Can Diffusion Model Generalize Well in Image Super Resolution with Limited Fine-Tuning?

Kejia Yin

Abstract—Image Super Resolution is an attracting computer vision task, and many deep learning based methods achieved outstanding performance, e.g. GANs. Very recently, Diffusion Probabilistic Model, a relatively new kind of generative model, also demonstrates its great potential to deal with this task. Despite the excellent results on their training and testing sets, almost nobody has shown existing methods' ability to generalize to new domain images. In this work, we take SR3 as an example and evaluate its generalization ability with limited fine-tuning steps, new domain training data and range of time steps both qualitatively and quantitatively. We also evaluate whether naive fine-tuning will impair the model's original performance.

Index Terms—Diffusion Probabilistic Model, Fine-tuning, Image Super Resolution

1 Introduction

Image Super Resolution is one image-to-image translation task, which aims to construct high-resolution (HR) image from low-resolution one [1]. Like other inverse problems in computer vision, e.g. image in-painting, SR is a hard problem because given a LR image there are multiple corresponding HR counterparts and vice versa.

The success of deep generative models demonstrates its ability to model complex image distribution, and have been applied to SR as solving a conditioned generation task, especially for generative adversarial networks (GANs) based methods [2]. Though GANs achieved relatively good performance, they typically require specially designed regularization and optimization tricks to deal with training instability and mode collapse. Recently, diffusion probabilistic model based method thrives and achieved comparable performance with SOTA methods in computer vision tasks. SR3 [3] is a simple SR method adapted from Denoising Diffusion Probabilistic Model (DDPM) [4], which simply minimizes a well-defined loss function and do not rely on those regularization and optimization tricks from GANs.

Even though there are various of generative models and they performed excellent with their testing data, it's still a question that whether those generative models can generalize well to the data beyond their training and testing datasets. Unfortunately, related research on the generalization ability of SR models are almost absent, and there only exist one research tries to benchmark it [5]. Generalization ability is a critical aspect of deep learning models and there are studies point out existing SR model may not be able to generalize well to new domain data [6]. As shown in Figure 1, the pre-trained SR3 model can obtain good SR result on FFHQ and CelebA-HQ, which are the original training and testing datasets, but fails to construct satisfying SR image on animation faces. We can see there are obvious distortions in the SR animation face.

In this work, we want to explore the generalization ability of SR3 through limited fine-tuning, and mainly answer the following questions:

1) Can pre-trained SR3 be directly applied to new domain of data? We can clearly see in Figure 1 that this may not give us desired outcome.
2) Can we fine-tune the pre-trained model for few steps in order to have good SR results? Training from scratch often takes too much time which is not acceptable, however, fine-tuning the pre-trained model is much cheaper and probably can yield good result.
3) Can we fine-tune the pre-trained model with limited amount of new domain data in order to have good SR results? It's not enough if the pre-trained model can be fine-tuned with few steps, we also want it to be fine-tuned with as few new domain data as possible, because we may not have much new domain data before and it's costly to collect the data.

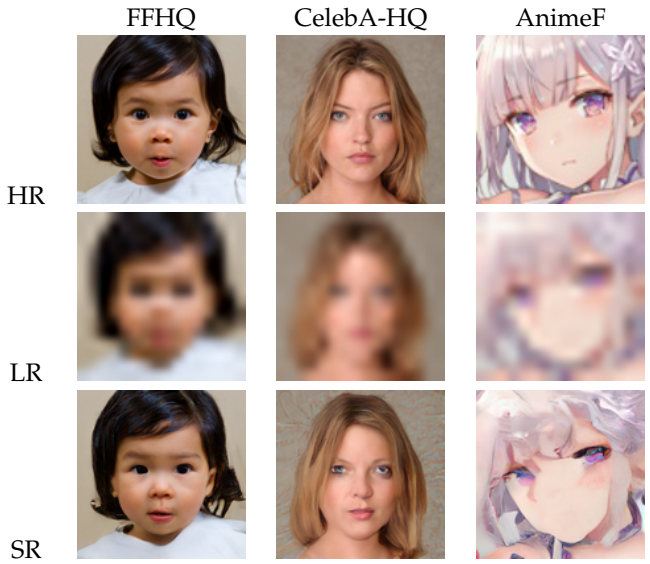

Fig. 1: HR, LR and SR images using pretrained model on different Datasets. The performance of pretrained model degrades when there exists domain gap between training and testing data.

