Confidence-Based Iterative Generation for Real-World Image Super-Resolution

Jialun Peng[1], Xin Luo[1], Jingjing Fu[2⋆], and Dong Liu[1⋆]

[1] University of Science and Technology of China
{pjl,xinluo}@mail.ustc.edu.cn,dongeliu@ustc.edu.cn
[2] Microsoft Research Asia
jifu@microsoft.com

Abstract. Real-world image super-resolution deals with complex and unknown degradations, making it challenging to produce plausible results in a single step. In this work, we propose a transformer model with an iterative generation process that iteratively refines the results based on predicted confidences. It allows the model to focus on regions with low confidences and generate more confident and accurate results. Specifically, our model learns to predict the visual tokens of the high-resolution image and their corresponding confidence scores, conditioned on the low-resolution image. By keeping only the most confident tokens at each iteration and re-predicting the other tokens in the next iteration, our model generates all high-resolution tokens within a few steps. To ensure consistency with the low-resolution input image, we further propose a conditional controlling module that utilizes the low-resolution image to control the decoding process from high-resolution tokens to image pixels. Experiments demonstrate that our model achieves state-of-the-art performance on real-world datasets while requiring fewer iteration steps compared to recent diffusion models.

1 Introduction

Image super-resolution (SR) aims at generating a high-resolution (HR) image that is consistent with an input low-resolution (LR) image. SR has been an active research topic for decades [2, 4, 47, 53, 54, 64] because of its high practical values in enhancing image details and visual quality. With the advance of deep neural networks, numerous deep learning-based methods [13, 17, 23, 25, 34, 52, 68] have been proposed for SR. Most methods [11, 12, 61, 73] assume a naive bicubic degradation process from HR to LR images. However, the performance of these methods will deteriorate a lot in real applications because of the mismatch between bicubic and real-world degradations.

The real-world degradations are complex since real LR images are usually corrupted by multiple types of degradations (*e.g.*, blur, noise, down-sampling, and JPEG compression). These degradations are unknown due to the varying

⋆ Jingjing Fu and Dong Liu are the corresponding authors. This work was done when Jialun Peng was an intern at Microsoft Research Asia, from Jan. 2023 to Aug. 2023.

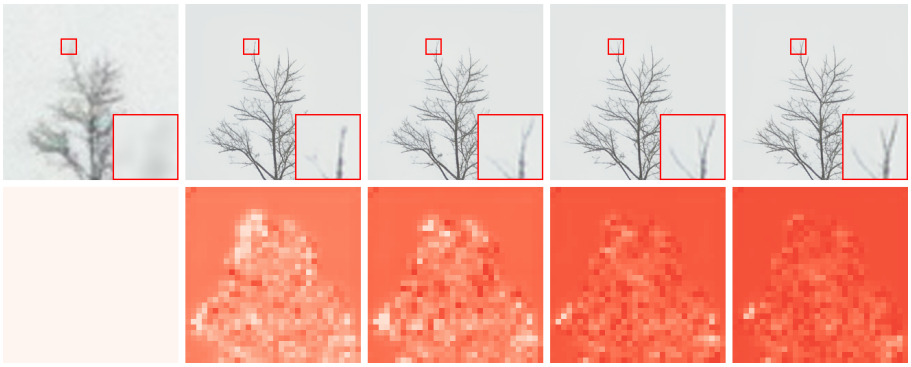

Fig. 1: Visualizations of our confidence-based iterative generation process for real-world SR. The iterative process leads to better perceptual quality and improved confidence scores across multiple steps. (Top) The LR input image and the generated HR images. (Bottom) The predicted confidence scores for HR tokens (darker denotes higher confidence scores).

image degradation processes, different imaging devices, and diverse image signal processing methods [36, 63]. Therefore, real-world SR is a challenging task which demands robust SR methods to generate high-resolution and high-quality images while removing the complex unknown degradations. Most of the existing methods [8,22,32,59,63,65] adopt Generative Adversarial Networks (GANs) [20] for real-world SR. However, these GANs suffer from the well-known problem of training instability. Much efforts have been made to carefully design the network architectures and optimization tricks of GANs in order to obtain plausible results [59]. Still, these GAN-based methods often produce visual artifacts when dealing with different degradations. This can be attributed to the one-step generation process of GANs, which performs an intractable task to simultaneously generate all image pixels regardless of whether the regions are simple or complex.

In this paper, we aim to improve the visual quality of SR results by generating HR images in an iterative manner. We propose RealSRT, a Real-world SR Transformer model with a confidence-based iterative generation process. The iterative process revisits the HR output and the LR input multiple times, predicting confidence scores for both simple and complex regions at each iteration. These confidence scores provide a measure of certainty about the generated content, allowing the model to focus on uncertain regions with low confidence and generate more confident and accurate results. Consequently, our confidence-based iterative generation process enhances the model's adaptability and performance in handling complex and unknown degradations, resulting in better perceptual quality and improved confidence scores (as shown in Fig. 1).

The key design of our RealSRT is a masked transformer [6, 7, 28] in the discrete token space of a pre-trained VQGAN [18]. Inspired by Token-Critic [27], our masked transformer consists of a generator network that predicts tokens and a critic network that predicts confidence scores. The generator is trained on a masked modeling task that predicts randomly masked HR tokens given the

unmasked HR tokens and all LR tokens. The critic is trained to distinguish which HR tokens are incompatible by considering self-attention between HR tokens and cross-attention from LR tokens to HR tokens. During inference, our masked transformer iteratively generates HR tokens starting from a blank canvas with all HR tokens masked out. At each iteration, the generator predicts the masked HR tokens in parallel and the critic predicts confidence scores for HR tokens. By keeping only high-confidence HR tokens and masking out low-confidence HR tokens for the next iteration, our masked transformer generates all HR tokens within a few iteration steps.

It is important for the HR output image to be consistent with the LR input image. To ensure consistency with the LR image, we propose a conditional controlling module (CCM) that utilizes the LR image as a condition to control the decoding process. The CCM employs a trainable copy of the pre-trained VQGAN encoder and an attention block. The trainable encoder extracts LR features from the LR image, while the attention block fuses these LR features with the generated HR tokens to learn a residual for enhancing the decoding process of HR tokens. The proposed CCM improves the consistency with the LR input in terms of image color and details.

