Learning Coupled Dictionaries from Unpaired Data for Image Super-Resolution

Longguang Wang[1], Juncheng Li[2], Yingqian Wang[3], Qingyong Hu, Yulan Guo[4*]
[1]Aviation University of Air Force [2]Shanghai University
[3]National University of Defense Technology [4]Sun Yat-sen University

Abstract

*The difficulty of acquiring high-resolution (HR) and low-resolution (LR) image pairs in real scenarios limits the performance of existing learning-based image super-resolution (SR) methods in the real world. To conduct training on real-world unpaired data, current methods focus on synthesizing pseudo LR images to associate unpaired images. However, the realness and diversity of pseudo LR images are vulnerable due to the large image space. In this paper, we circumvent the difficulty of image generation and propose an alternative to build the connection between unpaired images in a compact proxy space. Specifically, we first construct coupled HR and LR dictionaries, and then encode HR and LR images into a common latent code space using these dictionaries. In addition, we develop an autoencoder-based framework to couple these dictionaries during optimization by reconstructing input HR and LR images. The coupled dictionaries enable our method to employ a shallow network architecture with only 18 layers to achieve efficient image SR. Extensive experiments show that our method (**DictSR**) can effectively model the LR-to-HR mapping in coupled dictionaries and produces state-of-the-art performance on benchmark datasets.*

1. Introduction

Image super-resolution (SR) aims at reconstructing a high-resolution (HR) image from low-resolution (LR) observations. To this end, most existing learning-based methods [1–4] rely on paired training data to build an LR-to-HR mapping. Since paired data is difficult to acquire in the real world, manually designed degradations are commonly employed to synthesize paired data for training [5, 6]. Nevertheless, as pre-defined degradations are empirical, synthetic images cannot meet the diversity of real-world images and limit the performance of previous methods in real scenarios [7, 8].

To remedy this problem, numerous efforts have been made to directly conduct training on unpaired data [7–9]. To enable the learning of LR-to-HR mapping, existing methods focus on synthesizing pseudo paired data for training. Given HR images $x \sim p_x$ (target domain) and LR images $y \sim p_y$ (source domain), pseudo LR images are generated by learning an HR-to-LR mapping. Then, HR images and the resultant pseudo LR images are adopted to learn the LR-to-HR mapping. Early methods [7, 9] commonly use generative adversarial networks (GANs) with cycle-consistency or adversarial loss to synthesize pseudo LR images. However, these GAN-based methods usually suffer mode collapse and require elaborate fine-tuning. Recently, more powerful diffusion model [10] is introduced for pseudo LR image synthesis. The large latent space in the diffusion model improves the diversity of synthesized images at the cost of lower content fidelity, which hinders further performance improvements.

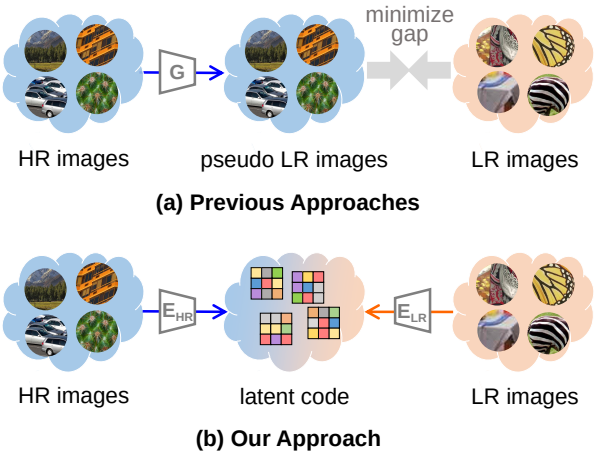

Figure 1. Comparison between previous approaches and our approach. Instead of synthesizing pseudo LR images and minimizing the gap in the image space (a), we encode HR and LR images into a common latent code space using coupled dictionaries to associate unpaired HR and LR images (b).

Due to the low rank characteristic of natural images, an image can be represented as a spare code on an over-complete dictionary [11, 12]. On top of this perspective, we seek to associate unpaired LR and HR images in a compact space without relying on synthesizing pseudo LR im-

ages high in diversity and realness (Fig. 1). By representing images using spatial codes over a dictionary, the description length is significantly reduced, which allows us to efficiently minimize the gap between HR and LR image domains. To this end, we use autoencoders to learn LR and HR dictionaries rich of context information from LR and HR images. To fit the large space of real-world LR patches, our LR dictionary explicitly models the content and degradation variations in a hierarchical manner (Fig. 2). During training, by coupling HR and LR dictionaries, an LR-to-HR mapping can be learned in the dictionaries. During inference, an LR image is first decomposed over the LR dictionary and then using the HR dictionary for reconstruction to produce the SR result. The contributions of this paper can be summarized as follows:

- We associate unpaired images using coupled HR and LR dictionaries instead of synthesizing pseudo LR images. By decomposing unpaired images over the dictionary to obtain compact proxy codes, an LR-to-HR mapping can be efficiently learned in the code space.
- We formulate the LR dictionary as a combination of an LR content dictionary and a degradation dictionary to model LR images in a hierarchical manner.
- We develop an autoencoder-based framework to optimize coupled dictionaries together with the network parameters in an end-to-end manner.
- Extensive experiments show that our coupled dictionaries can effectively learn the LR-to-HR mapping and our method (**DictSR**) achieves state-of-the-art performance on benchmark datasets.

2. Related Works

In this section, we first review recent advances of paired and unpaired image SR methods. Then, we discuss dictionary learning approach that is related to our work.

2.1. Paired Image SR

Most previous learning-based image SR methods rely on paired data with aligned contents for training. Early methods [1–3, 13–15] commonly employ bicubic downsampling to synthesize HR-LR image pairs as the training data. Despite promising results on synthetic data, these methods suffer severe performance drop on real-world images since their degradations differ from the bicubic one. Later, several methods adopt more complicated degradations including blur, compression, and noise to generate HR-LR image pairs [5, 16–19]. Although the handcrafted degradations span a large space, the gap between real-world degradations still remains and limits the accuracy in real scenarios. Recently, a number of real-world paired datasets are developed to capture real degradations [20, 21]. However, the high labor cost limits the scales of these datasets, thereby hindering further performance improvement.

