5}$$

where $f(\left\|\widetilde{\mathbf{z}}_i^{(k)} - \widetilde{\mathbf{z}}_j^{(k)}\right\|_2^2)$ measures the signal flow between nodes (i, j). To reduce complexity from $O(N^2)$ to $O(N)$, we approximate the diffusion rate using a dot-product with L2 normalization:

$$\omega_{ij}^{(k)} = f(\left\|\widetilde{\mathbf{z}}_i^{(k)} - \widetilde{\mathbf{z}}_j^{(k)}\right\|_2^2) = \mathbf{1} + \left(\frac{\mathbf{z}_i^{(k)}}{\left\|\mathbf{z}_i^{(k)}\right\|_2}\right)^{\mathrm{T}} \left(\frac{\mathbf{z}_j^{(k)}}{\left\|\mathbf{z}_j^{(k)}\right\|_2}\right) \tag{6}$$

Furthermore, the computation of the aggregated signal flow is expressed as:

$$\begin{aligned} \sum_{j=1}^{N} \mathbf{S}_{ij}^{(k)} \mathbf{z}_j^{(k)} &= \sum_{j=1}^{N} \frac{1 + (\tilde{\mathbf{z}}_i^{(k)})^\top \tilde{\mathbf{z}}_j^{(k)}}{\sum_{l=1}^{N} \left(1 + (\tilde{\mathbf{z}}_i^{(k)})^\top \tilde{\mathbf{z}}_l^{(k)}\right)} \mathbf{z}_j^{(k)} \\ &= \frac{\sum_{j=1}^{N} \mathbf{z}_j^{(k)} + \left(\sum_{j=1}^{N} \tilde{\mathbf{z}}_j^{(k)} \cdot (\mathbf{z}_j^{(k)})^\top\right) \cdot \tilde{\mathbf{z}}_i^{(k)}}{N + (\tilde{\mathbf{z}}_i^{(k)})^\top \sum_{l=1}^{N} \tilde{\mathbf{z}}_l^{(k)}} \end{aligned} \tag{7}$$

To enhance contextual modeling, we introduce an auxiliary adaptive token dictionary. This dictionary learns prior contextual information from the input data and aligns it with specific test images. The Diffusion Iterative Implicit Function is modified as:

$$\mathbf{z}_i^{(k+1)} = \alpha \mathbf{z}_i^{(k)} + (1 - \alpha) \sum_{j=1}^{N} (\mathbf{S}_{ij}^{(k)} + \mathbf{A}_{ij}^{(k)}) \mathbf{z}_j^{(k)} \tag{8}$$

This integration ensures robust predictions by leveraging both contextual and adaptive information, enabling efficient and accurate super-resolution across arbitrary scales.

3.3 Network Architecture

Based on the above discussion, we modeled the proposed Diffusion Iteration Implicit Networks (DIIN), as illustrated in Figure 2. In this section, we will provide a detailed explanation of the implementation of each corresponding module.

Position-Aware Embedding Block. To enhance the spatial dependencies between input samples in the Diffusion Iteration Image Function, we generate conditional embeddings using coordinate and scale information. These conditional embeddings are used to adaptively modulate the sampled features, thereby strengthening the spatial correlation between input samples. This process provides more distinctive features for the subsequent diffusion process.

Specifically, given query HR coordinates $x_q \in \mathbf{x}^{HR}$, the corresponding input features $z_q \in \mathbb{R}^d$ for the Diffusion Iteration Layer are obtained through modulation by the Positional-Aware Embedding Block (PAEB):

$$z_q = PAEB(\hat{a}(x_q), \Delta\hat{coord}(x_q), cell), \tag{9}$$

$$\Delta\hat{coord}(x_q) = \mathrm{Concat}(\{x_q - \hat{x}_l\}_{l=1}^{4}), \tag{10}$$

$$\hat{a}(x_q) = \mathrm{Concat}(\{s_l \cdot a_l\}_{l=1}^{4}), \tag{11}$$

where x_l and a_l represent the coordinates and corresponding features of the four neighbors of x_q, respectively. $\Delta\hat{coord}(x_q)$ denotes the relative coordinates between x_q and its four neighbors. s_l is the feature integration weight used to reduce blocky artifacts generated by direct interpolation of low-resolution feature maps. $cell = (2/r_x, 2/r_y)$ represent a local region of size $r_x \times r_y$ in HR image, where r_x and r_y are the scaling factors.