4) Do we need to fine-tune the model with all the timesteps? DM generates new data by transform the simple distribution through a Markov Chain process step by step, and the simple distribution will gradually become our desired empirical image distribution. Since we are starting from the same simple distribution for different image domains, the later steps should share more similarities than the starter steps, and the domain gap between new data and pre-trained data may be aligned by fine-tune the starter steps only.
5) Will the fine-tuning harm the performance on original datasets? Ideally, we want our model could deal with all the tasks well at the same time, and it may not be satisfying if fine-tuning will sacrifice its original performance.

2 Related Work

In this section, we will quickly review some related work in image super resolution and diffusion probabilistic model.

Super Resolution: Image Super Resolution (SR) is an important computer vision task, which aims to generate high-resolution (HR) image from low-resolution (LR) image [1]. The final objective can be considered to learn a function that can reverse the degradation process from HR to LR images. There are numerous existing works addressing this issue, and can be roughly divided into two categories: real-world image pair based method, synthesized image pair based method. Note that we mainly focus on deep learning methods for SR.

For real-world image pair based methods, they require HR and LR image pairs captured in the real-world by adjusting the focal length and other parameters of real camera [7]. This is beneficial because the model have the chance to directly learn the degradation process in the real-world. However, collecting such data is not easy because you have to keep the other conditions the same while taking photos with different resolutions.

Synthesized image pair based methods also required paired HR and LR image, but the LR image is synthesized by algorithms, e.g. bicubic. This makes it easy to collect HR-LR image pairs, because you only need to have the HR image. And methods trained with synthesized image pairs also achieved astonishing super resolution results [2]. Within this category, we can even done super resolution only with LR images. ZSSR is a zero-shot super resolution method which explores the internal recurrence of information inside the LR image during the test time [8]. More specifically, ZSSR synthesized LRLR image from testing image, and utilize LRLR-LR image pair to train a small image-specific CNN at test time.

Diffusion Probabilistic Model: Diffusion Probabilistic Model (DM) was first introduced by [9] in 2015, and plays an increasingly important role in computer vision tasks after the success of DDPM [4]. DM is a kind of generative models which transform the data from a simple distribution, e.g. Gaussian, to our desired distribution following a Markov Chain process. DDPM directly works on images to perform unconditional image generation. SR3 adopted similar architecture with DDPM, but conditioned on LR image to perform image super resolution [3]. Rather than directly works on pixel spaces, latent diffusion models (LDMs) applied DM in latent space by adding an encoder and a decoder to transform the image pixels to its latent representations [10]. In this work, we mainly focus on the generalization ability of SR3, which is the most simple DM based SR method, and leave the discussion of LDMs for future work.

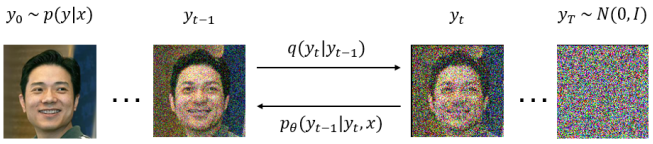

Fig. 2: From left to right is the diffusion process, which gradually add noise to the image. From right to left is the inference process which iteratively recover the image from noise. Note that the condition x is not shown in the figure.

3 SR3

In this section, we will briefly review the basic concepts of DM and how SR3 works.

3.1 Conditioned Diffusion Model

Consider a conditional distribution $p(y|x)$, and we are given samples drawn form this distribution, represented as input and output image pairs $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$. This is a one-to-many mapping problem, meaning given a certain x, we have multiple corresponding ys, and our goal is to approximate the conditional distribution $p(y|x)$ with a function f_θ, where θ denoted the parameters of the model. In image super resolution tasks, input image and output image pairs are LR and HR images pairs.

SR3 follows the idea of DDPM and adapts it to this conditional distributed problem. The diffusion process and reverse process are illustrated in Figure 2. From left to right is the diffusion process, which gradually transforms $y_0 \sim p(y|x)$ to pure noise image $y_T \sim N(0, I)$ by adding noise at each time step t. From right to left is the reverse process, which iteratively recovers the output image. Suppose both diffusion and reverse process satisfies Markov Chain assumption, denoted by $q(y_t|y_{t-1})$ and $p_\theta(y_{t-1}|y_t, x)$ respectively. Our goal is to learn the function f_θ, which can model the distribution $p_\theta(y_{t-1}|y_t, x)$ well.