2.2. Unpaired Image SR

To circumvents the difficulty of acquiring real-world paired data, a number of methods directly conduct training on unpaired images. Early approaches assume real-world LR images follow a specific distribution and employ GANs to synthesize pseudo LR image in this distribution. Specifically, Bulat *et al.* [22] and Lugmayr *et al.* [23] first trained a degradation network to synthesize pseudo LR images from HR images and then use these paired images to learn the LR-to-HR mapping. Yuan *et al.* [9] and Maeda *et al.* [7] developed unified frameworks to simultaneously learn a degradation network and an SR network, which produce superior accuracy. Liu *et al.* [24] and Yang *et al.* [25] introduced physical properties to LR image synthesis as regularizions for higher realness. Despite improved performance against paired approaches, these GAN-based methods usually suffer mode collapse and cannot generate LR images as diverse as real ones.

To address the aforementioned problem of GAN-based methods, Wolf *et al.* [26] developed a flow-based model to synthesize LR images. Although the diversity of synthesized LR images is improved, this method has a high computational cost due to the limited expressive power of flow-based models [27, 28]. Recently, Yang *et al.* [10] introduced the powerful diffusion model for pseudo LR image synthesis. The large latent space in the diffusion model improves the diversity but decreases the content fidelity, which also results in limited performance.

2.3. Dictionary Learning

Dictionary learning aims at finding a sparse representation of the input data in the form of a linear combination of basic elements as well as those basic elements themselves. The idea of dictionary learning can be dated back to several decades ago [29, 30]. Before the era of deep learning, dictionary learning is widely used to model the LR-to-HR mapping for image SR. Specifically, Yang *et al.* [12] proposed coupled dictionaries to associate LR and HR image patches via sparse coding. Zhang *et al.* [11] developed a multi-scale dictionary to simultaneously model local and non-local priors in the images. Lu *et al.* [31] took the geometrical structure of the dictionary into consideration and proposed a geometry constrained sparse coding method for image SR.

Since the huge success of neural networks, learning-based methods have dominated a wide range of tasks. Recently, the idea of dictionary learning has been introduced to learning-based frameworks and produces superior performance. Specifically, Van *et al.* [32] proposed a vector quantized variational autoencoder (VQ-VAE) to learn discrete representations of an image over a codebook. Then, a CNN is trained to model their distribution for image synthesis. Then, Razavi *et al.* [33] extended this approach to a