As depicted in Figure 2(c), $\Delta\hat{coord}(x_q)$ and $cell$ are processed through two separate MLPs to obtain their respective embeddings. These embeddings are subsequently fused to produce the corresponding conditional embedding. This conditional embedding is then used to adaptively modulate the sampled features, resulting in the initial embeddings $Z_q^{(0)}$ for the Diffusion Iteration Layer.

Diffusion Implicit Block. As shown in Figure 2(b), we model the implicit iterative process at each layer based on Eqn. 7, where the input features are transformed into Q, K, and V, resulting in the following expressions:

$$\mathbf{R} = \mathrm{diag}^{-1}\left(N + \tilde{\mathbf{Q}}\left((\tilde{\mathbf{K}})^\top \mathbf{1}\right)\right), \tag{12}$$

$$\mathbf{P} = \mathbf{R}\left[\mathbf{1}\left(\mathbf{1}^\top \mathbf{V}\right) + \tilde{\mathbf{Q}}\left(\tilde{\mathbf{K}}^\top \mathbf{V}\right)\right], \tag{13}$$

where $\mathbf{1}_{N\times 1}$ is an all-one vector, $\tilde{\mathbf{Q}} = \mathbf{L2}(Q)$ and $\tilde{\mathbf{K}} = \mathbf{L2}(K)$ denotes $L2$ normalization applied to the query and key vectors, respectively.

Specifically, given the input features of each layer $\mathbf{Z} \in \mathbb{R}^{H\times W\times C}$, we first apply a 1×1 convolution layer to transform the features. The resulting features are then divided into three branches: (1) Two branches undergo $L2$ normalization to generate the query vector $\mathbf{Q}$ and key vector $\mathbf{K}$, which are used to compute attention weights; (2) The third branch retains the unnormalized features $\mathbf{V}$ as the value for feature enhancement.

Subsequently, the diagonal matrix $\mathbf{R}$ and the weighted fusion of $\mathbf{V}$ are calculated based on Eqn.12 and Eqn.13, producing the final output $\mathbf{P}$. The diagonal matrix $\mathbf{R}$ balances feature importance by aggregating global and local information, while the subsequent feature aggregation combines global pooling and local attention to effectively integrate global and local dependencies, yielding the enhanced features.

Dictionary Cross-Attention Block. Since query points are randomly sampled from the HR image, the input samples lack complete contextual information. To address this, we employ an adaptively refined token dictionary to provide auxiliary support for the iterative updates of each query point.

Inspired by [Zhang *et al.*, 2024], we introduce an additional dictionary, $\mathbf{D} \in \mathbb{R}^{M\times d}$, initialized as network parameters to encapsulate external priors during training. As illustrated in Figure 2, the learned token dictionary $\mathbf{D}$ interacts with the latent code $\mathbf{X}$ via a cross-attention block (CSB) to adaptively capture global information, which serves as auxiliary input for the subsequent Diffusion Iteration Layer. The dictionary $\mathbf{D}$ is further used to generate the Key dictionary $\mathbf{K_D}$ and the Value dictionary $\mathbf{V_D}$. Simultaneously, the input feature $\mathbf{X} \in \mathbb{R}^{N\times d}$ generates Query tokens:

$$Q_X = XW^Q, \quad K_D = DW^K, \quad V_D = DW^V, \tag{14}$$

Encoder	Method	In-distribution			Out-of-distribution				
		×2	×3	×4	×6	×12	×18	×24	×30
-	*Bicubic*	31.01	28.22	26.66	24.82	22.27	21.00	20.19	19.59
EDSR_baseline	EDSR only	34.52	30.89	28.98	-	-	-	-	-
	MetaSR	34.64	30.93	28.92	26.61	23.55	22.03	21.06	20.37
	LIIF	34.67	30.96	29.00	26.75	23.71	22.17	21.28	20.48
	LTE	34.72	31.02	29.04	26.81	23.78	22.23	21.24	20.53
	CLIT	34.81	31.12	29.15	26.92	23.83	22.29	21.26	20.53
	SRNO	34.85	31.11	29.16	26.90	23.84	22.29	21.27	20.56
	DIIN (ours)	34.87	31.13	29.18	26.92	23.86	22.31	21.32	20.61
RDN	RDN only	34.59	31.03	29.12	-	-	-	-	-
	MetaSR	35.00	31.27	29.25	26.88	23.73	22.18	21.17	20.47
	LIIF	34.99	31.26	29.27	26.99	23.89	22.34	21.31	20.59
	LTE	35.04	31.32	29.33	27.04	23.95	22.40	21.36	20.64
	CLIT	35.10	31.39	29.39	27.12	24.01	22.45	21.38	20.64
	SRNO	35.16	31.42	29.42	27.12	24.03	22.46	21.41	20.68
	DIIN (ours)	35.17	31.44	29.43	27.13	24.05	22.49	21.44	20.70