3.2 Gaussian Diffusion Process

In SR3, the diffusion process is defined as Gaussian Diffusion Process which adds Gaussian noise at each time step. Since diffusion process follows Markov Chain assumption, we define q as the following:

$$q(y_{1:T}|y_0) = \prod_{t=1}^{T} q(y_t|y_{t-1}), \tag{1}$$

$$q(y_t|y_{t-1}) = N(y_t|\sqrt{\alpha_t}y_{t-1}, (1-\alpha_t)I), \tag{2}$$

where α_t is hyper-parameter, subject to $0 < \alpha_t < 1$, which defines the variance of the noise added at each time step.

Algorithm 1 Optimizing Process of SR3

1: **repeat**
2: Sample (x, y_0) from Dataset
3: Sample time step $t \sim U(1, MAX\_TIMESTEP)$
4: Sample noise $\epsilon \sim N(0, I)$
5: Take gradient step on

$$\nabla\theta \| f_\theta(x, \sqrt{\beta_t} y_0 + \sqrt{1-\beta_t}\epsilon, \beta_t) - \epsilon \|_p^p$$

6: **until** converged

Algorithm 2 Inference Process of SR3

1: Sample $y_T \sim N(0, I)$
2: **for** $t = T, ..., 1$ **do**
3: Sample noise $\epsilon \sim N(0, I)$ if $t > 1$, else $\epsilon = 0$
4:

$$y_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\beta_t}} f_\theta(x, y_t, \beta_t)) + \sqrt{1-\alpha_t}\epsilon$$

5: **end for**
6: **return** y_0

And according to Eq.1 and Eq. 2 and Markov Chain, we can derive $q(y_t|y_0)$ as:

$$q(y_t|y_0) = N(y_t|\sqrt{\beta_t} y_0, (1-\beta_t)I), \tag{3}$$

where $\beta_t = \prod_{i=1}^{t} \alpha_i$. Moreover, as illustrated in [4], we can derive the posterior distribution of y_{t-1} given (y_0, y_t) as

$$\begin{aligned} q(y_{t-1}|y_0, y_t) &= N(y_{t-1}|\mu, \sigma^2 I) \\ \mu &= \frac{\sqrt{\beta_{t-1}(1-\alpha_t)}}{1-\beta_t} y_0 + \frac{\sqrt{\alpha_t}(1-\beta_{t-1})}{1-\beta_t} y_t \\ \sigma^2 &= \frac{(1-\beta t-1)(1-\alpha_t)}{1-\beta_t}. \end{aligned} \tag{4}$$

With this posterior distribution, we can parameterize the reverse process and it helps us formulate the loss function in later sections.

3.3 Optimizing SR3

In DDPM, it treats the our function f_θ as a noise predictor, which means given a specific time step t and the corresponding noisy output image, the function should predict the noise used to generate that noisy image. SR3 follows the same idea, and with Eq.3, given the original output image y_0, we can derive the noisy output image at time step t as:

$$\tilde{y_t} = \sqrt{\beta_t} y_0 + \sqrt{1-\beta_t}\epsilon, \quad \epsilon \sim N(0, I). \tag{5}$$

Based on Eq.5, we can rewrite our denoising model as $f_\theta(x, \tilde{y}, \beta)$, which takes the conditioned input image x, the sufficient statistics for the variance of the noise β and sampled Gaussian noise ϵ as input. Note that the time step t used in DDPM's optimization is embedded into β in SR3's optimization. Our objective is to let $f_\theta(x, \tilde{y}, \beta)$ predict the sampled noise ϵ, and the objective function can be written as

$$\mathbb{E}_{(x,y)} \mathbb{E}_{\epsilon,\beta} \| f_\theta(x, \sqrt{\beta} y_0 + \sqrt{1-\beta}\epsilon, \beta) - \epsilon \|_p^p, \tag{6}$$

where (x, y) is sampled from training dataset, and $p \in \{1, 2\}$. In [4], it justifies that when $p = 2$, the objective function can be interpreted as the variational lower bound on the marginal log-likelihood of $\mathbb{E}_{x,y} \log p_\theta(y|x)$. The whole optimizing process is illustrated in Algorithm1. When $p = 1$, it becomes L1 loss, and when $p = 2$, it becomes L2 loss.

