DiffGAR: Model-Agnostic Restoration from Generative Artifacts Using Image-to-Image Diffusion Models

Yueqin Yin  
School of Artificial Intelligence, University of Chinese Academy of Sciences  
Institute of Automation, Chinese Academy of Sciences  
Beijing, China

Lianghua Huang  
Machine Intelligence Technology Lab, Alibaba Group  
Beijing, China

Yu Liu  
Machine Intelligence Technology Lab, Alibaba Group  
Beijing, China

Kaiqi Huang  
School of Artificial Intelligence, University of Chinese Academy of Sciences  
Institute of Automation, Chinese Academy of Sciences  
CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China

ABSTRACT
Recent generative models show impressive results in photo-realistic image generation. However, artifacts often inevitably appear in the generated results, leading to downgraded user experience and reduced performance in downstream tasks. This work aims to develop a plugin post-processing module for diverse generative models, which can faithfully restore images from diverse generative artifacts. This is challenging because: (1) Unlike traditional degradation patterns, generative artifacts are non-linear and the transformation function is highly complex. (2) There are no readily available artifact-image pairs. (3) Different from model-specific anti-artifact methods, a model-agnostic framework views the generator as a black-box machine and has no access to the architecture details. In this work, we first design a group of mechanisms to simulate generative artifacts of popular generators (i.e., GANs, autoregressive models, and diffusion models), given real images. Second, we implement the model-agnostic anti-artifact framework as an image-to-image diffusion model, due to its advantage in generation quality and capacity. Finally, we design a conditioning scheme for the diffusion model to enable both blind and non-blind image restoration. A guidance parameter is also introduced to allow for a trade-off between restoration accuracy and image quality. Extensive experiments show that our method significantly outperforms previous approaches on the proposed datasets and real-world artifact images.

CCS CONCEPTS
• Computing methodologies → Computer vision tasks.

KEYWORDS
datasets, generative modeling, image generation, image restoration

This work is licensed under a Creative Commons Attribution International 4.0 License.

CSAI 2022, December 9–11, 2022, Beijing, China  
© 2022 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9777-3/22/12.  
https://doi.org/10.1145/3577530.3577539

1 INTRODUCTION
Generative models such as generative adversarial networks [17], autoregressive models [24], diffusion models [11], variational autoencoders [34], and normalization flows [20], show impressive capability in photo-realistic image synthesis and editing in recent years. However, due to the imperfect learning of the mapping between distributions or the discontinuity in the latent space, artifacts usually inevitably appear in some of the generation results, as shown in the first column of each subgraph in Fig. 3. These artifacts may reduce the generation performance and the performance of downstream tasks.

The generative artifacts can be viewed as a special form of image degradation with respect to the implicit original image. However, different from traditional image degradation modes [18], [36], [37] such as JPEG artifacts, Gaussian blur, Gaussian noise and downsampling artifacts, which can usually be represented as a low-order transformation function (e.g., $y = [(x \otimes k) \downarrow_s + n]_{\text{JPEG}}$), generative artifacts are non-linear, model-specific, and the transformation function is highly complex, which cast challenges in the modeling of the restoration process. As far as we know, only a few works have been proposed to deal with the artifacts of generative models. Two works [14], [33] have been conducted to detect artifacts produced by GAN models. However, these models are model-specific since they are limited to the latent space of GAN. EdiBERT [13] learns a masked token prediction model on discredited images, and it assumes that the artifact regions correspond to lower-probability tokens. It achieves image restoration by replicating those tokens with low probabilities. Although these algorithms can restore images from a specific type of generative artifacts, they are typically model-specific and cannot generalize to different types of generative models.
In this paper, we would like to develop a model-agnostic general framework for the faithful restoration of clean images from various types of generative artifacts. It is worth noting that the causes of various types of artifacts generated by different generative models vary by model structure, and it is difficult to propose a general method to directly avoid the artifacts in the generation process. Therefore, the focus of this work is not to analyze the reasons why the model produces artifacts during generation, but to propose a model-agnostic artifact removal method to improve the quality of image generation. The term model-agnostic denotes that we have no access to the detailed implementation of the generator, but rather consider it as a black-box module. To achieve this, we consider generative artifact removal as an image-to-image translation task, where the only input is the degraded image with generative artifacts, while the output is the restored image.