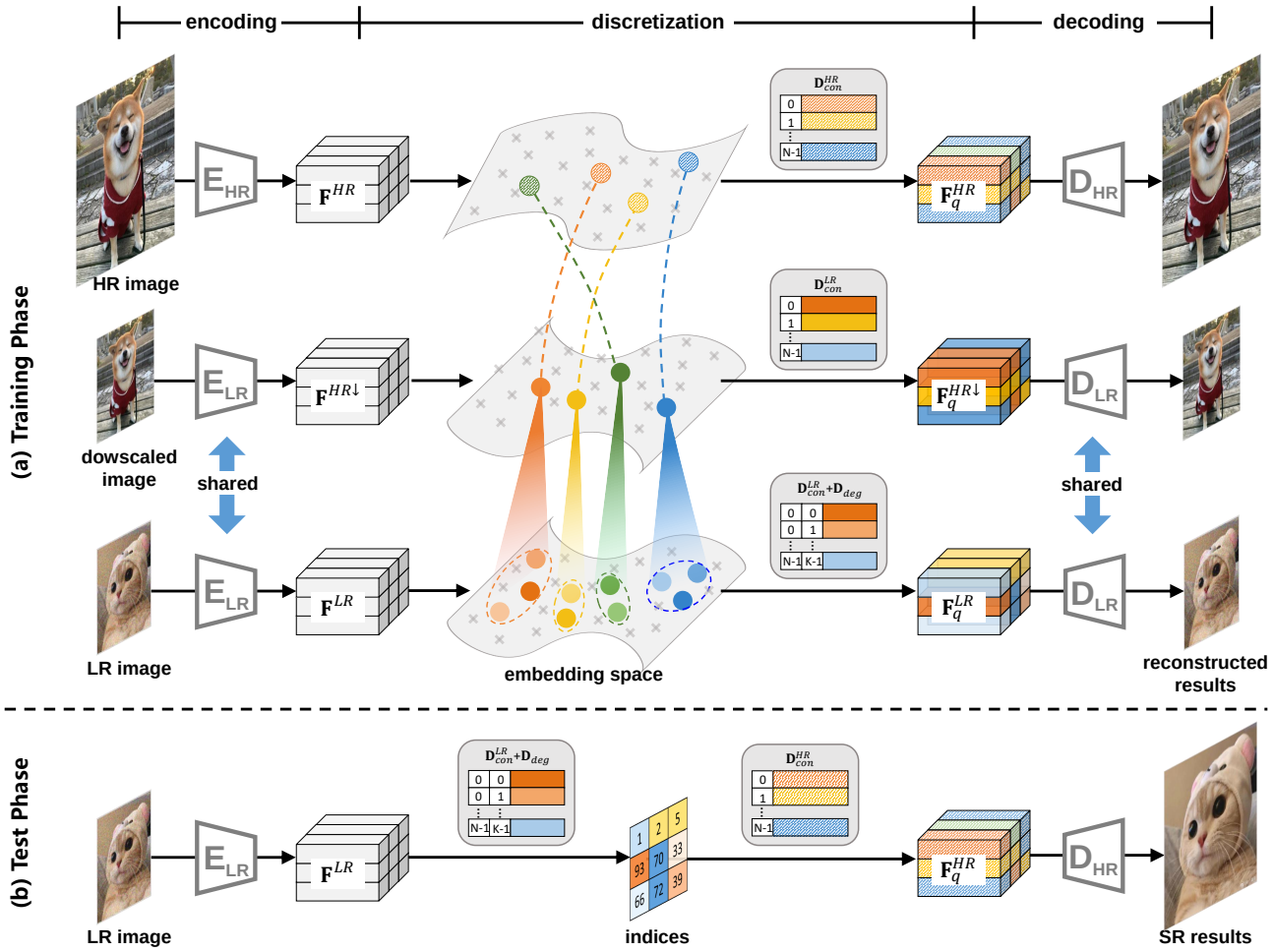

Figure 2. An illustration of our framework during training phase (a) and test phase (b). Entries with different colors correspond to various contents while entries with different shades refer to the same content but diverse degradations.

hierarchical one to learn discrete representation at different scale, which achieves superior performance. Later, Esser *et al.* [34] further developed VQ-GAN by leveraging the powerful transformer to model the composition of images. Recently, Maeda *et al.* [35] introduced dictionary learning for image SR by plugging a learnable dictionary into a CNN framework. However, this method relies on paired data to learn the LR-to-HR mapping and cannot be extended to unpaired SR.

3. Methodology

Ideally, a pair of LR and HR images can be decomposed to a common latent code over coupled LR-HR dictionaries. From this point of view, we seek to learn coupled dictionaries from unpaired data for image SR. To this end, we formulate the coupled dictionaries as a combination of an HR content dictionary $\mathbf{D}^{HR}$, an LR content dictionary $\mathbf{D}^{LR}$, and a degradation dictionary $\mathbf{D}_{deg}$. These three dictionaries are tightly coupled and jointly optimized in three branches, as illustrated in Fig. 2. During the training phase, the pipeline for each branch consists of three stages, including encoding, descretization, and decoding. Specifically, input images are first encoded into embeddings. Then, these embeddings are discretized in the embedding space to the entries in corresponding dictionaries. Finally, the discretized embeddings are fed to the decoders to reconstruct the input image. During the test phase, the input LR image is first encoded to an indicex map using $\mathbf{D}^{LR}$ and $\mathbf{D}_{deg}$. Then, the corresponding entries in $\mathbf{D}^{HR}$ is retrieved to reconstruct the SR result.

3.1. Encoding Stage

Given an HR image $\mathbf{I}^{HR} \in \mathbb{R}^{H\times W\times 3}$ and a unpaired LR image $\mathbf{I}^{LR} \in \mathbb{R}^{h\times w\times 3}$, $\mathbf{I}^{HR}$ is first downscaled to obtain $\mathbf{I}^{HR\downarrow} \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times 3}$. Here, s is the scale factor and set to

4 as we focus on $\times 4$ SR in this paper. Then, $\mathbf{I}^{HR}$ is fed to the HR image encoder to produce $\mathbf{F}^{HR} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 512}$, with each embedding corresponding to a 8×8 patch in the input HR image. Meanwhile, $\mathbf{I}^{LR}$ and $\mathbf{I}^{HR\downarrow}$ are passed to the LR image encoder to obtain $\mathbf{F}^{LR} \in \mathbb{R}^{\frac{h}{2} \times \frac{w}{2} \times 128}$ and $\mathbf{F}^{HR\downarrow} \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 128}$, respectively. Each embedding in the resultant feature maps corresponds to a 2×2 patch in the input LR image.

The HR image encoder employs a fully convolutional structure and consists of four residual blocks at four resolution levels (*i.e.*, $H \times W$, $\frac{H}{2} \times \frac{W}{2}$, $\frac{H}{4} \times \frac{W}{4}$, and $\frac{H}{8} \times \frac{W}{8}$). Meanwhile, the LR image encoder consists of two residual blocks at $h \times w$ and $\frac{h}{2} \times \frac{w}{2}$ resolution levels.

3.2. Discretization Stage

After the encoding stage, the resultant embedding at each location of the feature map is discretized to its closest dictionary entry in the embedding space, resulting in $\mathbf{F}_q^{HR}$, $\mathbf{F}_q^{HR\downarrow}$, and $\mathbf{F}_q^{LR}$. Motivated by [36], we introduce a linear projection to map the output feature maps to a low-dimensional latent space (4-dim in our experiments) for code index lookup. Then, the discretized embedding is projected back to the embedding space through another projection layer. During backpropagation, straight-through estimator (STE) is employed for end-to-end optimization.

To associate HR and LR images, HR and LR dictionaries should be coupled at this stage. However, there exists two major challenges: **(1) Misaligned Content.** Under paired settings, corresponding patches in HR and LR images share the same content, which can be employed as a cue to couple HR and LR dictionaries for joint optimization. However, this cue does not hold for unpaired images, which imposes great challenges to learning coupled dictionaries. **(2) Diverse Degradations.** Since an HR image corresponds to infinite LR images under various degradations, modeling such a one-to-many correlation using coupled dictionaries is quite difficult. To address the above challenges, we formulate the coupled dictionaries as a combination of an HR content dictionary, an LR content dictionary, and a degradation dictionary. The LR content dictionary couples with the HR content dictionary to build connections between LR and HR image contents. Meanwhile, the degradation dictionary captures the degradation variations for different image contents.