Table 1: Quantitative comparison with state-of-the-art methods for arbitrary-scale super-resolution on the DIV2K validation set (PSNR in dB). In each column, the best result is highlighted in red, while the second-best result is highlighted in blue. '-' indicates that the result is unavailable in the literature or the model's source code has not been released.

where $\mathbf{W_Q} \in \mathbb{R}^{d\times d/r}$, $\mathbf{W_K} \in \mathbb{R}^{d\times d/r}$, and $\mathbf{W_V} \in \mathbb{R}^{d\times d}$ are linear transformation matrices for the query tokens, key dictionary tokens, and value dictionary tokens, respectively. Then, we use the key dictionary and the value dictionary to enhance query tokens via cross-attention calculation:

$$\begin{aligned} \boldsymbol{A} &= \mathrm{SoftMax}(\mathrm{Sim}_{\cos}(\boldsymbol{Q}_X, \boldsymbol{K}_D)/\tau), \\ \mathrm{DCAB}(\boldsymbol{Q}_X, \boldsymbol{K}_D, \boldsymbol{V}_D) &= \boldsymbol{S}, \boldsymbol{A}. \end{aligned} \tag{15}$$

where τ is a learnable parameter used to adjust the range of similarity values, and $\mathrm{Sim}_{\cos}(\cdot,\cdot)$ denotes the calculation of cosine similarity between two tokens. The output of the DCAB is divided into two parts: $\mathbf{S} = \boldsymbol{A} \cdot \boldsymbol{V}_D$, which represents the auxiliary information obtained from the token dictionary, used for the iterative update of each query point, and $\boldsymbol{A}$, the attention map calculated, which is used for the subsequent adaptive refinement of the token dictionary.

Diffusion Iteration Layer. As shown in Figure 2(b), the DIL serves as the core module of our method, comprising the Diffusion Implicit Block (DIB) and the Dictionary Cross-Attention Block (DCAB). The first DIL layer takes two inputs: (1) features modulated by the Position-Aware Embedding Block, which integrates coordinate information and latent codes, and (2) a token dictionary adaptively refined through cross-attention with the encoder's latent codes. For subsequent layers, the input is the output of the previous layer.

Specifically, $\mathbf{Z}^{(l-1)}$ is obtained by combining the DIB output $\boldsymbol{P}$ and the DCAB output $\boldsymbol{S}$ through addition, followed by a 1×1 convolution. Finally, the learned parameter α_1 is applied to adaptively combine the results, producing $\mathbf{Z}^{(l)}$:

$$\mathbf{Z}^{(l)} = \mathrm{Conv}(\boldsymbol{P} + \boldsymbol{S}) + \alpha \mathbf{Z}^{(l-1)}, \tag{16}$$

To adaptively refine the token dictionary, we select the corresponding enhanced feature $\mathbf{Z}^{(l)}$ to reconstruct the new token dictionary $\mathbf{D}^{(l)}$:

$$\begin{aligned} \hat{\boldsymbol{D}}^{(l)} &= \mathrm{SoftMax}(\mathrm{Norm}(\boldsymbol{A}^{(l)^T}))\boldsymbol{X}^{(l+1)}, \\ D^{(l)} &= \sigma \hat{D}^{(l)} + (1-\sigma) D^{(l-1)}, \end{aligned} \tag{17}$$

where Norm is normalization layer to adjust the range of attention map, and σ is a learnable parameter.

4 Experiments

4.1 Experiments Setting

Datasets. Similar to [Chen *et al.*, 2021; Wei and Zhang, 2023], we use 800 high-quality images with a 2K resolution from the DIV2K [Agustsson and Timofte, 2017] dataset as training data. During testing, the model is evaluated on the DIV2K validation set and several commonly used benchmark datasets, including Set5 [Bevilacqua *et al.*, 2012], Set14 [Zeyde *et al.*, 2010], B100 [Martin *et al.*, 2001] and Urban100 [Huang *et al.*, 2015]. Following previous works [Chen *et al.*, 2021; Wei and Zhang, 2023], we evaluated HR image quality using PSNR, calculated on the RGB channels for DIV2K and on the Y channel (YCbCr space) for other benchmarks.