One challenge is that no groundtruth is available for each degraded image. To generate training data for our restoration model, we design a set of mechanisms to degrade real images that simulate different types of generative artifacts. Specifically, by exploring the components or parameters that may lead to generative artifacts, we synthesize four generative artifacts from clean images, corresponding to three popular generative models, GAN, autoregressive model and diffusion model. Notably, it is inevitable that there is a certain gap between the artifact we build and the real generative artifact produced by generative models. However, we will demonstrate in the experimental part that the model trained on the synthetic artifact dataset also shows great restoration ability to the real-world generative artifacts.

Next, we propose a baseline method based on a conditional diffusion model, called DiffGAR, for this generative artifact removal task. Given an artifact image and a source image, our algorithm can generate a clean image removing artifacts. Traditional degradation restoration tasks aim to estimate the degradation kernel and then produce a clean image through the inversion process [30]. Inspired by the recently proposed conditional image-to-image diffusion model [29], we treat this artifact removal task as a task to learn the conditional distribution of original clean images given input artifacts, using diffusion models can capture the multi-modal distributions in the high-dimensional image spaces. We then train our algorithm on our newly proposed artifact-image pairs dataset. Besides, among the traditional degradation restoration tasks, many works focus on solving blind image restoration tasks [7]. Blind artifacts removal aims to restore the original images suffering from unknown degradation kernels or noise levels. Similarly, we design a conditioning scheme for the diffusion model to enable blind artifact removal with an extra blind class token embedding.

The overall contributions of this work can be summarized as follows:

- We design methods for synthesizing four types of generative artifacts that are as close as possible to real artifacts for restoration model training.
- We propose a baseline algorithm, DiffGAR, based on the conditional diffusion model, which achieves high quality restoration from artifact images and shows better reconstruction accuracy and image fidelity than previous works.
- We conduct thorough experiments on different settings of the modeling of DiffGAR, including both blind and non-blind restoration mechanisms. We also support a flexible trade-off between image quality and consistency of the results.
- The proposed DiffGAR model trained on the synthetic artifact datasets can restore real image artifacts and achieve higher human visual preference than previous works, making it more practical for real-world applications.

2 RELATED WORK

2.1 Image Generation

Deep generative models have made great progress in image synthesis tasks recently [6], [9], [25]. GAN-based methods have shown amazing capabilities in generating high-fidelity samples, but they have poor mode coverage. Autoregressive models cover data manifold faithfully, but they often suffer from low sample quality. Concurrently, the diffusion models have emerged as a promising generative modeling framework, advancing the development of image, audio, and video generation tasks. However, artifacts are often generated due to architectural limitations of the generative model itself and its inability to capture the complete data manifold pattern [38], [14], [32], highlighting the importance of generative artifacts removal for producing more visually appealing images.

2.2 Traditional Degradation Restoration

Many traditional tasks in image restoration can be transformed into linear inverse problems, such as super-resolution, colorization, deblurring, and compressive sensing. GFFGAN [36] incorporate GAN prior to traditional blind facial image restoration. [37] trains a practical Real-ESRGAN for real-world blind super-resolution with pure synthetic training pairs. Besides, diffusion models have recently been introduced into the image restoration tasks. Denoising Diffusion Restoration Models (DDRM) [18] has been proposed as a general linear inverse problem solver based on unconditional diffusion generative models.

2.3 Generative Artifacts Restoration

Recently, there are a few works focusing on exploring the unique artifacts in GAN model architectures. The author [14] removes artifacts through ablating units that are related to artifact generations. The paper [32] propose a novel pixel-instance normalization (PIN) layer to remove the circular artifacts for vanilla StyleGAN. However, these methods are specifically designed for partial GAN artifacts. EdiBERT [13] is a kind of generative model which models the VQGAN latent space in a non-autoregressive manner. EdiBERT replaces the autoregressive GPT model in Taming Transformers with a bi-directional BERT [5] model. When applied to image denoising tasks, EdiBERT will first detect discrete tokens in a given artifact image that do not fit appropriately in the sequence s, and then change these tokens to new tokens so that the likelihood of the new sequence can be higher.