3.2.1 HR and LR Content Dictionary

First, we construct an HR content dictionary $\mathbf{D}^{HR} \in \mathbb{R}^{N \times 4}$ and an LR content dictionary $\mathbf{D}^{LR} \in \mathbb{R}^{N \times 4}$ to model the content variation in the embedding space, where N is the number of entries. To handle the first challenge, we introduce downscaled version of HR images as a bridge to associate HR and LR content dictionaries to enable joint training. For embeddings $f^{HR} \in \mathbf{F}^{HR}$ (branch 1) and $f^{HR\downarrow} \in \mathbf{F}^{HR\downarrow}$ (branch 2), the discretized embeddings f_q^{HR} and $f_q^{HR\downarrow}$ are obtained as:

$$i = \arg\min_i ||f^{HR\downarrow} - \mathbf{D}^{LR}(i)||, \tag{1}$$

$$\begin{cases} f_q^{HR\downarrow} = \mathbf{D}^{LR}(i) \\ f_q^{HR} = \mathbf{D}^{HR}(i) \end{cases}, \tag{2}$$

where i is the index of the closest entry in $\mathbf{D}^{LR}$ to the embedding $f^{HR\downarrow}$. By employing the same spatial codes (*i.e.*, i) to represent both f^{HR} and $f^{HR\downarrow}$, HR and LR content dictionaries are coupled. Inspired by [34], the discretization losses for branches 1 and 2 are defined as:

$$\begin{cases} \mathcal{L}_{dis}^1 = ||\mathbf{F}^{HR} - \mathbf{F}_q^{HR}||_2^2 \\ \mathcal{L}_{dis}^2 = ||\mathbf{F}^{HR\downarrow} - \mathbf{F}_q^{HR\downarrow}||_2^2 \end{cases}. \tag{3}$$

3.2.2 Degradation Dictionary

Second, we construct a degradation dictionary $\mathbf{D}_{deg} \in \mathbb{R}^{N \times K \times 4}$ on top of $\mathbf{D}^{LR}$ to model the degradation variations. Specifically, each group of entries (*e.g.*, $\mathbf{D}_{deg}(i) \in \mathbb{R}^{K \times 4}$) is used to capture the degradation variation around the corresponding entry in the content dictionary (*i.e.*, $\mathbf{D}^{LR}(i)$). During discretization, a hierarchical approach is employed to obtain the discretized embedding f_q^{LR}:

$$\begin{cases} j = \arg\min_j ||f^{LR} - \mathbf{D}^{LR}(j)|| \\ k = \arg\min_k ||f^{LR} - (\mathbf{D}^{LR}(j) + \mathbf{D}_{deg}(j,\ k))|| \end{cases}. \tag{4}$$

For each embedding f^{LR} in $\mathbf{F}^{LR}$ (branch 3), the closest entry in $\mathbf{D}^{LR}$ is first selected. $\mathbf{D}^{LR}(j)$ depicts the clean image content in f^{LR} but ignores the degradation. Then, the degradation is taken into consideration by retrieving the closest entry in $\mathbf{D}^{LR}(j) + \mathbf{D}_{deg}(j)$, resulting in f_q^{LR}. Similarly, a discretization loss is defined as:

$$\mathcal{L}_{dis}^3 = ||\mathbf{F}^{LR} - \mathbf{F}_q^{LR}||_2^2. \tag{5}$$

3.3. Decoding Stage

After the discretization stage, the feature map produced by the encoders are represented using the dictionaries. Then, $\mathbf{F}_q^{HR}$ is fed to the HR image decoder to reconstruct the HR image $\hat{\mathbf{I}}^{HR}$. Meanwhile, $\mathbf{F}_q^{HR\downarrow}$ and $\mathbf{F}_q^{LR}$ are passed to the LR image decoder to produce $\hat{\mathbf{I}}^{HR\downarrow}$ and $\hat{\mathbf{I}}^{LR}$, respectively. The reconstruction loss is defined as the L1 loss between the reconstructed results and the input images:

$$\begin{aligned} \mathcal{L}_{rec} = &||\hat{\mathbf{I}}^{HR} - \mathbf{I}^{HR}|| + ||\phi(\hat{\mathbf{I}}^{HR}) - \phi(\mathbf{I}^{HR})|| + \\ &||\hat{\mathbf{I}}^{HR\downarrow} - \mathbf{I}^{HR\downarrow}|| + ||\hat{\mathbf{I}}^{LR} - \mathbf{I}^{LR}||, \end{aligned} \tag{6}$$

where $\phi(\mathbf{I}^{HR})$ refers to the feature maps extracted by a pre-trained VGG model [37].

Similar with the encoders, the HR image decoder is composed of four residual blocks at different resolution levels and a tail convolutional layer to regress the reconstructed images. Meanwhile, the LR image decoder comprises of two residual blocks and a tail convolutional layer.

3.4. Loss Function

The overall loss function used for training is defined as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \times (\mathcal{L}^1_{dis} + \mathcal{L}^2_{dis} + \mathcal{L}^3_{dis}) \tag{7}$$

where λ_1 is empirically set to 10 in our experiments.

4. Experiments

In this section, we first introduce the experimental setup. Then, we compare our method with previous paired and unpaired image SR methods on benchmark datasets. Finally, we conduct experiments to investigate the effectiveness of our major designs.

4.1. Experimental Setup

4.1.1 Datasets and Metrics

We conduct experiments on two widely applied unpaired SR datasets, including AIM-RWSR [38] and NTIRE-RWSR [39]. Note that, we focus on $\times 4$ SR in this paper and our method can be extended to SR tasks with different scale factors.

AIM-RWSR. The AIM-RWSR dataset is a synthetic dataset and uses handcrafted degradations to synthesize 2650 noisy and compressed LR images. Meanwhile, 800 images in the DIV2K dataset are employed as HR images.

NTIRE-RWSR. The NTIRE-RWSR dataset follows the same setting as the AIM-RWSR dataset to produce unpaired HR and LR images. Different from AIM-RWSR, NTIRE-RWSR features a more complicated degradation type that consists of highly related high-frequency noises.

For evaluation on the AIM-RWSR and NTIRE-RWSR datasets, we use peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and learned perceptual image patch similarity (LPIPS) as metrics.