Implementation Details. We primarily follow the previous implementation [Chen *et al.*, 2021; Wei and Zhang, 2023], using Bicubic downsampling [Boor, 1962] in PyTorch to obtain paired data for training an arbitrary-scale super-resolution model. Specifically, we crop $128s \times 128s$ patches as ground truth (GT), where s is a scaling factor sampled from the uniform distribution $U(1, 4)$. We use the existing SR models, such as EDSR [Lim *et al.*, 2017] and RDN [Zhang *et al.*, 2018], as backbones without their upsampling modules to evaluate various arbitrary-scale upsampling methods. We

Encoder	Database	Set5						Set14					
	Method	In-distribution			Out-of-distribution			In-distribution			Out-of-distribution		
		×2	×3	×4	×6	×8	×12	×2	×3	×4	×6	×8	×12
RDN	RDN only	38.24	34.71	32.47	-	-	-	34.01	30.57	28.81	-	-	-
	MetaSR	38.22	34.63	32.38	29.04	26.96	-	33.98	30.54	28.78	26.51	24.97	-
	LIIF	38.17	34.68	32.50	29.15	27.14	24.86	33.97	30.53	28.80	26.64	25.15	23.24
	LTE	38.23	34.72	32.61	29.32	27.26	24.79	34.09	30.58	28.88	26.71	25.16	23.31
	CLIT	38.26	34.79	32.69	29.39	27.34	-	34.21	30.66	28.98	26.83	25.35	-
	SRNO	38.32	34.84	32.69	29.38	27.28	-	34.27	30.71	28.97	26.76	25.26	-
	DIIN (ours)	38.30	34.83	32.72	29.40	27.32	24.89	34.26	30.72	28.98	26.78	25.30	23.36
RDN	Database	B100						Urban100					
	RDN only	32.34	29.26	27.72	-	-	-	32.89	28.80	26.61	-	-	-
	MetaSR	32.33	29.26	27.71	25.90	24.97	-	32.92	28.82	26.55	23.99	22.59	-
	LIIF	32.32	29.26	27.74	25.98	24.91	23.57	32.87	28.82	26.68	24.20	22.79	21.15
	LTE	32.36	29.30	27.77	26.01	24.95	23.60	33.04	28.97	26.81	24.28	22.88	21.22
	CLIT	32.39	29.33	27.80	26.07	25.00	-	33.14	29.05	26.93	24.44	23.04	-
	SRNO	32.43	29.37	27.83	26.04	24.99	-	33.33	29.14	26.98	24.43	23.02	-
	DIIN (ours)	32.45	29.38	27.82	26.08	25.03	23.64	33.34	29.17	26.99	24.45	23.05	21.25

Table 2: Quantitative comparison with state-of-the-art methods for arbitrary-scale super-resolution on the Set5, Set14, B100 and Urban100 (PSNR in dB). In each column, the best result is highlighted in red, while the second-best result is highlighted in blue. '-' indicates that the result is unavailable in the literature or the model's source code has not been released.

	In-distribution			Out-of-distribution		
	×2	×3	×4	×6	×12	×18
DIIN	34.87	31.13	29.18	26.92	23.86	22.31
DIIN (-c)	34.81	31.07	29.13	26.84	23.79	22.25
DIIN (-p)	34.84	31.10	29.15	26.88	23.83	22.29
DIIN (-a)	34.79	31.05	29.11	26.83	23.78	22.24
DIIN (-d)	34.73	31.02	29.07	26.80	23.73	22.19

Table 3: Ablation study on TD and PAEB. -c/p/a refers to removing the token dictionary, the position-aware embedding block, and both, respectively. -d indicates using an identity matrix for the diffusion rate, replacing the Diffusion Iteration Layer with a linear layer.