3 SYNTHESIS OF GENERATIVE ARTIFACTS

In this section, we describe our method of synthesizing different kinds of generative artifacts corresponding to three typical types
of generative models, GAN, Autoregressive model and DDIM. Furthermore, we synthesize training pairs using images from FFHQ dataset [16] and AFHQ-Dog dataset [4].

### 3.1 GAN Artifacts

Since it is difficult to directly synthesize high-quality, id-preserving artifacts-image pairs based on StyleGAN2 [17], we use VQGAN [8] instead, which is also a GAN model. Taming Transformers [8] is a two-stage generative model that first learns a latent representation of the data and then, in a second stage, an autoregressive gpt probabilistic model of the latent representation. VQGAN is a vector-quantized image model which can map an image into a sequence of discrete latent variables, serving as the first stage model of Taming Transformers. The vector-quantized image model consists of an encoder E, a decoder D and a codebook Z = {z_k}^K_{k=1} ∈ R^{K×d}, where K is the number of discrete codes in the codebook Z and d is the dimension of codes. The convolutional encoder E downsamples an image x ∈ R^{c×H×W} into a feature map z_e ∈ R^{d×h×w}. Then each spatial feature vector z_ij is substituted via a nearest-neighbour lookup onto a discrete codebook entry zk:

\[ z_q = \text{argmin}_{z_k} \| z_{ij} - z_k \|_2^2 \in \mathbb{R}^{d×h×w} \]  

(1)

where h × w represents the sequence length of the quantized image. Subsequently, the quantized encoding z_q is fed to the decoder D to reconstruct the input \( \hat{x} = D(z_q) \).

For VQGAN artifacts, given an image x, we can obtain the image tokens sequence \( s = (s_1, s_2, \cdots, s_j) \) for a given image. In the second stage of Taming Transformers, for a given discrete token sequence \( s = (s_1, s_2, \cdots, s_j) \), an autoregressive transformer is trained to predict the next sequence token conditioned on the previously predicted image tokens:

\[ p_{\theta}(s) = \prod_{i=1}^{l} p_{\theta}^l(s_i|s_{<i}) \]  

(2)

During inference, the most commonly used sampling strategies of autoregressive models are temperature sampling and top-k sampling. We can implement temperature sampling by dividing the logits outputs of the autoregressive transformer model by the temperature, which is then fed into a softmax layer and outputs the sampling probabilities of tokens in the vocabulary. Higher temperatures make the model increasingly confident in its top choices. The top-k sampling means ordering by probability and zeroing out the probabilities below the kth token. We simulate autoregressive model sampling artifacts via a higher temperature and top-k value.

In the remaining part of the paper, we use GPT sampling artifacts to represent the autoregressive model sampling artifacts for simplicity.

### 3.2 Autoregressive Model Sampling Artifacts

The second stage model of Taming Transformers [8] is a transformer-based autoregressive generative model.

With encoder E and decoder D available, we can obtain discrete tokens \( s = (s_1, s_2, \cdots, s_j) \) for a given image. In the second stage of Taming Transformers, for a given discrete token sequence \( s = (s_1, s_2, \cdots, s_j) \), an autoregressive transformer is trained to predict the next sequence token conditioned on the previously predicted image tokens:

\[ p_{\theta}(s) = \prod_{i=1}^{l} p_{\theta}^l(s_i|s_{<i}) \]  

(2)

During inference, the most commonly used sampling strategies of autoregressive models are temperature sampling and top-k sampling. We can implement temperature sampling by dividing the logits outputs of the autoregressive transformer model by the temperature, which is then fed into a softmax layer and outputs the sampling probabilities of tokens in the vocabulary. Higher temperatures make the model increasingly confident in its top choices. The top-k sampling means ordering by probability and zeroing out the probabilities below the kth token. We simulate autoregressive model sampling artifacts via a higher temperature and top-k value.
we perturb the obtained latent code with additive white Gaussian noise, \( \tilde{x}_t = x + \alpha \cdot n \), \( n \sim N(0, I) \), where \( \alpha \) is a hyperparameter controlling the strength of the added noise. Secondly, we perform scaling and shifting operation using the latent code \( x_{\text{aff}} = y \cdot x + \beta \), where \( y \) and \( \beta \) correspond to the scale and shift coefficients. These two operations sometimes result in streaky artifacts in the image.

