Conditioned Diffusion Model for Image Super-Resolution

Zijian HE,A59002893
UC San Diego
zihe@ucsd.edu
Github link.

1 Introduction

Deep generation models have been widely developed in recent years. In general, generative models can be categorized by 3 kinds: Generative adversarial networks (GANs), variational autoencoders (VAEs) and Diffusion Model. These models have different advantages can should be chosen by applications. For example, GAN has fast sampling rate and high quality sample while diffusion model has high diversity/mode and generation quality [8] Some novel diffusion models are proved to have better quality compare to best GAN models. Diffusion model also have advantage in training comparing to GAN because GAn suffers from mode collapse without delicate finetune. The only limit is slow sampling speed, since diffusion takes small step at each time. Based on above reason, we choose diffusion model as out generative model.

The diffusion model can be classified into 3 categories: Denoise Diffusion Probabilistic Model (DDPM), Noise Conditioned Score Network (NCSN) and Stochastic Differential Equations (SDE). Our model is based on DDPM which can be viewed as a special type of hierarchical VAE. The difference from VAE are fixed encoder, the same latent variable dimensions and shared denoising model among all time steps [6]. The diffusion model consists of 2 parts: (1)forward diffusion stage: Input data is corrupted by Gaussian noise at each step and becomes the pure Gaussian noise at the end. (2) reverse diffusion: A network is used to recover the original image from the noise introduced before step by step which is the reversed diffusion process.

Inspired by the excellent work of SR3 (Super-Resolution via Repeated Refinement)[10], we believe the diffusion model is suitable for the super-resolution (SR) task and our proposed method is based on SR3. SR3 is a new approach to conditional image generation based on DDPM model. SR3 uses a U-Net like architecture to iteratively remove different levels of noise from the output. Another difference from the vanilla model is SR3 can choose different magnification factors for outputs which means cascade is possible. In this case several smaller models are trained individually and each network has less steps which contribute to higher overall quality.

SR task can be evaluated with PSNR and SSIM score which measures the difference between outputs and the ground truth in pixel level. The author of SR3 argued these metrics are not suitable for SR since they penalize more for high-frequency details and the network might produce semantically similar but blur outputs as shortcuts. The author proposed human judgment(fool) which is evaluated by humans to judge if the image is synthesized or natural. However, it is infeasible for this project to finish and an alternative should be found. Therefore we decide to include Fréchet Inception Distance (FID) score [3].The FID uses a trained inception V3 network to extract the feature of both original and synthetic images and compare their value. Another metric is consistency, which measures the L2 distance between the original and generated images.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

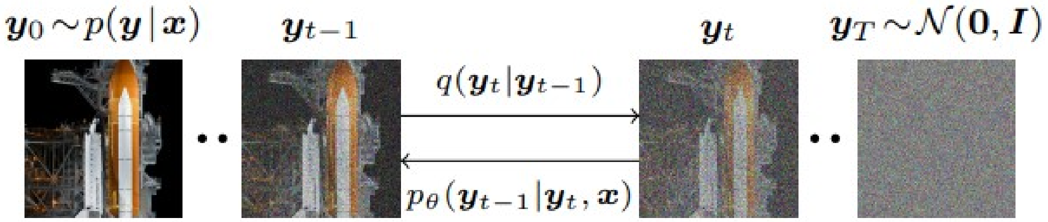


Figure 1: Forward and reverse diffusion process conditioned on image x

2 Conditional Denoising Diffusion Model

This section will cover details in both diffusion and model architecture. The dataset for SR task has the form of $D = \{x_i, y_i\}_{i=1}^N$ which is the input and output pair and we wish to learn the map from the source to the target image by parametric approximation of $P(y|x)$. Figure 1 explains the whole conditional diffusion process. Starting from the diffusion process q gradually add noise to the image y_0 and from noise $y_T \sim N(0, I)$, the model refines the image through iterative process $(y_T, y_{T-1}, ...y_0)$ condition on x such that $y_0 \sim p(y|x)$.