We conduct comprehensive experiments on real-world datasets and the experimental results demonstrate that RealSRT achieves state-of-the-art performance in real-world SR. Compared to GAN-based methods [8, 22, 32, 59, 65], RealSRT exhibits strong performance due to its confidence-based iterative generation process. Compared to the recent diffusion models [35, 57], RealSRT is significantly more efficient since it requires fewer iteration steps at inference time. The contributions of this work can be summarized as follows:

- We present RealSRT, a novel transformer model for real-world image super-resolution. This model uses a confidence-based iterative generation process that iteratively refines the HR results based on predicted confidences.
- We design a masked transformer, where a generator predicts HR tokens and a critic predicts confidence scores. The masked transformer generates all HR tokens by keeping only high-confidence tokens at each iteration and masking out low-confidence ones for the next iteration.
- To ensure the consistency of the HR output image with the LR input image, we propose a conditional controlling module that utilizes the LR image as a condition to control the decoding process of HR tokens.

2 Related Work

Image Super-Resolution. Classical single image super-resolution methods [9,13,16,17,26,42,45,68,72] are mainly designed for simple and uniform degradations (*e.g.*, bicubic degradations). Although significant improvements in terms of PSNR [11, 29, 30] and perceptual quality [31, 61] have been achieved, such methods usually fail in real-world SR tasks where the degradations are complex and unknown.

To address the above-mentioned problem, recent works have proposed several degradation models for real-world SR. Existing degradation models can be

categorized into implicit modeling and explicit modeling. Implicit degradation modeling [19, 41, 58, 63] aims to learn the degradation process from real-world images, which requires large external datasets for training [36]. In contrast, explicit degradation modeling aims to simulate real-world degradations by designing a random shuffle order [65] or a high-order degradation process [59]. These explicit degradation processes synthesize realistic LR images from HR images to collect large-scale training pairs, enabling the recent real-world SR methods [8, 32, 33, 35, 57, 59, 65] to achieve state-of-the-art results.

Most of the existing methods [22, 32, 33, 44, 59, 65, 70] for real-world SR are based on Generative Adversarial Networks (GANs) [20]. These GAN-based methods suffer from unstable training due to their min-max optimization. In addition, these GAN-based methods are prone to unpleasant artifacts because their one-step generation process often fails to adequately address the complex degradations in real-world scenarios. Recently, diffusion models [35, 50, 51, 57] perform super-resolution through a stochastic iterative denoising process. However, these diffusion models have low efficiency since they require a large number of iteration steps at inference time. In this work, we propose a novel transformer model with a confidence-based iterative generation process that iteratively generates high-resolution and high-quality images while requiring fewer iteration steps compared to diffusion models.

Generative Image Transformers. Inspired by the success of Transformer [55] and GPT [3] in the NLP field, generative transformers have been applied to various image synthesis tasks [6, 10, 48]. These generative image transformers consist of two stages. In the first stage, a vector-quantized (VQ) autoencoder [18, 46, 49] learns a discrete codebook to tokenize images into visual tokens. In the second stage, a transformer iteratively generates visual tokens based on the previously generated ones. Finally, the generated visual tokens are mapped into image pixels using the decoder from the first stage.

Early works applied autoregressive transformers [10, 18, 48] to generate one token at each iteration step. Recently, non-autoregressive transformers [7, 69] utilize iterative parallel decoding that significantly accelerates the inference time. In particular, Masked Generative Image Transformer (MaskGIT) [7] uses the mask prediction inspired by BERT [14] to generate all tokens in a few iteration steps. At each iteration, the model predicts all tokens in parallel but only keeps the most confident ones. The remaining tokens are masked out and will be re-predicted in the next iteration. MaskGIT has demonstrated highly-competitive image generation performance and orders of magnitude faster inference than its autoregressive counterpart [18].

A major challenge of non-autoregressive transformers is to decide which tokens to keep and which to mask. MaskGIT relies on the generator's predicted confidences that are independent for each token, which hinders capturing rich correlations between tokens. Token-Critic [27] improves MaskGIT by introducing a second transformer to distinguish which tokens are unlikely under the true distribution, obtaining much more reliable confidence scores. We extend this method to super-resolution with cross-attention from LR tokens to HR tokens.

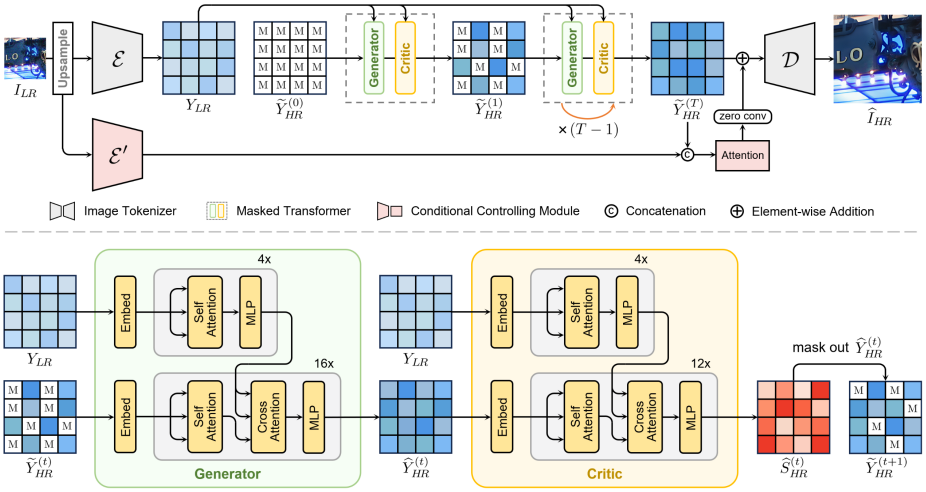

Fig. 2: Overview of RealSRT. (Top) RealSRT consists of an image tokenizer, a masked transformer, and a conditional controlling module. The image tokenizer is a pre-trained VQGAN with an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$. The masked transformer iteratively generates HR tokens within T iteration steps. The conditional controlling module employs a trainable encoder $\mathcal{E}'$ and an attention block to utilize the LR image as a condition to control the decoding process. (Bottom) The masked transformer consists of a generator network and a critic network. Both networks are conditioned on LR tokens with self-attention and cross-attention. At each iteration, the generator predicts masked HR tokens, while the critic predicts confidence scores to mask out low-confidence HR tokens for the next iteration.