4  CONDITIONAL DIFFUSION MODELS

We utilize the diffusion model as the generative artifact removal backbone due to its powerful generative capability. Following Latent Diffusion [27], to reduce the computational cost and fitting difficulty of training a diffusion model, we use a two-stage diffusion model as our method. In the first stage, an autoencoder model compresses the input image into a low-dimensional latent space. We then construct a powerful conditional diffusion model over the compressed latent space to learn the joint distribution. Fig. 1 shows the structure of our model DiffGAR.

Stage One: AutoEncoder Model. Based on [8] which learns a discrete latent space, we learn a continuous compressed latent space which allows training a continuous diffusion model. Given an image \( x \in \mathbb{R}^{3 \times H \times W} \) in RGB space, the encoder \( E \) first encodes the image into a latent representation \( z \in \mathbb{R}^{d \times H \times W} \), then \( z \) can be fed into the decoder \( D \) to reconstruct the input image.

Stage Two: Conditional Diffusion Model. Given the generative artifacts dataset of artifact-image pairs, denoted \( D = \{ X_i, Y_i \}_{i=1}^{N} \), the goal of our method is to learn a parametric approximation to \( p(Y|X) \) that maps a source artifact image \( X \) to the clean target image \( Y \). With the trained first-stage autoencoder, we can obtain the latent representation \( x, y_0 \) of a given artifact image and a clean target image. Compared to the original high-dimensional image space, the compressed low-dimensional pixel space is more efficient to train the conditional diffusion model. Taking the source artifact image \( x \) and the noisy target image \( y_t \) as input, we can train a neural backbone \( E_D \) to progressively denoise \( y_t \) and produce the target image \( y_0 \). During training, we use a shared U-Net [28] architecture for all timesteps \( t \), by injecting the current timestep \( t \) with sinusoidal position embedding [35]. Additionally, since the artifact images in the proposed dataset have different artifact class labels, we can utilize the class labels as additional information to guide the diffusion process. Specifically, we learn a class embedding matrix for the proposed four types of generative artifacts during training and add the class embedding to the timestep embedding for each sample during training. However, similar to the blind image restoration tasks where the degradation model is not provided, when applied to artifact images generated directly from the generative models, we cannot know exactly which of the four artifacts in the dataset the current artifact image belongs to. Therefore, we learn an extra [MASK] class token \( c_{\text{ls}} \) which represents the unknown artifact class then it can be applied to blind image restoration. During training, we randomly set 50% of the input artifact class tokens to the [MASK] class token. The detailed training algorithm is shown in Algorithm 1.

Blind / Non-Blind Generative Artifacts Restoration. In the inference stage, we accelerate the generative process by using fewer discretization steps [31]. Besides, we design three kinds of generative artifacts restoration methods, non-blind restoration, blind restoration and a tradeoff between them. For non-blind image restoration, we set the artifact class token to the true artifact class. For blind image restoration, the [MASK] class token is fed into the noise diffusion model. Besides, inspired by the classifier-free guidance sampling in previous conditional image generation tasks [12], [23] which
We evaluate the effectiveness of our DiffGAR model on the proposed synthetic generative artifact images as well as the real artifact images sampled from the generative models. We compare DiffGAR with two traditional restoration models GFGAN [36] (for FFHQ artifact dataset), Real-ESRGAN [37] (for AFHQ-Dog artifact dataset) and a generative artifacts removal model EdiBERT [13].

Datasets Construction. We generate artifact images on the FFHQ and AFHQ-Dog dataset at 256×256 resolution. For the FFHQ dataset, we use 68k images for training and 2k images for inference. For the AFHQ-Dog dataset, 4735 images are used for training, and 500 images are used for inference. For GPT sampling artifacts, we use the well-trained model from [8] and set different temperatures and top-k values to simulate GPT sampling artifacts. For the FFHQ dataset, we set the number of top-k to 500 and the sampling temperature to 21. For the AFHG-Dog dataset, we set the number of top-k to 600 and the sampling temperature to 22. Notably, in order to keep the identity of the object in the image, we set 90% of image tokens unchanged and only 10% of the tokens will be resampled using the given temperature and top-k. For replace_token artifacts, we randomly sample a 4×4 rectangle and replace the tokens within the selected rectangle with randomly sampled tokens from the codebook. When generating ddim_gaussian artifacts, we set the stopping step of the forwarding process T0 to 840, forward diffusion steps and inversion steps to 48, α to 0.3. Ddim_scale artifacts share the same diffusion steps as ddim_gaussian, and γ, β is set to be 1.015, 0.01 for the FFHQ dataset. For AFHQ-Dog, we set γ to 1.015, β to 0.001.