Layers N	In-distribution			Params (M)	Inference time (s)
	×2	×3	×4		
$N=1$	34.79	31.05	29.09	2.2	1.19
$N=2$	34.87	31.13	29.18	2.7	1.94
$N=3$	34.88	31.14	29.19	3.2	2.71
$N=4$	34.88	31.14	29.20	3.7	3.45

Table 4: Comparison of performance gain, parameters and inference times of different depths.The inference time refers to the average duration required by the model to process the DIV2K validation set on an NVIDIA RTX 3090 GPU.

train all models with the Adam optimizer [Kingma and Ba, 2015], starting from an initial learning rate of 4×10^{-5} and minimizing the L_1 loss for 1500 epochs using a batch size of 32. The learning rate is updated by a cosine-annealing schedule every 50 epochs. All experiments are implemented in PyTorch [Paszke *et al.*, 2019] and executed on four NVIDIA RTX 3090 GPUs.

4.2 Comparisons With State-of-the-Art

Quantitative Results. Table 1-2 present a quantitative comparison of the proposed DIIN method with existing arbitrary-scale SR methods, including MetaSR [Hu *et al.*, 2019], LIIF [Chen *et al.*, 2021], LTE [Lee and Jin, 2022], CLIT [Chen *et al.*, 2023], and SRNO [Wei and Zhang, 2023]. We evaluate the performance of EDSR [Lim *et al.*, 2017] and RDN[Zhang *et al.*, 2018] as backbones across five test datasets, considering upsampling factors ranging from ×2 to ×30. Table 1 summarizes the quantitative results in terms of PSNR (dB) on the DIV2k dataset. It can be observed that our DIIN model consistently delivers outstanding super-resolution performance across all scaling factors, with up to a 0.05dB improvement over the second-best performer. Notably, our method demonstrates significant performance gains at the ×30 scaling factor, even surpassing some methods that use RDN as the encoder. Table 2 compares DIIN with prior works [Chen *et al.*, 2021; Wei and Zhang, 2023] on widely-used benchmark datasets, including Set5 [Bevilacqua *et al.*, 2012], Set14 [Zeyde *et al.*, 2010], B100 [Martin *et al.*, 2001], and Urban100 [Huang *et al.*, 2015], using RDN as the backbone. We observe that DIIN consistently achieves either the best or second-best performance across these benchmark datasets, further demonstrating the effectiveness of our method.

Qualitative Results. In Figure 3, we present a qualitative comparison with other arbitrary-scale super-resolution (SR) methods. Our model demonstrates the ability to generate SR images with sharper and more coherent textures, particularly excelling in areas with regular patterns. In the first-row example, our approach effectively restores the smoothness