3.4 Inference with SR3

Inference with SR3 is the reverse process of diffusion model. This process starts with a pure Gaussian noise y_T, and can be formulated as the following:

$$p_\theta(y_{0:T}|x) = p(y_T) \prod_{t=1}^{T} p_\theta(y_{t-1}|y_t, x) \tag{7}$$

$$p(y_T) = N(y_T|0, I) \tag{8}$$

$$p_\theta(y_{t-1}|y_t, x) = N(y_{t-1}|\mu_\theta(x, y_t, \beta_t), \sigma_t^2 I). \tag{9}$$

The inference process is defined as isotropic Gaussian conditional distributions, $p_\theta(y_{t-1}|y_t, x)$, which are learned.

Our denoising model f_θ is trained to predict noise ϵ, and if we substitute the noise in Eq.5 with the predicted noise, we can approximate y_0 as:

$$\hat{y_0} = \frac{1}{\sqrt{\beta_t}}(y_t - \sqrt{1-\beta_t} f_\theta(x, y_t, \beta_t)). \tag{10}$$

Recall the Eq.4 in previous section, we can substitute the estimated $\hat{y_0}$ into the posterior distribution $q(y_{t-1}|y_0, y_t)$, and this gives us the mean of $p_\theta(y_{t-1}|y_t, x)$ as:

$$\mu_\theta(x, y_t, \beta_t) = \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\beta_t}} f_\theta(x, y_t, \beta_t)). \tag{11}$$

Note that the variance of $p_\theta(y_{t-1}|y_t, x)$, σ_t^2, is still not defined, we set this variance to $(1-\alpha_t)$, which is the same as the diffusion process.

Now we can estimate y_{t-1} at each time step in the reverse process through parameterization as:

$$y_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(y_t - \frac{1-\alpha_t}{\sqrt{1-\beta_t}} f_\theta(x, y_t, \beta_t)) + \sqrt{1-\alpha_t}\epsilon_t, \tag{12}$$

where $\epsilon_t \sim N(0, I)$. After iteratively estimate y_{t-1} from y_t, x and f_θ, we can eventually recover the output image.

3.5 SR3 Model Architecture

SR3 adopts a modified DDPM's U-Net architecture, which mainly replaces and increases the residual blocks used in DDPM, and re-scales skip connections by $\frac{1}{\sqrt{2}}$. More details can be found in original paper of SR3 [3]. In order to condition on the LR image during training, SR3 uses bicubic interpolation to up-sample the LR image to the same resolution as HR image. The up-sampled LR image, x, is simply concatenated with y_t along the channel dimension as the input to f_θ. The whole inference process is illustrated in Algorithm2.

Fig. 3: The leftmost column shows the HR images and the second column shows SR image using the pretrained model without fine-tuning. Other columns show SR images with different fine-tuning steps. SR images above are fine-tuned with L1 loss, and below are fine-tuned with L2 loss.

4 Experimental Results

4.1 Training Details and Noise Schedule

Training Details: For all experiments, we used Adam optimizer with fixed learning rate of 1e-4 and batch size of 4. We also set the dropout rate to 0.2 following SR3. The total time step was set to 2000, and we performed $16 \times 16 \rightarrow 128 \times 128$ for all super resolution experiments. Images with different original resolution were resized with bicubic interpolation. All experiments were carried out on a single GTX 1060 GPU.
Noise Schedule: We adopted a linear noise schedule for α:

$$\alpha_t = 10^{-6} + \frac{10^{-2} - 10^{-6}}{T}(t-1).$$

4.2 Datasets and Pre-trained Model

In this work, we used three datasets: FFHQ [11], CelebA-HQ [12] and an animation faces dataset AnimeF.

FFHQ and CelebA-HQ: Both FFHQ and CelebA-HQ are real human faces dataset, which contians 70,000 and 30,000 images respectively. Different from CelebA-HQ, FFHQ covers a wider variation in terms of age, ethnicity, image background, and accessories such as eyeglasses, sunglasses, hats, etc.

AnimeF: This is an animation character faces dataset released by Prof. Huang-yi Lee in his Machine Learning courses at National Taiwan University. It contains 71,314 images from the internet.