Evaluation Metrics. We use FID [10] as the generation quality comparison metric. The image reconstruction quality is evaluated by mean squared error (MSE) in image pixel level space. Besides, we adopt pixel-wise metrics PSNR and SSIM. To measure the identity preserving ability, we use iResNet [1] and CLIP [26] to extract image features for images in FFHQ and AFHQ-Dog datasets, respectively, and then calculate the similarity between features.

Training Details. We use the pretrained model of an autoencoder network from Latent-diffusion [27], which maps the input image into a latent representation $z \in \mathbb{R}^{512 \times 64 \times 64}$. We train our DiffGAR model for 390k and 87k training steps on the FFHQ dataset and AFHQ-Dog Dataset dataset with a batch size of 24, respectively. We adopt a U-Net architecture for the diffusion model and consider $T = 1000$ for the diffusion process. During inference, we use 30 DDIM sampling timesteps.

5 EXPERIMENTS

We evaluate the effectiveness of our DiffGAR model on the proposed synthetic generative artifact images as well as the real artifact images sampled from the generative models. We compare DiffGAR with two traditional restoration models GFGAN [36] (for FFHQ artifact dataset), Real-ESRGAN [37] (for AFHG-Dog artifact dataset) and a generative artifacts removal model EdiBERT [13].

Datasets Construction. We generate artifact images on the FFHQ and AFHQ-Dog dataset at 256×256 resolution. For the FFHQ dataset, we use 68k images for training and 2k images for inference. For the AFHQ-Dog dataset, 4735 images are used for training, and 500 images are used for inference. For GPT sampling artifacts, we use the well-trained model from [8] and set different temperatures and top-k values to simulate GPT sampling artifacts. For the FFHQ dataset, we set the number of top-k to 500 and the sampling temperature to 21. For the AFHG-Dog dataset, we set the number of top-k to 600 and the sampling temperature to 22. Notably, in order to keep the identity of the object in the image, we set 90% of image tokens unchanged and only 10% of the tokens will be resampled using the given temperature and top-k. For replace_token artifacts, we randomly sample a 4×4 rectangle and replace the tokens within the selected rectangle with randomly sampled tokens from the codebook. When generating ddim_gaussian artifacts, we set the stopping step of the forwarding process $T_0$ to 840, forward diffusion steps and inversion steps to 48, $\alpha$ to 0.3. Ddim_scale artifacts share the same diffusion steps as ddim_gaussian, and $\gamma, \beta$ is set to be 1.015, 0.01 for the FFHQ dataset. For AFHG-Dog, we set $\gamma$ to 1.015, $\beta$ to 0.001.

Evaluation Metrics. We use FID [10] as the generation quality comparison metric. The image reconstruction quality is evaluated by mean squared error (MSE) in image pixel level space. Besides, we adopt pixel-wise metrics PSNR and SSIM. To measure the identity preserving ability, we use iResNet [1] and CLIP [26] to extract image features for images in FFHQ and AFHQ-Dog datasets, respectively, and then calculate the similarity between features.

Training Details. We use the pretrained model of an autoencoder network from Latent-diffusion [27], which maps the input image into a latent representation $z \in \mathbb{R}^{512 \times 64 \times 64}$. We train our DiffGAR model for 390k and 87k training steps on the FFHQ dataset and AFHQ-Dog Dataset dataset with a batch size of 24, respectively. We adopt a U-Net architecture for the diffusion model and consider $T = 1000$ for the diffusion process. During inference, we use 30 DDIM sampling timesteps.