3 Method

Our main goal is to develop a new transformer model that iteratively generates high-resolution images for real-world super-resolution tasks. The overview of our proposed model (RealSRT) is illustrated in Fig. 2. RealSRT consists of an image tokenizer, a masked transformer, and a conditional controlling module. We first pre-train a VQGAN [18] to tokenize images into visual tokens. Then a masked transformer iteratively generates HR tokens based on the predicted confidences. In addition, a conditional controlling module utilizes the LR image as a condition to control the decoding process for ensuring consistency with the LR input. Finally, we propose an image patch aggregation algorithm to handle images of arbitrary resolutions.

3.1 Image Tokenizer

We pre-train a VQGAN [18] model as the image tokenizer that tokenizes an input image into a sequence of visual tokens. This model consists of an encoder and a decoder, with a vector quantized (VQ) layer that learns a discrete codebook to quantize image features into visual tokens. More precisely, given an input image

$I \in \mathbb{R}^{H\times W\times 3}$, the encoder $\mathcal{E}$ encodes I into a feature map $\mathcal{E}(I) \in \mathbb{R}^{h\times w\times c}$, where $(h, w) = (H/f, W/f)$ and f is the downsampling factor. The codebook $\mathbf{e}(k) \in \mathbb{R}^c, k \in 1, 2, \cdots, K$ serves for a nearest neighbor look up to quantize the feature map $\mathcal{E}(I)$ into a sequence of visual tokens $Y = [y_i]_{i=1}^N$, where K is the number of codebook embeddings and N is the sequence length. Each $y_i \in 1, 2, \cdots, K$ is an index of the codebook, representing the corresponding codebook embedding. Finally, the decoder $\mathcal{D}$ decodes Y into the reconstructed image $\hat{I} = \mathcal{D}(Y)$. We follow [18] to train VQGAN with pixel, perceptual and adversarial losses. The image tokenizer lowers the computational demands of training transformers by moving the bulk of the computation from pixel space to latent space [6]. Furthermore, the discrete nature of tokens enables effective cross-entropy loss at the output of transformers to predict masked tokens [6,7,28].

3.2 Masked Transformer

We propose a masked transformer for super-resolution that iteratively generates HR tokens in a constant number of iteration steps. As shown in Fig. 2, the masked transformer consists of a generator network and a critic network. Both networks are conditioned on LR tokens with self-attention and cross-attention. At each iteration, the generator predicts masked HR tokens given the unmasked HR tokens, while the critic predicts confidences for the predicted HR tokens and unmasked HR tokens. The critic keeps only high-confidence HR tokens and masks out low-confidence HR tokens for the next iteration. Finally, the masked transformer iteratively generates all HR tokens starting from a blank canvas filled with mask tokens.

Predicting Tokens with Generator. The generator predicts masked HR tokens based on the unmasked HR tokens and all LR tokens. Let I_{HR} and I_{LR} denote the high-resolution image and the low-resolution counterpart, respectively. We input I_{HR} to the VQGAN encoder $\mathcal{E}$ to obtain the high-resolution tokens Y_{HR}. We upsample I_{LR} using a bicubic interpolation and input it to $\mathcal{E}$ to obtain the low-resolution tokens Y_{LR} with the same size as Y_{HR}. Let $M = [m_i]_{i=1}^N$ denote a binary mask determining which HR tokens are to be masked. The HR tokens will be replaced with a learnable mask token [M] when $m_i = 1$, while they remain unchanged when $m_i = 0$. Let $\widetilde{Y}_{HR}$ denote the HR tokens with masking. During training, we randomly mask out HR tokens using a masking ratio sampled from the mask scheduling function $\gamma(r) \in [0, 1]$, where $r \in [0, 1]$ is an uniform random number. The number of masked HR tokens is $\lceil \gamma(r) \cdot N \rceil$, where N is the sequence length.

The generator follows a bidirectional transformer architecture [14]. For both Y_{LR} and $\widetilde{Y}_{HR}$, we utilize the learned codebook embeddings of VQGAN as the embedding layer to leverage the low-level knowledge in pre-trained VQGAN. Moreover, we add 2D sine and cosine positional embeddings [55] to the input embeddings. The generator consists of several transformer layers including self-attention, cross-attention, and MLP. These layers extract features from Y_{LR} and $\widetilde{Y}_{HR}$ with self-attention among tokens and cross-attention from LR tokens to

HR tokens. At the output layer, we use a linear layer to output logits in K classes. These generator logits can be converted to probabilities using a softmax function.

During training, the generator is trained to minimize the negative log-likelihood of the masked HR tokens. We use a cross-entropy loss between the ground-truth HR tokens and the output probabilities of generator:

$$\mathcal{L}_g = \mathbb{E}_{Y,M}\Big[\, CE\left(Y_{HR},\, p_\theta\Big(Y_{HR}\,|\,\widetilde{Y}_{HR}, Y_{LR}\Big)\right)\Big], \tag{1}$$

where p_θ is the probability predicted by the generator, and CE denotes the cross-entropy loss. Note that we only optimize this loss on masked HR tokens. During inference time, the generator predicts the class that has the highest predicted probability $p_\theta(Y_{HR} \mid \widetilde{Y}_{HR}, Y_{LR})$ for each masked HR token, and the unmasked HR tokens are copied into the output to form the generation results $\widehat{Y}_{HR}$.

Predicting Confidences with Critic. The critic predicts confidence scores for the generation results $\widehat{Y}_{HR}$ based on all LR tokens. The generation results include the predicted HR tokens and unmasked HR tokens. The critic has a similar architecture to the generator, and uses the same embedding layer and positional embeddings as the generator. The transformer layers of critic extract features from Y_{LR} and $\widehat{Y}_{HR}$ with self-attention and cross-attention. At the output layer, we use a linear layer to output logits in one single class. These critic logits can be converted to probabilities using a sigmoid function.

During training, the generator is held fixed, and the critic is trained to distinguish which of the HR tokens in the generation results were originally masked. We use a binary cross-entropy loss between the original mask and the output probabilities of critic:

$$\mathcal{L}_c = \mathbb{E}_{Y,M}\Big[\, BCE\left(M,\, p_\phi\Big(M\,|\,\widehat{Y}_{HR}, Y_{LR}\Big)\right)\Big], \tag{2}$$

where p_ϕ is the probability predicted by the critic, and BCE denotes the binary cross-entropy loss. We optimize this loss on both predicted and unmasked HR tokens. During inference time, the critic predicts the confidence score $1 - p_\phi(M \mid \widehat{Y}_{HR}, Y_{LR})$ for each HR token, which forms the confidence results $\widehat{S}_{HR}$.