2.1 Gaussian Diffusion Process

We can define a forward Markovian diffusion process as q. If we marginalizing all intermediate states, we get

$$q(y_t|y_0) = N(y_t|, \sqrt{\gamma_t}y_0, (1-\gamma_t)I) \tag{1}$$

$$\tilde{y} = \sqrt{\gamma_t}y_0 + (1-\gamma_t)\epsilon, \epsilon \sim N(0, I) \tag{2}$$

Where $\gamma_t = \prod_{i=1}^t \alpha_i$ and α_i is the variance schedule for each step. The forward process changes the output to pure Gaussian noise at step T. Also $\beta_t = 1 - \alpha_t$. One can derive the posterior distribution of y_{t-1} by

$$q(y_{t-1}|y_0, y_t) = y_{t-1}\mu + \epsilon\sigma \tag{3}$$

$$\mu = \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t}y_0 + + \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t}y_t \tag{4}$$

$$\sigma^2 = \tilde{\beta}_t = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t} \tag{5}$$

The above equations are used to sample y_0 during reverse process. Noticethe original DDPM implementation use B_t as posterior variance which almost has no influence.

2.2 Denoising Model Optimization

To recover the original image, we introduce the neural denoising model $f_\theta(x, \tilde{y}, \gamma)$ which takes input with source image, noisy image and variance of noise and aims to recover the noise vector ϵ. Specifically, the denoising function takes the input which consists of low-resolution references as well as noisy high-resolution image. The noisy image here is obtained from 1. The final training objective is

$$E_{(X,Y)}E_{\epsilon\gamma} = ||f_\theta(x, \tilde{y}, \gamma) - \epsilon||_p^p \tag{6}$$

where $P \in 1, 2$, X,Y are from dataset and γ is standard distribution. The training objective can be derived by optimizing the negative likelihood of generated result which has the variational bound. The original paper choose another simplified objective which lead to the sub-optimal log-likelihood.

2.3 Iterative Inference

Inference can be viewed as a reverse Markovian process that recovers y_0 from y_T. We can form expression of y_0 by rearranging terms in 2. Substitute y_0 into posterior$q(y_{t-1}|y0, y_t)$ we get $\mu_\theta(x, y_t, \gamma_t)$

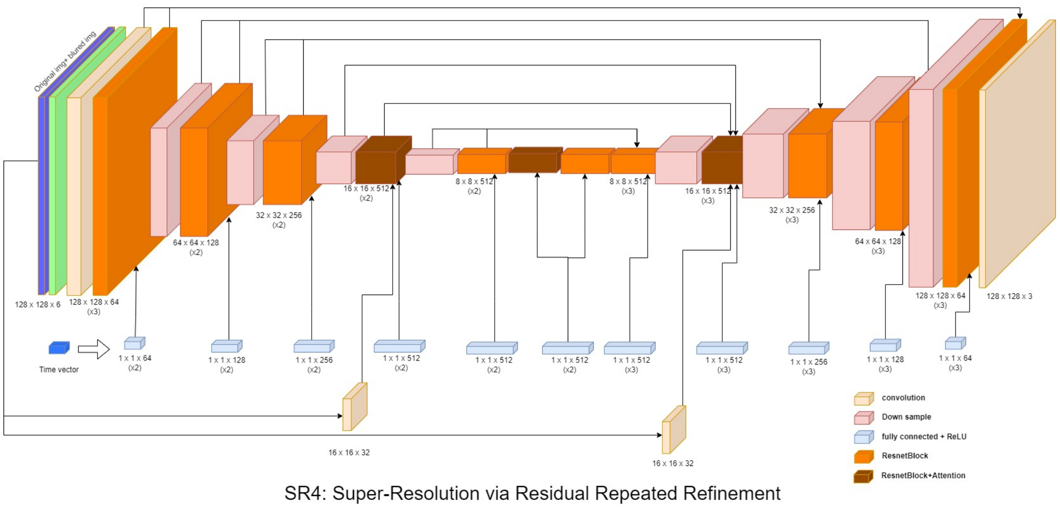


Figure 2: Proposed network structure

as parameterized mean in reverse process. If we set the variance in the reverse process to be equal to the one in forward process, we get the following inference equation.

$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(x, y_t, \gamma_t)) + \sqrt{1-\alpha_t}\epsilon_t \tag{7}$$

The above equation is equivalent to sampling a less noisy image at each step by using the posterior mean and variance derived above. However, in theory, the reversed process can not be solved directly. In the original SR3 paper, the author chooses a constant $\tilde{\beta}_t$to replace the variance. However, this is true only when the number of steps is large[7]. Also, the iteration steps are set to be 2000 and slow sampling is a major drawback of DDPM, which suggest the more efficient methods in sampling.

2.4 Model Architecture

The author of SR3 follows the U-Net design as the previous DDPM. However, residual blocks are used following [2]'s design with modification on skip connections. Input image x is upsampled by the bicubic method.