4.1.2 Training Details

During the training phase, 24 LR patches of size 64×64 and 24 HR patches of size 256×256 were randomly cropped from LR and HR images, respectively. Random rotation and random flipping were employed for data augmentation. The Adam method [40] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ was used for optimization. The learning rate was initialized as 2×10^{-4} and halved after every 25 epochs. The training was stopped after 100 epochs. In our experiments, the numbers of entries in the dictionaries (*i.e.*, N and K) were set to 4096 and 32, respectively.

Table 1. Results achieved on AIM-RWSR for $\times 4$ SR. Best and second best results are **highlighted** and underlined.

	Method	#Layers	MACs	LPIPS	PSNR	SSIM
Paired	Bicubic	-	-	0.537	22.35	0.617
	RCAN [3]	400	260G	0.472	22.32	0.604
	ZSSR [41]	8	-	0.639	22.21	0.603
	IKC [5]	188	80G	0.479	22.25	0.600
	DAN [42]	171	315G	0.471	22.41	0.609
	BSRGAN [17]	350	291G	0.299	**22.47**	**0.623**
	Real-ESRGAN [16]	350	291G	**0.238**	22.08	0.622
Unpaired	CycleGAN [38]	350	291G	0.476	21.19	0.530
	CinCGAN [9]	37	823G	0.461	21.60	0.613
	Lugmayr *et al.* [23]	350	291G	0.472	21.59	0.550
	FSSR [43]	350	291G	0.390	20.82	0.510
	DASR [8]	350	291G	0.336	21.60	0.564
	DeFlow [26]	350	291G	0.349	22.25	0.620
	PCR-ESRGAN [46]	350	291G	0.321	21.59	0.610
	DictSR (Ours)	18	129G	**0.259**	**22.46**	**0.629**

4.2. Performance Evaluation

For performance evaluation, six representative paired SR methods (RCAN [3], ZSSR [41], IKC [5], DAN [42], BSRGAN [17] and Real-ESRGAN [16]) and eight state-of-the-art unpaired image SR methods (CinCGAN [9], Lugmayr *et al.* [23], FSSR [43], Impressionism [44], DASR [8], DeFlow [26], DAP [45], and PCR-ESRGAN [46]) are included for comparison. Quantitative results are presented in Tables 1 and 2 while visual results are provided in Fig. 3. Note that, Lugmayr *et al.*, FSSR, DASR, DeFlow, Impressionism, and DAP employ ESRGAN as the SR model. For our DictSR, the LR image encoder and the HR image decoder are included to calculate the numbers of layers since only these modules are employed during inference. For other methods, the numbers of layers in the SR networks are presented. MACs (multiply-accumulate operations) is calculated based on 128×128 input LR images. The MACs of ZSSR is not reported since this method conducts training during the inference time and requires considerable computational cost.

Quantitative Results. For the AIM-RWSR dataset, it can be observed from Table 1 that our DictSR achieves the best performance in terms of all metrics among all unpaired SR methods. Moreover, our DictSR also produces competitive results as compared to paired SR methods. Previous unpaired SR methods synthesize pseudo LR images to associate unpaired LR and HR images. However, the large image space poses great challenges to these methods to balance the realness and diversity. In contrast, by building the connection between unpaired images in a compact