Iterative Generation. The masked transformer iteratively generates all HR tokens in T iteration steps. We start from a blank canvas filled with masked tokens, *i.e.* $\widetilde{Y}_{HR}^{(0)}$. At iteration $t \in 0, 1, \cdots, (T-1)$, we input $\widetilde{Y}_{HR}^{(t)}$ to the generator to predict masked HR tokens and obtain the generation results $\widehat{Y}_{HR}^{(t)}$. Then the critic predicts confidences for $\widehat{Y}_{HR}^{(t)}$ and obtains the confidence results $\widehat{S}_{HR}^{(t)}$. We mask out $n = \lceil \gamma(\frac{t+1}{T}) \cdot N \rceil$ HR tokens in $\widehat{Y}_{HR}^{(t)}$ with the lowest confidences to generate $\widetilde{Y}_{HR}^{(t+1)}$ for the next iteration. Note that $\gamma(1) = 0$, thus no masking will be performed at the last iteration and we obtain the final generation results $\widetilde{Y}_{HR}^{(T)}$.

MaskGIT [7] uses the generator's predicted probabilities as confidences to select which tokens to mask out. These generator confidences are independent for each token, which hinders capturing rich correlations between tokens. In

addition, the generator confidences are greedy and cannot correct previously generated tokens. Compared to generator confidences, critic confidences measure the likelihood of tokens under the true joint distribution by taking into account the correlations among tokens [27]. Moreover, the critic confidences allow to correct previously generated tokens that are no longer as likely based on the most recent generation results, addressing the greedy generation issue.

3.3 Conditional Controlling Module

It is important for the HR output image to be consistent with the LR input image [51]. We propose a conditional controlling module (CCM) to ensure consistency with the LR image. The CCM utilizes the LR image as a condition to control the decoding from $\widetilde{Y}_{HR}^{(T)}$ to the HR output image $\widehat{I}_{HR}$. More precisely, CCM employs a trainable copy of VQGAN encoder and an attention block. The trainable encoder $\mathcal{E}'$ encodes the upsampled LR image into a feature map and concatenates it to $\widetilde{Y}_{HR}^{(T)}$. The attention block, which has a self-attention layer between two residual layers, processes the concatenated features and then adds them to $\widetilde{Y}_{HR}^{(T)}$ with a zero convolution layer [66]. The weight and bias parameters of the zero convolution layer are initialized to zero, which ensures that no harmful noise is added to the generated tokens at the beginning of CCM training.

We train CCM while keeping the pre-trained VQGAN and the masked transformer frozen. The training losses of CCM include pixel, perceptual and adversarial losses, which are the same as that of VQGAN. We calculate these losses between the HR output image $\widehat{I}_{HR}$ and the ground-truth HR image I_{HR}. We also use the same weighting factors for each loss as the pre-training of VQGAN.

3.4 Image Patch Aggregation

The attention mechanism of the transformer puts limits on the sequence length N. To generate high-resolution images of arbitrary resolutions at inference time, we propose an image patch aggregation algorithm to process image patches independently and aggregate them using a Gaussian kernel. Specifically, we first split the LR input image into overlapping LR patches, where each patch can be upsampled and tokenized into a sequence of tokens with a length of N. The overlapping width is half of the patch size. Then the masked transformer and the conditional controlling module process each LR patch individually and generate the HR patches. To improve the color consistency, we employ adaptive instance normalization for color correction [57] on each HR patch, aligning its mean and variance with those of the corresponding LR patch.

Unlike StableSR [57] aggregates features in latent space at each step, we aggregate image patches in pixel space. We aggregate all the processed HR patches into full resolution using a centered Gaussian kernel. Overlapping pixels are weighted in accordance with their respective Gaussian weight maps. Let $\{I_{\Omega_j}\}_{j=1}^{P}$ denote the overlapping HR image patches, where P is the number of image patches and Ω_j is the coordinate set of the jth image patch in the full

resolution image. W_{Ω_j} denotes a Gaussian weight map of the same size as the full resolution image, whose entries follow up a centered Gaussian kernel in Ω_j and 0 elsewhere. The aggregation of HR image patches is formulated as follows:

$$I_{agg} = \sum_{j=1}^{P} \frac{W_{\Omega_j}}{W_{sum}} \odot I_{\Omega_j}, \tag{3}$$

where $W_{sum} = \sum_j W_{\Omega_j}$ and I_{agg} is the final aggregated HR output image. Our image patch aggregation algorithm eliminates the boundary artifacts in overlapped regions, resulting in coherent HR outputs.

4 Experiments

4.1 Experimental Details

Implementation. We pre-train a VQGAN on 256×256 cropped images with codebook size $K = 512$ and downsampling factor $f = 8$, resulting in tokens with spatial size 32×32 and sequence length $N = 1024$. We follow [18] to train VQGAN with a pixel loss, a perceptual loss, and an adversarial loss. The weighting factor for the perceptual loss is 0.1 and we use an adaptive weighting factor [18] for the adversarial loss. We use Adam optimizer [24] to train VQGAN for 200 epochs with the batch size 64 and the learning rate 4×10^{-5}.

Masked transformer consists of a generator network and a critic network. The generator has 16 transformer layers with cross-attention and 4 transformer layers without cross-attention. The critic has 12 transformer layers with cross-attention and 4 transformer layers without cross-attention. Both networks use 512 hidden dimensions, 8 attention heads, and 2D positional embeddings [55]. The sequence length is 1024. The output dimensions of generator and critic are 512 and 1, respectively. We use a square root mask scheduling function $\gamma(r) = 1 - \sqrt{r}$. Both generator and critic are trained for 800K iterations using AdamW optimizer [38], the batch size 64, and the initial learning rate 1×10^{-4} gradually decreased to 1×10^{-7} with the cosine annealing [37]. We perform exponential moving averaging (EMA) on both networks with a decay factor of 0.999. During inference time, we use $T = 4$ to achieve a good balance between performance and efficiency.

The CCM consists of a trainable copy of VQGAN encoder and an attention block. We freeze the pre-trained VQGAN and the masked transformer when training CCM. The training losses and weighting factors of CCM are the same as that of VQGAN. We use Adam optimizer to train CCM for 80K iterations with the batch size 8 and the learning rate 4×10^{-5}. We also perform exponential moving averaging (EMA) on CCM with a decay factor of 0.999.