3 Related Work

3.1 DDPM

Denoising Diffusion Probabilistic Models(DDPM) is thoroughly investigated by Ho, et al[5]. The authors combine diffusion probabilistic model with d Langevin dynamics to train the novel diffusion model for image generation. DDPM outperforms the SOTA methods during the same time such as GAN. However, the main drawback is the slow sampling rate to generate new images, which takes as long as 2000 steps. Therefore, the following researches focus on faster sampling and better image quality. Another trend is conditional diffusion model, which contain more information in the reverse process than the single noisy image.

3.2 Super Resolution

Super-resolution is a classical problem in computer vision. It can be solved by traditional, machine learning (regression-based), and deep learning methods. One example of a successful attempt in DL field is SRCNN[1]. After that, other advanced techniques such as attention and residual connection are used. However, humans cannot perceive the fine difference in pixel value but focus on semantic information. Also information is lost during downsampling so generative models are introduced to fill this gap. SR3 [10] is a diffusion model conditioned on a low-resolution image and output a high-resolution one. Since the vanilla DDPM has generation ability, SR3 uses the image as condition to control the result of DDPM.

SR is an important task especially in diffusion models. For example, Imagen[11],it an Text-to-Image Diffusion Models that use text as the condition to generate low-resolution image. several SR modules are then perform upscaling to generate final high-resolution image. Therefore, SR can be a crucial module in various image-generation tasks. Compatibility with module cascades is an important metric for SR model. Overall, SR task requires to use given information (low-resolution image) as the condition and adding extra information to reconstruct the original image. These 2 factors determine the quality of SR results.

4 Experiments

We evaluate the proposed model in several perspectives. Since our model is based on SR3, we set SR3 as the baseline. However, there is no official implementation and the unofficial is sub-optimal. For example, the paper lack several details and the large batch size is impossible to follow. Therefore, our target is then obtain the results close to the original implementation while using less computation. Specifically, the experiments include:

1. Train model for face-super resolution task from 16×16→128×128 with FFHQ and evaluated on CelebA- HQ.

We use multiple metrics for evaluation of generated images which include Structural SIMilarity (SSIM), peak signal-to-noise ratio (PSNR) and Frechet Inception Distance(FID). SSIM and PSNR is positively correlated with image quality while FID is the opposite. The evaluation of SR task is hard for generative model. Traditionally SR task rely on PSNR and SSIM to measure the difference in pixel level, which might penalize high-fidelity image generation since the nonalignment of details leads to lower scores. FID uses inception net to extract the semantic information of images which is more reasonable to be compared. However, our test result shows FID of generated portrait images is over 20, because training dataset of Inception net is not.

During training, we choose lr=1e-4, batch size=4, training iterations = 64,000,000, and downsampling scheme to be 128x128 to 16x16.

5 Proposed methods

DDPM is a general model and can be improved in many ways. It has been shown above conditional diffusion is different from the vanilla model from the noise vector simulation part. The condition is used and for SR task, the condition is a blurred image. In the following sections, several proposed improvements will be discussed.

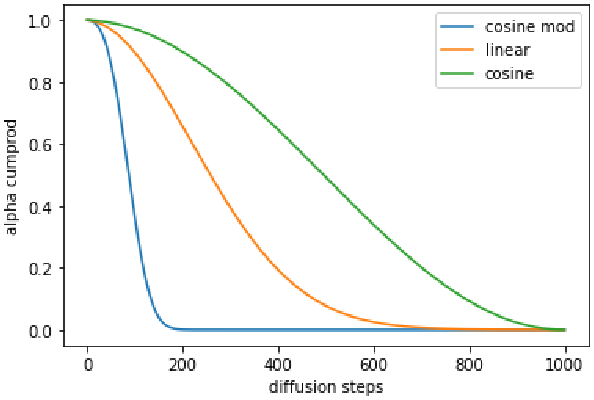
5.1 Attention-based inputs fusion

The first idea is a modification of net structure. Inspired by the AHDRNet[4], additional attention module is built at the beginning of network. Specifically, noisy image and low-resolution image are treated as query and key and are fed into 2 individual conv layers to produce an attention matrix, which is compressed to (0,1) range and multipied with value (low-resolution image). Finally 2 images are concatenated for later use. Ideally, the attention module should preserve important information such as edges. However, the proposed method produce PSNR score of 18.5. The problem is found by checking the attention map.Even after the model has converged, the attention map has a mean of 0.5 and a variance close to 0. Therefore, SR is a global task that the low-resolution image provides semantic information.