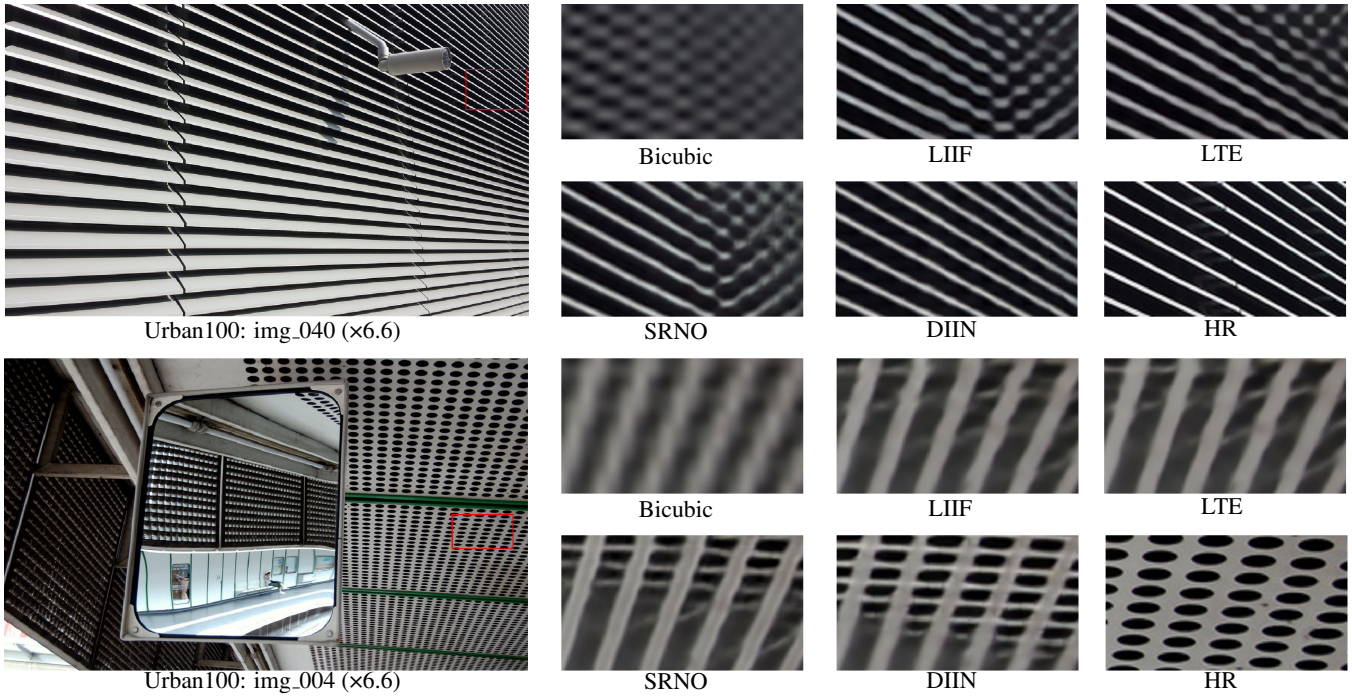


Figure 3: Comparison of super-resolution visual effects using different models. In challenging restoration regions, such as densely packed lines and areas with holes, our method demonstrates a clear advantage over previous approaches, achieving superior detail recovery.

and sharpness of lines while avoiding jagged artifacts. Additionally, it achieves more accurate recovery of hole edges and successfully suppresses artifacts in complex reflective regions. More visual results from the test set are provided in the supplementary material. Additional qualitative results can be found in the supplementary materials.

4.3 Ablation Study

In this section, we present a series of ablation studies to validate the design decisions proposed in this paper. All ablation experiments are conducted on the DIV2K validation set, using EDSR [Lim *et al.*, 2017] as the encoder baseline, and evaluated using the PSNR metric.

Validation of the Design Choices. To validate the effectiveness of each component in the proposed DIIN model, we constructed three models and compared their performance on image super-resolution tasks. Table 3 summarizes the quantitative contributions of these components. The results show that incorporating the Diffusion Iteration Layer significantly improves performance. Removing the Token Dictionary leads to some performance degradation, indicating its role in effectively supplementing information. Additionally, replacing the Position-Aware Embedding Block with a simple 1×1 convolution results in noticeable performance degradation, further demonstrating that using coordinate information to modulate features enhances spatial dependencies among input samples, thereby improving overall performance.

Effectiveness of Different Designs of Diffusion Iteration Layer. We evaluated the model's performance, parameter count, and inference time at different depths (number of layers N) to analyze the impact of iterative layers on the performance of the Diffusion Iteration Layer. As shown in Table 4, increasing the model depth leads to improved performance (PSNR) across different scaling factors ($\times 2, \times 3, \times 4$), with particularly significant gains observed for smaller scaling factors ($\times 2$ and $\times 3$). However, the marginal performance gains diminish as the depth further increases, while both the parameter count and inference time grow substantially. Balancing performance and computational cost, we ultimately selected $N = 2$ as the hyperparameter setting for our method.

5 Conclusion

In this paper, we rethink the modeling of implicit neural functions by introducing neighborhood interactions and redefining the forward process as a signal diffusion mechanism. To achieve this, we propose Diffusion Iterative Implicit Networks (DIIN), which efficiently promotes global signal flow with linear complexity. Key components such as the Dictionary Cross-Attention Block and the Position-Aware Embedding Block enhance contextual awareness and spatial dependencies, respectively. Experiments on benchmark datasets demonstrate that DIIN achieves state-of-the-art performance in arbitrary-scale super-resolution tasks, with significant improvements in detail restoration and contextual modeling. Ablation studies confirm the effectiveness of each module in the proposed framework. Looking ahead, DIIN could be extended to tasks like 3D reconstruction or video super-resolution, while further optimization may enhance its practicality in real-time applications. Our work establishes a promising direction for advancing implicit neural representation techniques.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China, under Grant (62302309, 62471310), Shenzhen Science and Technology Program JCYJ20220818101014030, and OPPO Research Fund.

References

[Agustsson and Timofte, 2017] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 1122–1131, 2017.

[Bevilacqua *et al.*, 2012] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on non-negative neighbor embedding. In *BMVC*, pages 1–10, 2012.

[Boor, 1962] Carl De Boor. Bicubic spline interpolation. *Journal of Mathematics and Physics*, 41(1-4):212–218, 1962.

[Chen *et al.*, 2021] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, pages 8628–8638, 2021.