In this wrok, due to computational limits, we only used 128 images from AmimeF for training and 8 images for

Fig. 4: The leftmost column shows the HR images and the second column shows SR image using the pretrained model without fine-tuning. Other columns show SR results fine-tuned on different amount of data. All SR images are fine-tuned 2000 steps with L1 loss.

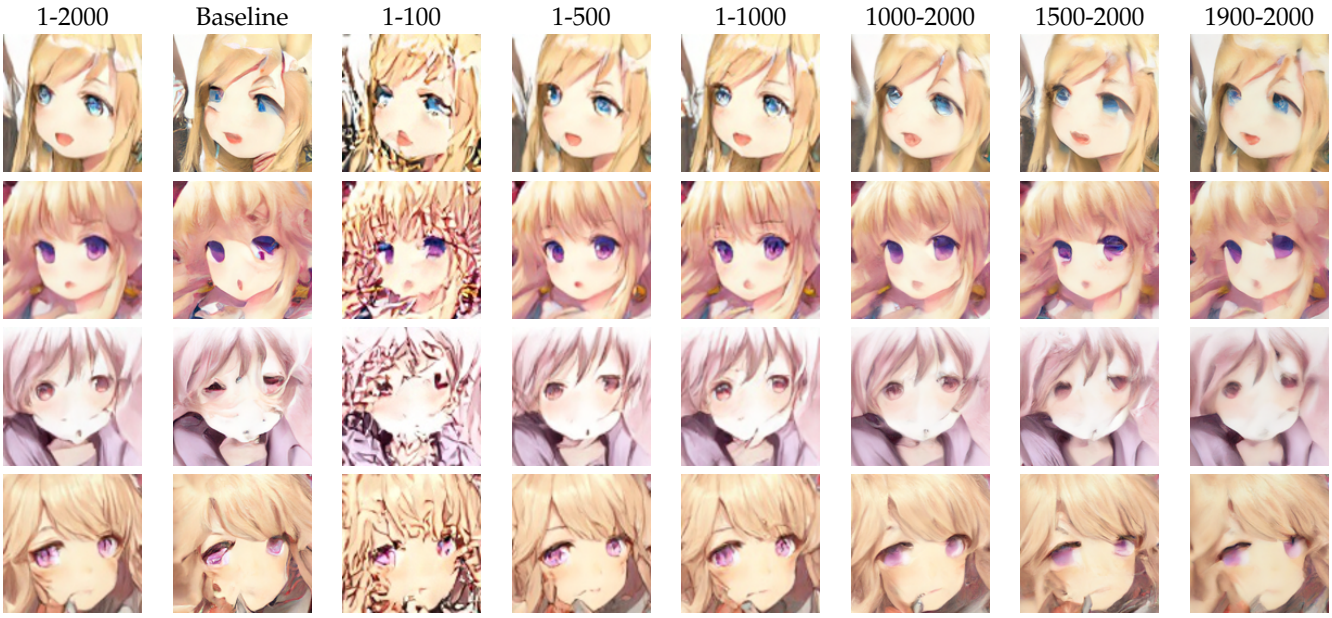

Fig. 5: The leftmost column shows the SR images fine-tuned with all possible time steps, and the second column shows SR image using the pretrained model without fine-tuning. Other columns show SR results fine-tuned on different range of time steps. All SR images are fine-tuned 2000 steps with L1 loss.

testing. For evaluations on FFHQ and CelebA-HQ, we only used 8 images from each dataset. Note that in SR3, it uses FFHQ as training set and CelebA-HQ as testing set.

Pre-trained Model: Since there isn't an official implementation of SR3, this work is based on the reproduction by Janspiry. As mentioned by Janspiry, there might be slight difference between the official implementation and reproduction, but since SR3 was a relatively simple method, we assume this discrepancy will not cause major difference in our experiments. The pre-trained model has been trained on FFHQ for 640k steps. All experiments were based on the same pre-trained model released at https://github.com/Janspiry/Image-Super-Resolution-via-Iterative-Refinement.