5.2 Improved noise schedule

The second idea is to modify the design of DDPM. Specifically, the linear noise schedule is proved to be not suitable for DDPM[7], because the later parts of diffusion process produce the result close to Gaussian noise. Therefore, the Cosine beta schedule is introduced to have linear property in the middle. It is proved that beta scheduler is more suitable for low-resolution generations and

Figure 3: Comparison of noise schedule



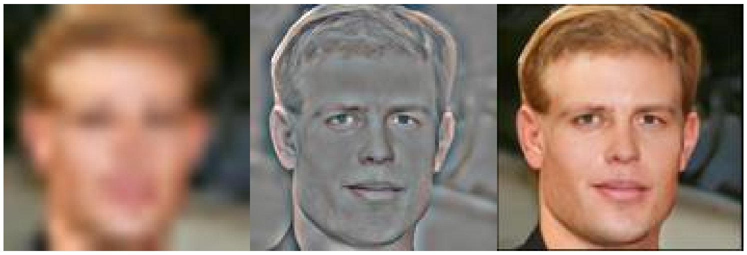


Figure 4: Generated image with residual prediction

fewer diffusion steps are required. Figure 3 show the difference of noise schedule in terms of alpha cumulative product. Notice the cosine schedule is flatter than other 2 variations at the beginning which means it preserves more details in diffusion steps.

5.3 Learnable posterior variance

Posterior variance is fixed in the original SR3 and we can learn the optimal weight by mixing $\tilde{\beta}_t$ and β_t. Specifically, we add the VLB term to the loss function as well.

$$L_{Hybrid} = L_{Simple} + \lambda L_{VLB} \tag{8}$$

Where λ is the blending weight and can be learned through training. However, the convergent speed is much slower than the baseline so we decide not to use the mixed loss function. Previous research also shows it might be suitable for longer training time (1,000,000 iterations).

5.4 Indirect generation by residual prediction

SR3 takes input of the low resolution image as the constraint for the reverse diffusion process. Inspired by SRDiff[9], we notice input of SR task is only low resolution images and the output is high-resolution images. The DDPM here provides the lost information. Instead of direct image generation, we can also predict the difference between low and high-resolution images. It is similar to the difference of Gaussian which is an edge-enhanced technique in image processing. The residual information mainly contain high frequency information and is potentially easier for the network to learn.

$$X_{SR} = X_0 + Upsample(X_L) \tag{9}$$

X_0 here is the output of the reverse process. The figure 10 shows how the residual prediction combines with the low-resolution image. Notice the residual is scaled to 0-1 for display. The residual has a smaller mean square value than high-resolution image which might be easier to train.

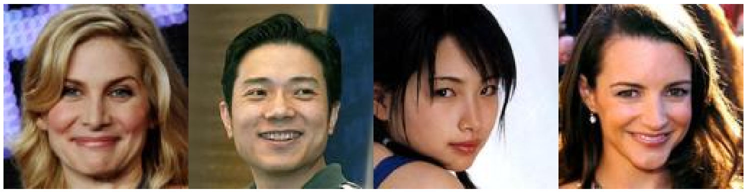


Figure 5: Original image

5.5 Modification from SR3

Our method is based on the unofficial SR3 implementation which has a lower PSNR score than stated in the paper. There are a few major difference from the SR3 paper. Posterior variance is set to be $\tilde{\beta}_t$ instead of β_t. Also, linear warm up lr is used. The biggest difference is batch size, which is set to be 4 instead of 256. Hence we use the same random noise for all samples in a mini-batch instead of different noise. It has minor impact when batch size is relatively small. Therefore the network tends to has sub-optimal performance compared to the paper with the same iterations (1,000,000).

5.6 Final proposed method: SR4

According to above experiments, we proposed Super-Resolution via Iterative Residual Refinement (SR4), which adapt the warm-up lr scheduler and cosine noise scheduler. Also, it uses residual prediction tasks for faster convergence. Different from SR3, the proposed method uses multi-head attention module in the middle (8x8 size) rather than near the middle (16x16) which has the potential to extract more details. Also inspired by the consistency score mentioned above, we believe it's important to optimize this metric. However, consistency cannot be optimized directly because there are hundreds of steps before SR image is generated. Therefore, we decide to indirectly optimize this metric by adding a residual connection between aided low-resolution image and middle layers. Since the middle layer has smaller tensor size, we test several downsample schemes which include direct downsample, bicubic sampling, tghe single large kernel convolution and several smaller convolutions as well as activation functions. It turns out the last method works well while achieving a large reception field and does not increase training time significantly. In comparison, bicubic interpolation fails to preserve details to smaller input sizes (128x128 to 16x16) we so believe learnable downsampling scheme is the most effective way. The whole model is described by graph 2.