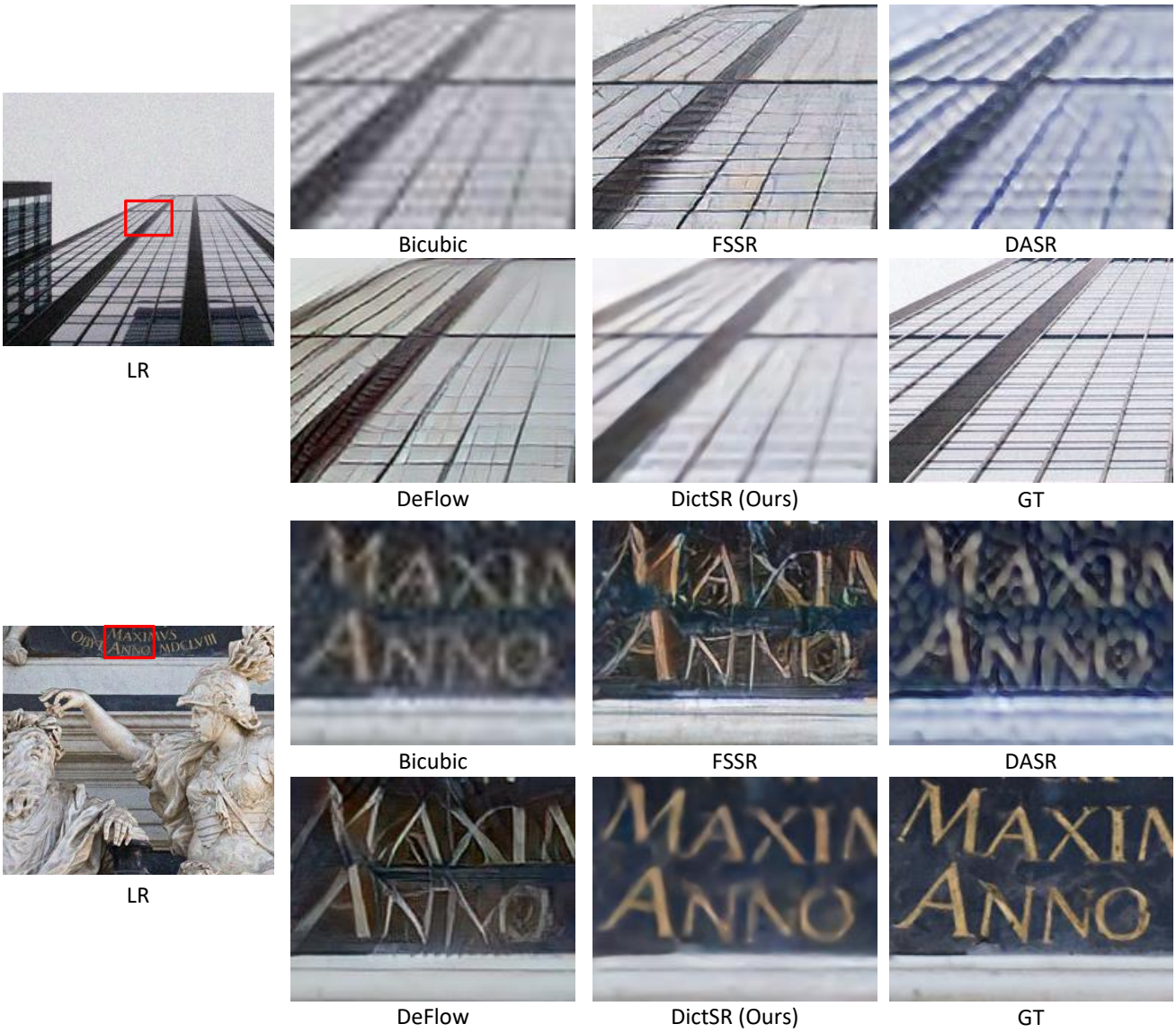

Figure 3. Visualization results produced by different methods.

proxy space using dictionaries, our DictSR produces superior quantitative results. In addition, our DictSR employs a shallow and efficient network structure with much lower computational complexity. Specifically, our method gradually reduces the spatial resolution of feature maps to $\frac{1}{4}$ size while previous methods maintain full-resolution feature maps. As compared to previous methods, our 18-layer network produces superior performance with only half MACs, which further demonstrates the effectiveness of our method.

For the NTIRE-RWSR dataset, we can observe from Table 2 that our DictSR also produces competitive results. As compared to DeFlow, although the improvements in terms of PSNR and SSIM are marginal, our method produces notable gains on LPIPS (0.204 vs. 0.218). This demonstrates the superior perceptual quality of our results. In addition, our DictSR also surpasses most previous paired methods with much a shallower network. This further shows the great potential of our learned coupled dictionaries.

Qualitative Results. Figure 3 compares the visual results produced by different unpaired image SR methods. It can be observed that our method produces higher perceptual quality with fewer artifacts. For example, in the second scene, FSSR and DeFlow produces results with unpleasant ringing artifacts while DASR suffer notable blurring artifacts. In contrast, the results of our DisctSR can better recover the text details. This further demonstrates the superiority of our method.

Table 2. Results achieved on NTIRE-RWSR for $\times 4$ SR. Best and second best results are **highlighted** and underlined.

	Method	#Layers	MACs	LPIPS	PSNR	SSIM
Paired	Bicubic	-	-	0.632	25.52	0.671
	RCAN [3]	400	260G	0.576	25.31	0.640
	ZSSR [41]	8	-	0.620	24.93	0.642
	IKC [5]	188	80G	0.384	**26.50**	**0.748**
	DAN [42]	171	315G	0.554	25.15	0.671
	BSRGAN [17]	350	291G	0.265	24.56	0.669
	Real-ESRGAN [16]	350	291G	**0.251**	24.68	0.687
Unpaired	CycleGAN [38]	350	291G	0.417	24.75	0.700
	CinCGAN [9]	37	823G	-	24.19	0.683
	Impressionism [44]	350	291G	0.232	24.67	0.683
	FSSR [43]	350	291G	0.332	23.04	0.590
	DeFlow [26]	350	291G	0.218	25.87	0.710
	DAP [45]	350	291G	0.252	25.40	0.707
	PCR-ESRGAN [46]	350	291G	0.223	24.97	0.682
	DictSR (Ours)	18	129G	**0.204**	**25.90**	**0.711**

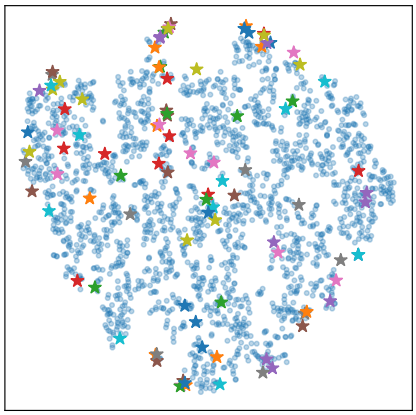

Figure 4. Visualization of the embedding space. Blue circles represent the embeddings of LR patches while other 100 stars correspond to 100 entries in the LR content dictionary $\mathbf{D}^{LR}$.

4.3. Model Analyses

In this subsection, we conduct experiments to study our method from three aspects. First, we study our dictionaries. Then, we demonstrate the effectiveness of our learning framework. Finally, we investigate the effect of different encoder and decoder architectures. All experiments are conducted on the AIM-RWSR dataset.

4.3.1 Dictionaries

(1) Hierarchical Structure

To cover diverse degradations in LR images while associating HR and LR image contents, a combination of an LR content dictionary $\mathbf{D}^{LR}$ and a degradation dictionary $\mathbf{D}_{deg}$ is employed. To demonstrate the effectiveness of such a hierarchical structure, we remove $\mathbf{D}_{deg}$ and directly use $\mathbf{D}^{LR}$ to discretize feature maps extracted from LR images (*i.e.*, A1 in Table 3). It can be observed that model A1 suffers an accuracy drop as compared to the full model (Ours in Table 3). Without the hierarchical structure, the LR content dictionary $\mathbf{D}^{LR}$ cannot well cover the large degradation space, thereby producing limited performance. In contrast, the hierarchical structure facilitates our method to achieve notable gains.

(2) Dictionary Size

Intuitively, larger dictionaries contribute to larger model capacity and ultimately result in higher performance. Consequently, we conduct experiments to study the effect of dictionary size from the following three aspects.

Entry. We first conduct experiments to investigate dictionaries with different numbers of entries. Specifically, we develop two network variants (B1/B2 in Table 3) by employing dictionaries with more/fewer entries. Using dictionaries with fewer entries (2048 entries), the performance of model B1 is slightly lower than our full model. However, further including more entries (8192 entries) cannot introduce consistent gains on all metrics. Consequently, we employ 4096 entries as the default setting.