4.3 Fine-tuning with Limited Steps

4.3.1 Qualitative Evaluation

Figure.3 shows the SR results of fine-tuning with different steps with L1 loss and L2 loss respectively. Images above

TABLE 1: Quantitative evaluations on different fine-tuning steps

	PSNR↑	SSIM↑	LPIPS↓
FFHQ	22.56	0.673	**0.2331**
CelebA-HQ	**23.27**	**0.676**	0.2355
Baseline	20.01	0.495	0.3770
L1 loss			
400	20.81	0.499	0.3314
1200	20.98	0.515	0.3192
1600	20.71	0.508	0.3111
2000	21.16	0.524	0.3136
4000	20.69	0.504	0.3088
12000	**21.57**	**0.540**	**0.3062**
L2 loss			
400	20.39	0.479	0.3499
2000	20.32	0.488	0.3236
6000	20.91	0.513	0.3091
8000	21.12	0.521	0.2985
12000	21.66	0.539	**0.2968**
18000	**21.75**	**0.542**	0.3002

are results of L1 loss, and we can see we obtains much better SR images after fine-tuning than the baseline, which corresponds to directly use the pre-trained model. And it is obvious that with the steps increased from 400 to 4,000, we get clearer results with more high frequency details in the image, e.g. hairs. Note that the step here is different from epoch, a step is a single update with gradient calculated from a single batch, while an epoch updates the model with the entire training set once. However, more fine-tuning steps does not necessarily mean better visual performance. We can see SR results fine-tuned with 12,000 steps are blurrier than SR results with 4,000 steps fine-tuning. Images below are SR results fine-tuned with L2 loss, and we have similar findings with the results fine-tuned with L1 loss. We think fine-tuned with 8,000 steps with L2 loss achieves the best visual performance, with fewer steps, there will be still some distortions in the image, with more steps, the image become less sharp.

Comparing SR results between L1 loss and L2 loss, we think their best visual performance are about the same, however, it takes more steps for L2 loss. Because of this, we will only report evaluations with L1 loss in the later sections.

4.3.2 *Quantitative Evaluation*

Table.1 shows the quantitative evaluations on different fine-tuning steps with L1 loss and L2 loss respectively. We use 3 three different metrics for evaluation. PSNR and SSIM [13] are the most widely used measurements for the quality of image restoration. In comparison, SSIM gives better indications in terms of image quality. However, these two metrics may not correlate well with human perception, especially when the input resolution is low and the magnification factor is large [3]. LPIPS [14] is a recently proposed metric, which measures the difference between the ground truth image and the target image in the latent space with pre-trained model. This metric is better correlated with human perception than PSNR and SSIM. We can see after fine-tuning, we have better results in terms of all three metrics. For results of L1 loss fine-tuning, PSNR and SSIM do not correlate well with our qualitative evaluation, this may because there exist squared error and covariance terms in PSNR and SSIM respectively, and L1 loss does not necessarily minimize thess terms. This is verified by the results of L2 loss fine-tuning which explicitly minimizes mean square error, and with more steps, we have better PSNR and SSIM results. LPIPS results matches well with our qualitative evaluation, while there still have some mismatches. We think there are two reasons for this issue: one is that qualitative evaluation is quite subjective and our perception may not represent the truth; another reason is that LPIPS's model is pre-trained with real-world images, which means it may not be able to transform animation characters into latent space well, resulting in inaccurate measurement. We believe the latter one may be the case, because even though we think the SR images are quite like the HR images, there is a big gap between the LPIPS's results of FFHQ, CelebA-HQ and AnimeF. However, there are also gaps between the PSNR's and SSIM's results, and all these gaps may be explained as the domain gaps between the datasets themselves. We can't determine the true reason for this and leave this for future work.

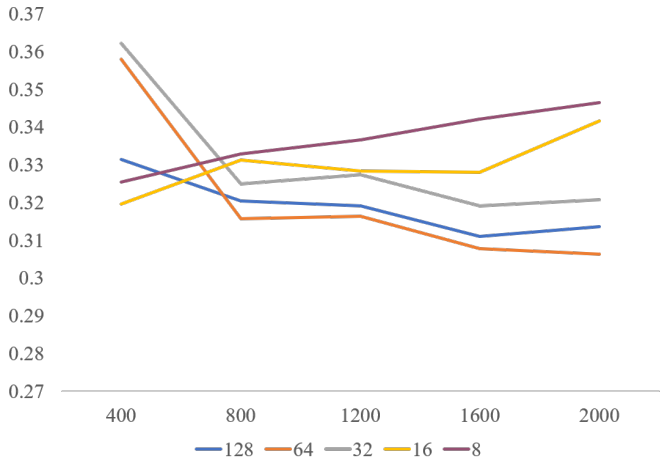

Fig. 6: LPIPS results for different steps and different amount of data for fine-tuning