The entire training process of RealSRT is conducted on 256×256 resolution with 8 NVIDIA 32G-V100 GPUs. We first train the generator. Then we freeze the generator and train the critic. Finally, we train the CCM while keeping the masked transformer frozen. For inference, we adopt the proposed image patch aggregation algorithm to handle arbitrary resolutions. The overlapping width is 128, and the variance of the centered Gaussian kernel is set to 0.05.

Datasets. Similar to Real-ESRGAN [59] and DASR [32], we adopt DIV2K [1], Flickr2K [53] and OST [60] datasets for training. The datasets contain 13774 high-quality and high-resolution images, including 800 images in DIV2K, 2650 images in Flickr2K and 10324 images in OST. We randomly crop 256×256 image patches from these HR images to train our RealSRT. To synthesize LR images, we use the high-order degradation modeling of Real-ESRGAN [59] to process the HR patches. Following previous works [35, 57], all experiments are performed with a scaling factor of ×4 between LR and HR images. Thus we synthesize 64×64 LR patches during training.

For evaluation, we mainly test our RealSRT on three real-world SR benchmark datasets, including RealSR [5], DRealSR [62], and DPED [21]. RealSR and DRealSR contain 100 and 93 LR-HR image pairs, respectively. DPED contains 100 LR images without ground-truth HR images. We also test RealSRT on some images from AIM19 [40] and NTIRE20 [39] datasets. The LR images in these two datasets are synthesized using artificial image degradations.

Metrics. We evaluate four full-reference metrics (*i.e.*, PSNR, MS-SSIM, LPIPS [67], and DISTS [15]) and two no-reference metrics (*i.e.*, NIQE [43] and CLIP-IQA [56]) on RealSR and DRealSR datasets since they have ground-truth HR images. We also evaluate NIQE and CLIP-IQA on DPED due to the absence of ground-truth. Please refer to the supplementary material for the quantitative results on DPED.

4.2 Comparisons with Existing Methods

We compare our RealSRT with existing GAN-based and diffusion-based methods. GAN-based methods include RealSR [22], BSRGAN [65], DASR [32], FeMaSR [8], Real-ESRGAN+ [59], and SwinIR-GAN [33]. Diffusion-based methods include DiffBIR [35] and StableSR [57]. We use the official code and models for these methods. Unlike StableSR [57] which resizes and crops image patches for evaluation, we directly evaluate the original testing images of benchmark datasets to make a fair comparison.

Quantitative Comparisons. The main quantitative comparisons are presented in Tab. 1 and Tab. 2. It can be seen that our RealSRT achieves state-of-the-art results in perceptual metrics LPIPS and DISTS on both RealSR [5] and DRealSR [62] datasets. It indicates that our RealSRT is more perceptually faithful to the ground-truth than other methods. Besides, our RealSRT obtains good PSNR and MS-SSIM scores. Even though DASR [32] also performs good in PSNR and MS-SSIM, it has an inferior performance in perceptual metrics LPIPS and DISTS. Note that DiffBIR [35] performs well in no-reference metrics NIQE and CLIP-IQA. However, DiffBIR often produces unexpected artifacts that are inconsistent with the input images, leading to bad results in full-reference metrics.

As shown in Tab. 1 and Tab. 2, our RealSRT requires the fewest steps among all non-GAN-based methods. Compared to diffusion-based methods DiffBIR [35] and StableSR [57], our RealSRT is more efficient at inference time, requiring significantly fewer iteration steps.

Table 1: Quantitative comparisons on RealSR [5]. Best and second best scores are **bolded** and underlined, respectively.

Method	Steps	PSNR↑	MS-SSIM↑	LPIPS↓	DISTS↓	NIQE↓	CLIP-IQA↑
RealSR [22]	1	22.45	0.8508	0.3092	0.1970	**3.587**	0.5115
BSRGAN [65]	1	24.88	0.8913	0.2685	0.1761	4.652	0.5438
DASR [32]	1	25.58	**0.8939**	0.3113	0.1838	5.969	0.3629
FeMaSR [8]	1	23.87	0.8819	0.2927	0.1940	4.758	0.5598
Real-ESRGAN+ [59]	1	24.31	0.8861	0.2728	0.1685	4.677	0.4899
SwinIR-GAN [33]	1	24.44	0.8911	0.2593	0.1609	4.636	0.4744
DiffBIR [35]	50	23.45	0.8476	0.3625	0.1860	3.708	**0.6731**
StableSR [57]	200	24.62	0.8864	0.2587	0.1582	5.127	0.5211
RealSRT	4	**25.60**	0.8928	**0.2527**	**0.1525**	5.084	0.4832

Table 2: Quantitative comparisons on DRealSR [62].

Method	Steps	PSNR↑	MS-SSIM↑	LPIPS↓	DISTS↓	NIQE↓	CLIP-IQA↑
RealSR [22]	1	25.31	0.8634	0.3578	0.1969	3.664	0.5024
BSRGAN [65]	1	26.18	0.8920	0.2930	0.1636	4.681	0.5705
DASR [32]	1	**27.39**	**0.9077**	0.2962	0.1689	6.347	0.3844
FeMaSR [8]	1	24.56	0.8624	0.3374	0.1766	4.218	0.6126
Real-ESRGAN+ [59]	1	25.82	0.8934	0.2818	0.1464	4.716	0.5179
SwinIR-GAN [33]	1	25.73	0.8909	0.2838	0.1462	4.566	0.5043
DiffBIR [35]	50	25.18	0.8461	0.4632	0.2105	**2.964**	**0.7045**
StableSR [57]	200	27.16	0.9060	0.2648	0.1390	5.575	0.4753
RealSRT	4	27.37	0.9061	**0.2640**	**0.1368**	5.323	0.5288

Qualitative Comparisons. Fig. 3 shows the qualitative comparisons on real-world datasets including RealSR [5], DRealSR [62], and DPED [21]. It can be observed that our RealSRT generates high-quality results while ensuring consistency with LR input images. Compared to GAN-based methods (*i.e.*, RealSR [22], BSRGAN [65], DASR [32], FeMaSR [8], Real-ESRGAN+ [59], and SwinIR-GAN [33]), our RealSRT performs better in recovering image details due to the iterative generation process. As for the diffusion-based methods, DiffBIR [35] tends to introduce unexpected artifacts that are inconsistent with input images, while StableSR [57] sometimes produces results that appear slightly blurry. In contrast, our RealSRT generates sharp and realistic results with fewer artifacts. Please refer to the supplementary material for more qualitative comparisons on real-world images with diverse resolutions.