Specifically, input of UNet is the up-sampled image and results from the previous iteration. Then, downsample layer and Resnet blocks are used for several times and the input size is reduced to 8x8 in the bottleneck layer.Resnet layer consist of several smaller blocks with group normalization as well as Swish activation. Group normalization is proved to be effective when batch size is small(4 in our case). Channel has increased from 6 to 512 accordingly. Some of the Resnet blocks are equipped with multi-head attention modules and we decide to use in middle layers. Several global residual connections are used as well which is shown in the graph. Time vector is reshaped to different positions as the embedded information. inspired by this design, SR4 also add additional information from the low-resolution inputs to 2 locations (16x16 shape input).

6 Results and discussion

Table 1 list results of all experiments. Notice we achieve the similar result as the original implementation with 640,000 iterations. Then we prove the degradation effect of fewer diffusion steps (500) and the effectiveness of the cosine scheduler under this condition. In terms of hybrid loss function, it has worse metrics because longer training time and less trained epochs. Also it's more suitable for the model with less diffusion steps (<100). For the attention based method, it has worse performance than baseline as well. Residual prediction task tends to perform better and the trend is more obvious when residual input of low-res image is used (SR4).

The similar trend can be seen from the generated examples. Notice the cosine diffusion schedule + linear warm-up produced better result than the baseline with fewer diffusion steps required(500) step. Residual prediction task preserves more details than baselines.

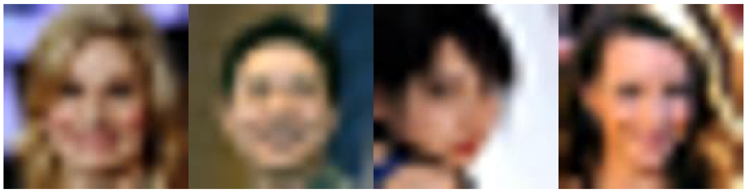


Figure 6: Low resolution image

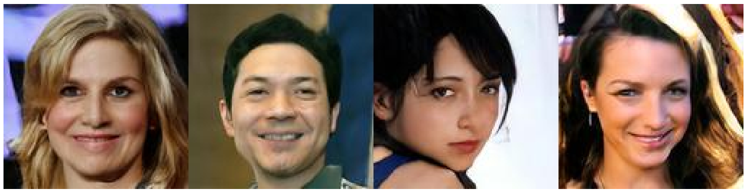


Figure 7: baseline (SR3)

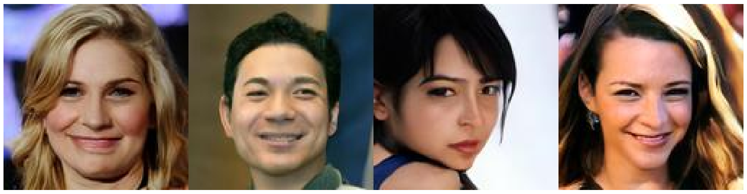


Figure 8: Cos scheduler

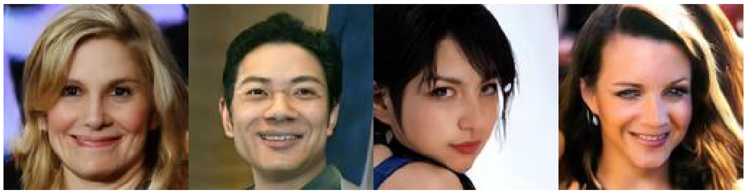


Figure 9: Residual prediction task

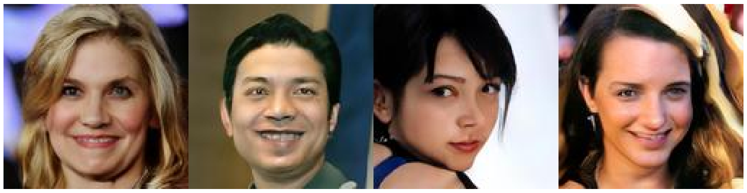


Figure 10: Residual input to middle layer (SR4)