TABLE 2: Quantitative evaluations on different amount of data for fine-tuning

	PSNR↑	SSIM↑	LPIPS↓
FFHQ	22.56	0.673	**0.2331**
CelebA-HQ	**23.27**	**0.676**	0.2355
Baseline	20.01	0.495	0.3770
128	21.16	0.524	0.3136
64	21.35	0.527	**0.3063**
32	21.56	0.536	0.3208
16	21.99	0.552	0.3417
8	**22.07**	**0.555**	0.3465

TABLE 3: Quantitative evaluations on different range of time steps for fine-tuning

	PSNR↑	SSIM↑	LPIPS↓
FFHQ	22.56	0.673	**0.2331**
CelebA-HQ	**23.27**	**0.676**	0.2355
Baseline	20.01	0.495	0.3770
1-2000	21.16	0.524	0.3136
1-100	15.62	0.302	0.4692
1-500	20.71	0.514	**0.3082**
1-1000	20.91	0.513	0.3096
1000-2000	**21.72**	**0.541**	0.3349
1500-2000	21.29	0.519	0.3678
1900-2000	20.93	0.527	0.3890

4.4 Fine-tuning with Limited New Domain Data

4.4.1 Qualitative Evaluation

Figure.4 shows the SR results fine-tuned with different amount of new domain data. For efficiency, we fine-tune the pre-trained model with 2,000 steps, which yield good visual performance in the previous section, using L1 loss for all different data sizes. Generally, we have better results with more new domain data, the SR images become blur when there are only few data in the training set. However, we obtain better results fine-tuned with 64 new images than 128 new images. This is because we fix the steps for fine-tuning and the corresponding epochs will change. For example, when we fine-tune 2,000 steps with 128 training images, we actually fine-tune the model for 62.5 epochs (batch size is 4). If we halve the training set, we double the epochs. And the epochs of training set with 64 images after 2,000 steps is the same as the epochs of training set with 128 images after 4,000 steps. Recall that we obtain best visual performance with 4,000 steps fine-tuning in the previous section, and this explains why we have better result fine-tuning on 64 new images. This can also explain why we get blur SR images with very few training data. In the previous section, we find that after more steps, the image become blurrier, and here we fine-tune our model with more epochs when there are fewer training data, which causes the SR images to be blur.

4.4.2 Quantitative Evaluation

Table.2 shows the quantitative evaluation with different training data. This time LPIPS perfectly matches our qualitative evaluation, but PSNR and SSIM don't. As argued in [3], PSNR and SSIM are extremely conservative with high frequency details, which prefer blurrier images. And this is not the only reason, as shown in Figure.6, even with fewer steps we still cannot obtain good results with insufficient amount of data, the lack of new data itself also causes the image to be blur.

4.5 Fine-tuning with Different Time Steps

4.5.1 Qualitative Evaluation

Diffusion model generates image based on the reverse Markov Chain, which starts with pure Gaussian noise. If we assume the reason that the pre-trained model cannot generalize well on the new dataset is that there exists domain gap between the datasets. We can expect that the domain gap will gradually diminish as t become larger, and eventually become zero. For image super resolution tasks, we need to condition on the LR image for generation, such that the domain gap will not become zero, but it's still reasonable to expect smaller gaps when t is large.

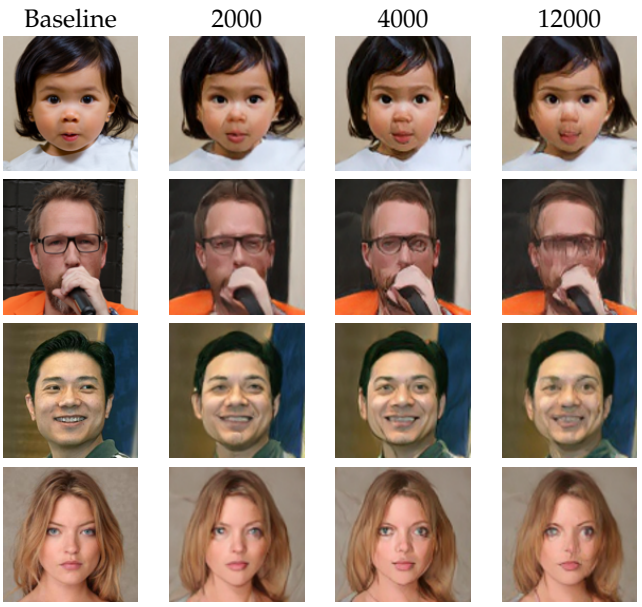

Fig. 7: SR results on FFHQ and CelebA-HQ after different steps of fine-tuning.