[Chen *et al.*, 2023] Hao-Wei Chen, Yu-Syuan Xu, Min-Fong Hong, Yi-Min Tsai, Hsien-Kai Kuo, and Chun-Yi Lee. Cascaded local implicit transformer for arbitrary-scale super-resolution. In *CVPR*, pages 18257–18267, 2023.

[Dai *et al.*, 2019] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, pages 11065–11074, 2019.

[Dai *et al.*, 2024] Tao Dai, Jianping Wang, Hang Guo, Jinmin Li, Jinbao Wang, and Zexuan Zhu. Freqformer: frequency-aware transformer for lightweight image super-resolution. In *IJCAI*, pages 731–739, 2024.

[Dong *et al.*, 2015] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2015.

[Gao *et al.*, 2022] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *CVPR*, pages 10598–10608, 2022.

[Gu and Dong, 2021] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *CVPR*, pages 9199–9208, 2021.

[Guo *et al.*, 2024a] Hang Guo, Tao Dai, Yuanchao Bai, Bin Chen, Xudong Ren, Zexuan Zhu, and Shu-Tao Xia. Parameter efficient adaptation for image restoration with heterogeneous mixture-of-experts. *NeurIPS*, 37:13522–13547, 2024.

[Guo *et al.*, 2024b] Hang Guo, Tao Dai, Zhihao Ouyang, Taolin Zhang, Yaohua Zha, Bin Chen, and Shu-tao Xia. Refir: Grounding large restoration models with retrieval augmentation. *NeurIPS*, 37:46593–46621, 2024.

[Guo *et al.*, 2024c] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. *arXiv preprint arXiv:2411.15269*, 2024.

[Guo *et al.*, 2024d] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *ECCV*, 2024.

[Hu *et al.*, 2019] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, pages 1575–1584, 2019.

[Huang *et al.*, 2015] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015.

[Kim *et al.*, 2016] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016.

[Kingma and Ba, 2015] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Lee and Jin, 2022] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *CVPR*, pages 1929–1938, 2022.

[Liang *et al.*, 2021] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCV*, pages 1833–1844, 2021.

[Lim *et al.*, 2017] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017.

[Lin *et al.*, 2024] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *ECCV*, pages 430–448. Springer, 2024.

[Martin *et al.*, 2001] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[Mildenhall *et al.*, 2020] 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 *ECCV*, pages 405–421. Springer, 2020.

[Park *et al.*, 2019] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove.

DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, and etc. James Bradbury. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 8024–8035, 2019.

[Ren *et al.*, 2024] Yulin Ren, Xin Li, Mengxi Guo, Bingchen Li, Shijie Zhao, and Zhibo Chen. Mambacsr: Dual-interleaved scanning for compressed image super-resolution with ssms. *arXiv preprint arXiv:2408.11758*, 2024.

[Saharia *et al.*, 2021] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv:2104.07636*, 2021.

[Sitzmann *et al.*, 2020] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. pages 7462–7473, 2020.

[Tancik *et al.*, 2020] 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. pages 7537–7547, 2020.

[Wang *et al.*, 2024a] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *IJCV*, 132(12):5929–5949, 2024.

[Wang *et al.*, 2024b] Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *CVPR*, pages 25796–25805, 2024.

[Wei and Zhang, 2023] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *CVPR*, pages 18247–18256, June 2023.

[Wu *et al.*, 2023] Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. Difformer: Scalable (graph) transformers induced by energy constrained diffusion. In *ICLR*, 2023.

[Wu *et al.*, 2024] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *CVPR*, pages 25456–25467, 2024.

[Xia *et al.*, 2023] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. *ICCV*, 2023.

[Xie *et al.*, 2023] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. Diner: Disorder-invariant implicit neural representation. In *CVPR*, 2023.

[Zeyde *et al.*, 2010] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, volume 6920 of *Lecture Notes in Computer Science*, pages 711–730, 2010.

[Zhang *et al.*, 2018] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018.

[Zhang *et al.*, 2024] Leheng Zhang, Yawei Li, Xingyu Zhou, Xiaorui Zhao, and Shuhang Gu. Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary. In *CVPR*, pages 2856–2865, June 2024.

[Zhou *et al.*, 2023] Yupeng Zhou, Zhen Li, Chun-Le Guo, Song Bai, Ming-Ming Cheng, and Qibin Hou. Srformer: Permuted self-attention for single image super-resolution. In *ICCV*, pages 12780–12791, 2023.