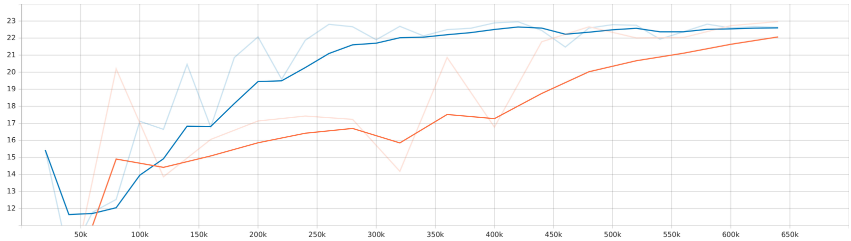


Figure 11: Residual prediction (blue) and baseline (orange) PSNR training curve

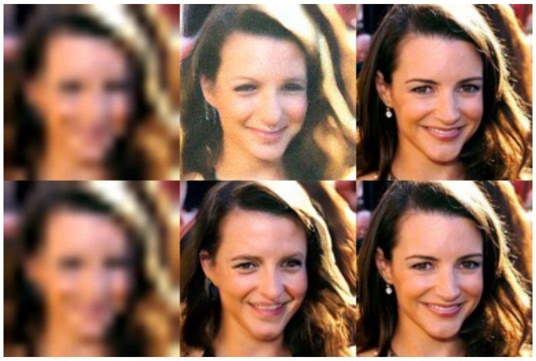


Figure 12: Residual prediction and baseline at 440,000 steps

6.1 Residual prediction at early stage

It can be found from figure12 that the Residual prediction task preserves much more details than the baseline (cosine scheduler) at 440,000 steps. The trend can be found even though the model is fully trained. This implies a faster convergence speed that is meaningful for mobile applications.

6.2 SR4 discussion

Overall, SR4 outperforms all other methods in every metric. The better consistency score can be explained by the extra indirect optimization goal on this score. Also, it has a faster convergence rate and might be useful for mobile applications. Also, notice the step size of all methods are 500 except for the baseline which is 2000. Therefore, the proposed methods can be suitable to real-time applications.

7 Conclusion

We present the SR4, which is based on the previous model: SR3. The proposed methods have similar general idea of using the low resolution as input to regulate generated images from the diffusion

Table 1: Evaluation results

Model	PSNR	SSIM	Consistency
SR3(baseline)	23	0.68	55.68
SR3(500 steps)	22.4	0.66	60.42
Simple+VLB loss	18.6	0.61	
Cos schedule	23.1	0.69	52.27
Attention input	18.8	0.63	
Res predict	23.1	0.68	54.31
SR4(Combine)	**23.1**	**0.69**	**51.75**

model. To speed up both training and inference time, we adopt the warm up lr scheduler, cosine noise scheduler and change the training objective to residual prediction between high and low-resolution images. We test the model on several metrics: SSIM and PSNR which outperform other methods. To enhance the consistency score performance, we construct the residual connection of inputs to the middle layer which works well. We believe SR is hard to be evaluated by a single metric and more comprehensive methods should be proposed.

References

[1] Chao Dong et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. “Large scale GAN training for high fidelity natural image synthesis”. In: *arXiv preprint arXiv:1809.11096* (2018).

[3] Ali Borji. “Pros and cons of gan evaluation measures”. In: *Computer Vision and Image Understanding* 179 (2019), pp. 41–65.

[4] Qingsen Yan et al. “Attention-guided network for ghost-free high dynamic range imaging”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1751–1760.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[6] Arash Vahdat and Jan Kautz. “NVAE: A deep hierarchical variational autoencoder”. In: *Advances in neural information processing systems* 33 (2020), pp. 19667–19679.

[7] Alexander Quinn Nichol and Prafulla Dhariwal. “Improved denoising diffusion probabilistic models”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8162–8171.

[8] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. “Tackling the generative learning trilemma with denoising diffusion GANs”. In: *arXiv preprint arXiv:2112.07804* (2021).

[9] Haoying Li et al. “Srdiff: Single image super-resolution with diffusion probabilistic models”. In: *Neurocomputing* 479 (2022), pp. 47–59.

[10] Chitwan Saharia et al. “Image super-resolution via iterative refinement”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[11] Chitwan Saharia et al. “Photorealistic text-to-image diffusion models with deep language understanding”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494.