Following this idea, Figure.5 shows the SR results fine-tuned with different range of time steps. We can see we achieve comparable or even better result only fine-tuning the first 500 or 1,000 time steps, which verifies our assumption above. Similar with the results in Section 4.4, we actually performs more fine-tuning if the range of time step is small, and this explains why we have even better results than fine-tuning with all the time steps. However, if the range of time step is too small, e.g. 0-100, the domain gap is still large and we have the worse result than the baseline. As our assumption suggested, only fine-tuning the later part of the Markov Chain will give us worse results than only fine-tuning the former part.

4.5.2 Quantitative Evaluation

Table.3 shows the quantitative evaluation with different range of time steps. Again, LPIPS matches well with our qualitative evaluation, but PSNR and SSIM don't.

4.6 Does Fine-tuning Impair Original Performance?

We have demonstrated that we can achieve good SR results on new domain data with limited fine-tuning in the previous sections. Now, the question is whether this will impair the model's original performance. Unfortunately, the answer is yes, as shown in Figure.7. We can see there are more distortions in the SR images when we fine-tune more steps. The quantitative results in Table.4 also verifies this fact. The only good news is that there are only few distortions in SR images on FFHQ and CelebA-HQ, when we obtain the best performance on AnimeF, namely fine-tuning with 4,000 steps. This indicates that it may not impair

TABLE 4: Quantitative evaluations on FFHQ and CelebA-HQ after fine-tuning

	PSNR↑	SSIM↑	LPIPS↓
FFHQ			
Baseline	22.56	**0.673**	**0.2331**
2000	22.89	0.666	0.2525
4000	22.69	0.661	0.2592
12000	**23.07**	0.666	0.2659
CelebA-HQ			
Baseline	23.27	0.676	0.2355
2000	23.76	**0.701**	**0.2320**
4000	23.39	0.689	0.2427
12000	**23.96**	**0.701**	0.2498

the model's original performance much with few steps of fine-tuning, and we may obtain satisfying results with new domain data.

5 Conclusion

In this work, we explore the generalization ability of SR3 in image super resolution task by fine-tuning the pre-trained model with limited steps, new domain training data and range of time steps. We demonstrate that fine-tuning with less than 1% of pre-training steps, we can obtain fairly good results on new domain data. The amount of the new domain data needed for fine-tuning is a little bit tricky, if you have too much data, it may take more steps for fine-tuning, and if you only have very insufficient data, you will not have satisfying SR results. We have also shown that it's not necessary to fine-tune all the time-steps, fine-tuning only on the first half will give you almost the same or even better result.

However, there are still problems we haven't resolve yet, and can be considered as future work. First, we only explore the generalization ability of SR3 which works on pixel space, the diffusion model works on the latent space may not share the same result, e.g. LDMs. Secondly, there exists a gap between quantitative results on FFHQ, CelebA-HQ and AnimeF. Lastly, naive fine-tuning will impair the model's original performance. Hope these problems will be resolved in the future.

Acknowledgments

We would like to thank Janspiry for reproducing and open-sourcing SR3. This work is mainly built on their code and pre-trained model.

References

[1] H. Chen, X. He, L. Qing, Y. Wu, C. Ren, R. E. Sheriff, and C. Zhu, "Real-world single image super-resolution: A brief review," *Information Fusion*, vol. 79, pp. 124–145, 2022.

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

[3] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[5] Y. Liu, H. Zhao, J. Gu, Y. Qiao, and C. Dong, "Evaluating the generalization ability of super-resolution networks," *arXiv preprint arXiv:2205.07019*, 2022.

[6] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 2437–2445.

[7] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, "Camera lens super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1652–1660.

[8] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3118–3126.

[9] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[11] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[12] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

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