Dimension. During the discretization stage, a linear projection is employed to map the feature maps to a low-dimensional latent space for code index lookup. Here, we conduct experiments to study the dimension of the latent space. As the dimension of the latent codes is reduced from 64 to 4, the number of activated entries is increased from 105 to 4050. This indicates that the dictionary can better cover the embeddings in the low-dimensional latent space to activate more entries. As a results, the LPIPS/PSNR/SSIM scores are improved from 0.268/21.99/0.616 to 0.259/22.46/0.629. In our experiments, the dimension of 4 is adopted as the default setting.

(3) Visualization of Dictionaries

To further study the embedding space of LR feature maps, we visualize the entries of the LR content dictionary $\mathbf{D}^{LR}$ and the embeddings of $\mathbf{F}^{LR}$ in Fig. 4. As we can see, the entries in $\mathbf{D}^{LR}$ span the space of embeddings extracted from LR patches. This validates that $\mathbf{D}^{LR}$ can well cover the content variation in the LR embedding space.

4.3.2 Learning Framework

(1) Encoder

Each entry in our LR/HR dictionaries corresponds a $2\times2/8\times8$ patch in the LR/HR image. We further conduct experiments to investigate the effect of the patch size. Specifically, we deepen the LR/HR encoders to encode larger patches into embeddings for discretization. Small patch size cannot well distinguish the degradations and image contents in the LR image. As a result, the degradations will

Table 3. Results achieved by our method with different settings.

	Dictionary			Framework		Activated Entries ($\mathbf{D}^{LR}$)	LPIPS (↓)	PSNR (↑)	SSIM (↑)
	Hierarchical	Entries	Dim	Patch (LR/HR)	Coupling				
A1	✗	4096	4	$2\times2/8\times8$	✓	3560	0.281	22.40	0.619
B1	✓	2048	4	$2\times2/8\times8$	✓	2027	0.278	22.41	0.621
B2	✓	8192	4	$2\times2/8\times8$	✓	8050	0.269	22.44	0.626
C1	✓	4096	16	$2\times2/8\times8$	✓	568	0.262	22.25	0.625
C2	✓	4096	64	$2\times2/8\times8$	✓	105	0.268	21.99	0.616
D1	✓	4096	4	$1\times1/4\times4$	✓	1522	0.271	22.41	0.623
D2	✓	4096	4	$4\times4/16\times16$	✓	3810	0.313	21.43	0.588
E1	✓	4096	4	$2\times2/8\times8$	✗	4096	0.533	8.28	0.050
Ours	✓	4096	4	$2\times2/8\times8$	✓	4050	0.259	22.46	0.629

Table 4. Results achieved by our method with different network architectures. $\mathrm{E_{HR}}$, $\mathrm{D_{HR}}$, $\mathrm{E_{LR}}$ and $\mathrm{D_{LR}}$ represent HR image encoder, HR image decoder, LR image encoder, and LR image decoder, respectively.

	$\mathrm{E_{HR}}$	$\mathrm{D_{HR}}$	$\mathrm{E_{LR}}$	$\mathrm{D_{LR}}$	LPIPS (↓)	PSNR (↑)	SSIM (↑)
F1	20-layer	12-layer	6-layer	6-layer	0.271	22.46	0.625
F2	12-layer	20-layer	6-layer	6-layer	0.256	22.49	0.631
F3	12-layer	12-layer	14-layer	6-layer	0.263	22.48	0.629
F4	12-layer	12-layer	6-layer	14-layer	0.261	22.48	0.630
Ours	12-layer	12-layer	6-layer	6-layer	0.259	22.46	0.629

also be reconstructed in the SR results with inferior performance. Although large patch size can identify structured image contents from degradations, the reconstructed SR results are usually of high quality but low fidelity [47]. In summary, a reasonable patch size is critical to the performance. Compared to model D1 and D2, our model produces superior accuracy in terms of all metrics. Consequently, patch sizes of 2×2 and 8×8 are adopted as the default setting.

(2) Coupling Operation

By employing the same spatial codes over HR content dictionary $\mathbf{D}^{HR}$ and LR content dictionary $\mathbf{D}^{LR}$ to represent paired HR and LR images, these dictionaries are tightly coupled. To investigate its effectiveness, we introduce a network variant (E1 in Table 3) by removing this coupling operation. Specifically, HR and LR feature maps are represented using separate spatial codes during the discretization stage. From Table 3 we can see that model E1 produces extremely low quantitative scores. Without the coupling operation, the HR and LR content dictionaries are not associated such that the reconstructed SR result does not share the same content with the LR input. In contrast, the coupling operation enables our model to produce SR results faithful to the LR image.

4.3.3 Network Architecture

Our network consists of four modules, including HR image encoder, HR image decoder, LR image encoder, and LR image decoder. We conduct experiments to study the effect of the model capacity for these four modules. Specifically, we first develop four network variants (F1-F4 in Table 4) by deepening these four modules. As we can see, our method does not obtain consistent performance gains as the HR encoder, LR encoder and LR decoder are deepened. Meanwhile, with larger HR decoder, model F2 produces slight improvements on all metrics. This indicates that our method relies more on the design of dictionaries (Table 3) rather than the designs of the network architecture. Despite the marginal improvements brought by larger HR decoder, a 12-layer decoder is employed to achieve the accuracy-efficiency balance.

5. Conclusion

In this paper, we propose to associate unpaired LR and HR images using coupled dictionaries rather than synthesizing pseudo LR images. Specifically, we construct an HR content dictionary, an LR content dictionary, and a degradation dictionary to model unpaired HR and LR images. To couple these dictionaries, we develop an autoencoder-based framework for optimization. By representing images using compact proxy codes, an LR-to-HR mapping can be effectively learned by our dictionaries. Experiments on several benchmarks show that our method produces superior performance against previous state-of-the-art approaches.