4.3 Ablation Studies

We conduct ablation experiments on the RealSR [5] dataset to analyze our RealSRT. We first build a baseline model (Exp. (a)) that employs the generator's predicted probabilities as confidences for token selection and uses the VQGAN decoder for decoding. Then we build a modified model (Exp. (b)) without CCM, which uses the critic confidences for token selection and still uses the VQGAN decoder for decoding. Our full model (Exp. (c)), equipped with CCM, uses critic confidences for token selection and incorporates the CCM into the decoding process. All these models utilize our proposed image patch aggregation algorithm with color correction to generate images of arbitrary resolutions. Next, we compare these models to validate the effectiveness of our critic and CCM.

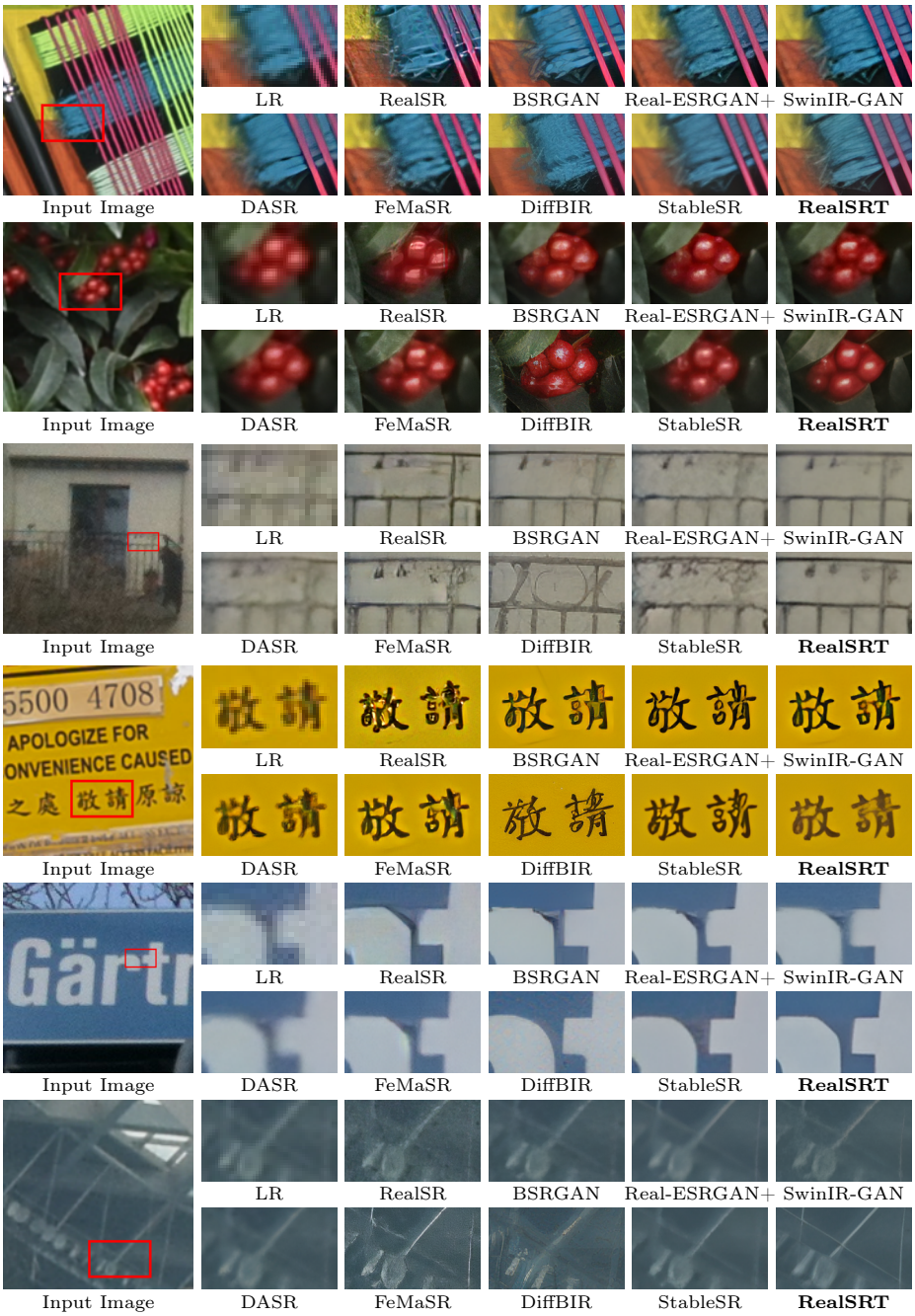

Fig. 3: Qualitative comparisons for ×4 SR on real-world datasets (please zoom in for best view). Compared to existing methods, our RealSRT generates high-quality and perceptually faithful results that are more consistent with input images.

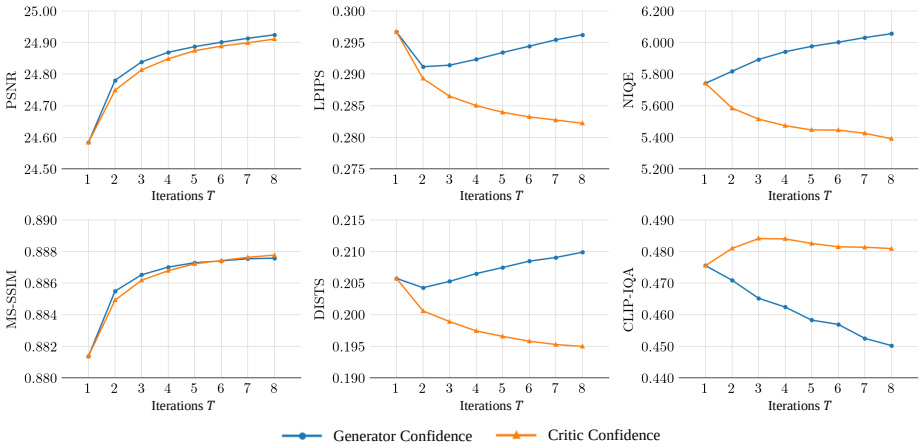

Fig. 4: Effects of confidences and the number of iterations on RealSR [5]. Higher is better for PSNR, MS-SSIM and CLIP-IQA (↑), while lower is better for LPIPS, DISTS and NIQE (↓). Critic confidences are more reliable than generator confidences, leading to better performance in LPIPS, DISTS, NIQE, and CLIP-IQA. In our experiments, we use critic confidences and $T = 4$ to achieve a good balance between performance and efficiency.