Acknowledgments: This work was partially supported by the National Natural Science Foundation of China (No. 62301601, 62301306).

References

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199, 2014. 1, 2

[2] 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.

[3] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 1646–1654, 2018. 2, 5, 7

[4] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCVW*, pages 1833–1844, 2021. 1

[5] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *CVPR*, pages 1604–1613, 2019. 1, 2, 5, 7

[6] Zongsheng Yue, Qian Zhao, Jianwen Xie, Lei Zhang, Deyu Meng, and Kwan-Yee K Wong. Blind image super-resolution with elaborate degradation modeling on noise and kernel. In *CVPR*, pages 2128–2138, 2022. 1

[7] Shunta Maeda. Unpaired image super-resolution using pseudo-supervision. In *CVPR*, pages 291–300, 2020. 1, 2

[8] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *CVPR*, pages 13385–13394, 2021. 1, 5

[9] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPRW*, pages 701–710, 2018. 1, 2, 5, 7

[10] Tao Yang, Peiran Ren, Lei Zhang, et al. Synthesizing realistic image restoration training pairs: A diffusion approach. *arXiv*, 2023. 1, 2

[11] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Multi-scale dictionary for single image super-resolution. In *CVPR*, pages 1114–1121, 2012. 1, 2

[12] Jianchao Yang, Zhaowen Wang, Zhe Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, aug 2012. 1, 2

[13] Xiangyu Chen, Xintao Wang, Jiantao Zhou, and Chao Dong. Activating more pixels in image super-resolution transformer. In *CVPR*, 2023. 2

[14] Ziwei Luo, Haibin Huang, Lei Yu, Youwei Li, Haoqiang Fan, and Shuaicheng Liu. Deep constrained least squares for blind image super-resolution. In *CVPR*, pages 17642–17652, 2022.

[15] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *ICCV*, pages 12312–12321, 2023. 2

[16] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, pages 1905–1914, 2021. 2, 5, 7

[17] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, pages 4791–4800, 2021. 5, 7

[18] Xuhai Chen, Jiangning Zhang, Chao Xu, Yabiao Wang, Chengjie Wang, and Yong Liu. Better" cmos" produces clearer images: Learning space-variant blur estimation for blind image super-resolution. In *CVPR*, pages 1651–1661, 2023.

[19] Zeshuai Deng, Zhuokun Chen, Shuaicheng Niu, Thomas Li, Bohan Zhuang, and Mingkui Tan. Efficient test-time adaptation for super-resolution with second-order degradation and reconstruction. In *NeurIPS*, volume 36, 2024. 2

[20] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *ICCV*, pages 3086–3095, 2019. 2

[21] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *CVPR*, pages 1652–1660, 2019. 2

[22] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a GAN to learn how to do image degradation first. In *ECCV*, pages 185–200, 2018. 2

[23] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *ICCVW*, pages 3408–3416, 2019. 2, 5

[24] Yang Liu, Ziyu Yue, Jinshan Pan, and Zhixun Su. Unpaired learning for deep image deraining with rain direction regularizer. In *ICCV*, pages 4753–4761, 2021. 2

[25] Yang Yang, Chaoyue Wang, Risheng Liu, Lin Zhang, Xiaojie Guo, and Dacheng Tao. Self-augmented unpaired image dehazing via density and depth decomposition. In *CVPR*, pages 2037–2046, 2022. 2

[26] Valentin Wolf, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. DeFlow: Learning complex image degradations from unpaired data with conditional flows. In *CVPR*, pages 94–103, 2021. 2, 5, 7

[27] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2020. 2

[28] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 2

[29] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *ICML*, pages 689–696, 2009. 2

[30] Ivana Tošić and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011. 2

[31] Xiaoqiang Lu, Haoliang Yuan, Pingkun Yan, Yuan Yuan, and Xuelong Li. Geometry constrained sparse coding for single image super-resolution. In *CVPR*, pages 1648–1655, 2012. 2

[32] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 2

[33] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 32, 2019. 2

[34] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 3, 4

[35] Shunta Maeda. Image super-resolution with deep dictionary. In *ECCV*, pages 464–480, 2022. 3

[36] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. In *ICLR*, 2021. 4

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[38] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Manuel Fritsche, Shuhang Gu, Kuldeep Purohit, Praveen Kandula, Maitreya Suin, AN Rajagoapalan, Nam Hyung Joon, et al. AIM 2019 challenge on real-world image super-resolution: Methods and results. In *ICCVW*, pages 3575–3583, 2019. 5, 7

[39] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. NTIRE 2020 challenge on real-world image super-resolution: Methods and results. In *CVPRW*, pages 494–495, 2020. 5

[40] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[41] Assaf Shocher, Nadav Cohen, and Michal Irani. "Zero-shot" super-resolution using deep internal learning. In *CVPR*, pages 3118–3126, 2018. 5, 7

[42] Zhengxiong Luo, Yang Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In *NeurIPS*, pages 5632–5643, 2020. 5, 7

[43] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *ICCVW*, pages 3599–3608, 2019. 5, 7

[44] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *CVPRW*, pages 466–467, 2020. 5, 7

[45] Wei Wang, Haochen Zhang, Zehuan Yuan, and Changhu Wang. Unsupervised real-world super-resolution: A domain adaptation perspective. In *ICCV*, pages 4318–4327, 2021. 5, 7

[46] Andrés Romero, Luc Van Gool, and Radu Timofte. Unpaired real-world super-resolution with pseudo controllable restoration. In *CVPRW*, pages 798–807, 2022. 5, 7

[47] Yuchao Gu, Xintao Wang, Liangbin Xie, Chao Dong, Gen Li, Ying Shan, and Ming-Ming Cheng. Vqfr: Blind face restoration with vector-quantized dictionary and parallel decoder. In *ECCV*, pages 126–143. Springer, 2022. 8