Effects of Confidences and Iteration Steps. We compare Exp. (a) and Exp. (b) to investigate the significance of critic confidences. Exp. (a) uses generator confidences while Exp. (b) uses critic confidences. Fig. 4 shows the effects of different confidences and the number of iterations. Increasing the number of iterations tends to improve PSNR and MS-SSIM results for both confidences. However, when using generator confidences, using more iterations deteriorates LPIPS, DISTS, NIQE, and CLIP-IQA results. It is due to that using more iterations tends to produce blurry results when using generator confidences (please refer to the supplementary material for visual samples). In contrast, when using critic confidences, using more iterations tends to enhance LPIPS, DISTS, and NIQE results. Moreover, using $T = 3$ or $T = 4$ achieves the best CLIP-IQA results. Beyond $T = 4$, more iterations lead to worse CLIP-IQA results. Fig. 4 suggests that critic confidences are more reliable than generator confidences, enabling iterative generation for real-world SR tasks. We use critic confidences and $T = 4$ to achieve a good balance between performance and efficiency. As shown in Tab. 3, Exp. (b) performs better than Exp. (a) in LPIPS, DISTS, NIQE, and CLIP-IQA when $T = 4$. Fig. 5 presents the visual comparisons. The critic improves the baseline model by generating more image details, thus enhancing the perceptual quality of visual results.

Importance of CCM. We compare Exp. (b) and Exp. (c) to verify the importance of CCM. Exp. (b) directly uses the VQGAN decoder to decode the HR tokens into image pixels. Exp. (c) incorporates the CCM into the decoding process. The CCM utilizes the LR input as a condition to control the decoding of HR tokens, which ensures the consistency of the HR output with the LR in-

Fig. 5: Visual comparisons of critic and CCM. The baseline model only uses the generator without using the critic and CCM, leading to slight blur. The critic improves the perceptual quality due to its reliable confidences. The CCM further improves the consistency with the LR input in image color and details.

Table 3: Ablation studies of critic and CCM on RealSR [5]. We set $T = 4$.

Exp.	Critic	CCM	PSNR$^{\uparrow}$	MS-SSIM$^{\uparrow}$	LPIPS$^{\downarrow}$	DISTS$^{\downarrow}$	NIQE$^{\downarrow}$	CLIP-IQA$^{\uparrow}$
(a)			24.87	0.8870	0.2923	0.2065	5.942	0.4624
(b)	✓		24.85	0.8868	0.2850	0.1974	5.473	**0.4840**
(c)	✓	✓	**25.60**	**0.8928**	**0.2527**	**0.1525**	**5.084**	0.4832

put. As shown in Tab. 3, CCM significantly improves PSNR, MS-SSIM, LPIPS, DISTS, and NIQE results. The reason is that CCM mitigates the color inconsistency issue which is typical for generative models in latent space [35, 57, 71]. As shown in Fig. 5, CCM improves the consistency with the LR input in terms of image color and details.

5 Conclusion

In this paper, we present RealSRT, a real-world super-resolution transformer with a confidence-based iterative generation process. Trained on a mask modeling task in discrete token space, our masked transformer iteratively generates high-resolution tokens based on predicted confidences. At each iteration, high-confidence tokens are kept and low-confidence tokens are masked out for the next iteration, resulting in better perceptual quality and improved confidence scores across multiple iteration steps. To improve the consistency with the low-resolution input image, we also propose a conditional controlling module that utilizes the low-resolution image to control the decoding process of high-resolution tokens. Experimental results demonstrate that RealSRT is efficient and achieves state-of-the-art performance on real-world datasets.

Ethics statement

RealSRT is purely a research project. Currently, we have no plans to incorporate RealSRT into a product or expand access to the public. We will also put Microsoft AI principles into practice when further developing the models. All the datasets used in this paper are public and have been reviewed to ensure they do not contain any personally identifiable information or offensive content.

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPRW. pp. 126–135 (2017)
2. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: The 2018 PIRM challenge on perceptual image super-resolution. In: ECCVW. pp. 334–355 (2018)
3. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. In: NeurIPS. pp. 1877–1901 (2020)
4. Cai, J., Gu, S., Timofte, R., Zhang, L.: NTIRE 2019 challenge on real image super-resolution: Methods and results. In: CVPRW. pp. 2211–2223 (2019)
5. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image super-resolution: A new benchmark and a new model. In: ICCV. pp. 3086–3095 (2019)
6. Chang, H., Zhang, H., Barber, J., et al.: Muse: Text-to-image generation via masked generative transformers. In: ICML. pp. 4055–4075 (2023)
7. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: Masked generative image transformer. In: CVPR. pp. 11315–11325 (2022)
8. Chen, C., Shi, X., Qin, Y., Li, X., Han, X., Yang, T., Guo, S.: Real-world blind super-resolution via feature matching with implicit high-resolution priors. In: ACM MM. pp. 1329–1338 (2022)
9. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: CVPR. pp. 12299–12310 (2021)
10. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: ICML. pp. 1691–1703 (2020)
11. Chen, X., Wang, X., Zhou, J., Qiao, Y., Dong, C.: Activating more pixels in image super-resolution transformer. In: CVPR. pp. 22367–22377 (2023)
12. Chen, Z., Zhang, Y., Gu, J., Kong, L., Yang, X., Yu, F.: Dual aggregation transformer for image super-resolution. In: ICCV. pp. 12312–12321 (2023)
13. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR. pp. 11065–11074 (2019)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
15. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. IEEE TPAMI **44**(5), 2567–2581 (2020)
16. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. pp. 184–199 (2014)
17. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE TPAMI **38**(2), 295–307 (2015)

18. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12873–12883 (2021)
19. Fritsche, M., Gu, S., Timofte, R.: Frequency separation for real-world super-resolution. In: ICCVW. pp. 3599–3608 (2019)
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)
21. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: DSLR-quality photos on mobile devices with deep convolutional networks. In: ICCV. pp. 3277–3285 (2017)
22. Ji, X., Cao, Y., Tai, Y., Wang, C., Li, J., Huang, F.: Real-world super-resolution via kernel estimation and noise injection. In: CVPRW. pp. 466–467 (2020)
23. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. pp. 1646–1654 (2016)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
25. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. pp. 624–632 (2017)
26. Ledig, C., Theis, L., Huszár, F., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. pp. 4681–4690 (2017)
27. Lezama, J., Chang, H., Jiang, L., Essa, I.: Improved masked image generation with token-critic. In: ECCV. pp. 70–86 (2022)
28. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: MAGE: Masked generative encoder to unify representation learning and image synthesis. In: CVPR. pp. 2142–2152 (2023)
29. Li, W., Lu, X., Qian, S., Lu, J.: On efficient transformer-based image pre-training for low-level vision. In: IJCAI. pp. 1089–1097 (2023)
30. Li, Y., Fan, Y., Xiang, X., Demandolx, D., Ranjan, R., Timofte, R., Van Gool, L.: Efficient and explicit modelling of image hierarchies for image restoration. In: CVPR. pp. 18278–18289 (2023)
31. Liang, J., Zeng, H., Zhang, L.: Details or artifacts: A locally discriminative learning approach to realistic image super-resolution. In: CVPR. pp. 5657–5666 (2022)
32. Liang, J., Zeng, H., Zhang, L.: Efficient and degradation-adaptive network for real-world image super-resolution. In: ECCV. pp. 574–591 (2022)
33. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: SwinIR: Image restoration using swin transformer. In: ICCVW. pp. 1833–1844 (2021)
34. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. pp. 136–144 (2017)
35. Lin, X., He, J., Chen, Z., Lyu, Z., Fei, B., Dai, B., Ouyang, W., Qiao, Y., Dong, C.: DiffBIR: Towards blind image restoration with generative diffusion prior. arXiv preprint arXiv:2308.15070 (2023)
36. Liu, A., Liu, Y., Gu, J., Qiao, Y., Dong, C.: Blind image super-resolution: A survey and beyond. IEEE TPAMI **45**(5), 5461–5480 (2022)
37. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017)
38. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
39. Lugmayr, A., Danelljan, M., Timofte, R.: NTIRE 2020 challenge on real-world image super-resolution: Methods and results. In: CVPRW. pp. 494–495 (2020)
40. Lugmayr, A., Danelljan, M., Timofte, R., et al.: AIM 2019 challenge on real-world image super-resolution: Methods and results. In: ICCVW. pp. 3575–3583 (2019)

41. Maeda, S.: Unpaired image super-resolution using pseudo-supervision. In: CVPR. pp. 291–300 (2020)
42. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: CVPR. pp. 3517–3526 (2021)
43. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE SPL **20**(3), 209–212 (2012)
44. Mou, C., Wu, Y., Wang, X., Dong, C., Zhang, J., Shan, Y.: Metric learning based interactive modulation for real-world super-resolution. In: ECCV. pp. 723–740 (2022)
45. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: ECCV. pp. 191–207 (2020)
46. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: NeurIPS. pp. 6309–6318 (2017)
47. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: a technical overview. IEEE SPM **20**(3), 21–36 (2003)
48. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML. pp. 8821–8831 (2021)
49. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: NeurIPS. pp. 14866–14876 (2019)
50. Sahak, H., Watson, D., Saharia, C., Fleet, D.: Denoising diffusion probabilistic models for robust image super-resolution in the wild. arXiv preprint arXiv:2302.07864 (2023)
51. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. IEEE TPAMI **45**(4), 4713–4726 (2022)
52. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. pp. 3147–3155 (2017)
53. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: NTIRE 2017 challenge on single image super-resolution: Methods and results. In: CVPRW. pp. 114–125 (2017)
54. Timofte, R., Gu, S., Wu, J., Van Gool, L.: NTIRE 2018 challenge on single image super-resolution: Methods and results. In: CVPRW. pp. 852–863 (2018)
55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 6000–6010 (2017)
56. Wang, J., Chan, K.C., Loy, C.C.: Exploring CLIP for assessing the look and feel of images. In: AAAI. pp. 2555–2563 (2023)
57. Wang, J., Yue, Z., Zhou, S., Chan, K.C., Loy, C.C.: Exploiting diffusion prior for real-world image super-resolution. IJCV (2024)
58. Wang, L., Wang, Y., Dong, X., Xu, Q., Yang, J., An, W., Guo, Y.: Unsupervised degradation representation learning for blind super-resolution. In: CVPR. pp. 10581–10590 (2021)
59. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In: ICCVW. pp. 1905–1914 (2021)
60. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: CVPR. pp. 606–615 (2018)
61. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.C.: ESRGAN: Enhanced super-resolution generative adversarial networks. In: ECCVW. pp. 63–79 (2019)
62. Wei, P., Xie, Z., Lu, H., Zhan, Z., Ye, Q., Zuo, W., Lin, L.: Component divide-and-conquer for real-world image super-resolution. In: ECCV. pp. 101–117 (2020)

63. Wei, Y., Gu, S., Li, Y., Timofte, R., Jin, L., Song, H.: Unsupervised real-world image super resolution via domain-distance aware training. In: CVPR. pp. 13385–13394 (2021)
64. Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.H., Liao, Q.: Deep learning for single image super-resolution: A brief review. IEEE TMM **21**(12), 3106–3121 (2019)
65. Zhang, K., Liang, J., Van Gool, L., Timofte, R.: Designing a practical degradation model for deep blind image super-resolution. In: ICCV. pp. 4791–4800 (2021)
66. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV. pp. 3836–3847 (2023)
67. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
68. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV. pp. 286–301 (2018)
69. Zhang, Z., Ma, J., Zhou, C., Men, R., Li, Z., Ding, M., Tang, J., Zhou, J., Yang, H.: M6-UFC: Unifying multi-modal controls for conditional image synthesis via non-autoregressive generative transformers. arXiv preprint arXiv:2105.14211 (2021)
70. Zhou, H., Zhu, X., Zhu, J., Han, Z., Zhang, S.X., Qin, J., Yin, X.C.: Learning correction filter via degradation-adaptive regression for blind single image super-resolution. In: ICCV. pp. 12365–12375 (2023)
71. Zhou, S., Chan, K., Li, C., Loy, C.C.: Towards robust blind face restoration with codebook lookup transformer. In: NeurIPS. pp. 30599–30611 (2022)
72. Zhou, S., Zhang, J., Zuo, W., Loy, C.C.: Cross-scale internal graph neural network for image super-resolution. In: NeurIPS. pp. 3499–3509 (2020)
73. Zhou, Y., Li, Z., Guo, C.L., Bai, S., Cheng, M.M., Hou, Q.: SRFormer: Permuted self-attention for single image super-resolution. In: ICCV. pp. 12780–12791 (2023)