Algorithm 1: Training a denoising model $\theta$. 

```
1: repeat 
2: \( (X, Y, \text{cls}) \leftarrow \text{sample training artifact-image pair and artifacts class token} \)
3: \( x \leftarrow E(X), y_t \leftarrow E(Y) \)
4: \( t \leftarrow \text{Uniform}([1, \ldots, T]) \)
5: \( \epsilon \sim N(0, I) \)
6: \( L \leftarrow \|\nabla\theta(x, \sqrt{\alpha_t}y_t + \sqrt{1-\alpha_t}\epsilon, t|\text{cls}) - \epsilon\|^2 \)
7: \( \theta \leftarrow \theta - \eta \Delta \theta L \)
8: until \( \text{converged} \)
```

Algorithm 2: Inference of the denoising model, given fast inference. 

```
1: \( T, \theta \leftarrow E(X) \), \text{cls}_t \) is the artifacts class token of \( X \)
2: \( y_t \leftarrow N(0, 1) \)
3: while \( t > 0 \) do 
4: \( \epsilon \leftarrow \hat{\epsilon}_\theta(x, y_t, t|\text{cls}_t) \)
5: \( y_0 = \frac{y_t - \sqrt{1-\alpha_t}\epsilon}{\sqrt{\alpha_t}} \)
6: \( y_t = \sqrt{1-\alpha_t}y_0 + \sqrt{1-\alpha_t}\epsilon \)
7: \( t \leftarrow t - \Delta t \)
8: end while
9: return \( D(y_t) \)
```
Table 1: Quantitative restoration results on synthetic datasets between GFPGAN ([36]), Real-ESRGAN ([37]), EdiBERT ([13]) and Ours. The comparison results show that our method produces higher quality output images.

| Metric   | Method            | Replace Token | GPT Sampling | DDIM Gaussian | DDIM Scale | ALL       |
|----------|-------------------|---------------|--------------|---------------|------------|-----------|
|          |                   | FID ↓         | FID ↑         | MSE ↓         | PSNR ↑     | SSIM ↑    | ID Consistency ↑ |
|          | GFPGAN            | 19.32         | 0.0384       | 0.0701        | 20.585     | 0.6101    | 0.6169    | 0.5852  |
|          | Real-ESRGAN       | 27.43         | 0.0412       | 0.0741        | 20.17      | 0.5189    | 0.5027    | 0.9019  |
|          | EdiBERT           | 15.55         | 0.0336       | 0.0734        | 18.053     | 0.5415    | 0.5628    | 0.9592  |
|          | DiffGAR(Ours)     |               |              |               |            | 0.6182    | 0.5571    | 0.9596  |
|          |                   |               |              |               |            |           |           |        |

Table 2: Comparison with previous work on real-world generative artifacts (human preference).

| Model     | Preference ↓ |
|-----------|--------------|
| GFPGAN    | 27%          |
| EdiBERT   | 18%          |
| DiffGAR(Ours) | 55%        |

Figure 2: Restoration results comparison on proposed FFHQ artifacts dataset (a) and AFHQ-Dog artifacts dataset (b). Each row shows the different types of generative artifacts.

Figure 3: Restoration results of real artifacts generated by GAN [17], GPT [8], DDPM model [3], respectively.

Qualitative Evaluation. Qualitative results are presented in Fig. 2 and Fig. 3. Compared to other methods, DiffGAR produces high quality restoration results, as shown in Fig. 2. The two traditional image restoration models GFPGAN [36] and Real-ESRGAN [37] do not have the ability to restore image regions with semantic errors. While EdiBERT can remove discrete artifacts such as Replace Token artifacts and GPT Sampling artifacts as shown in the first two rows of Fig. 2, the identity of the person changes slightly, while our method DiffGAR can preserve the identity better and successfully remove the artifacts. For the remaining DDIM-related artifacts involving the entire image, only DiffGAR can successfully remove the artifacts. Fig. 3 shows the restoration results of real artifacts generated by GAN [17], GPT [8], DDPM model [3], respectively.
DiffGAR: Model-Agnostic Restoration from Generative Artifacts Using Image-to-Image Diffusion Models

Figure 3: Comparisons of GFPGAN [36], EdiBERT [13], and our DiffGAR restoration results on real image artifacts generated by different generative models (StyleGAN2 [17], Autoregressive Model [8], DDPM [3]). DiffGAR (the last column) trained with synthetic artifacts dataset can successfully remove the real-world artifacts produced by various generative models.

Figure 4: Ablation study on the classifier-free guidance scale and the number DDIM sampling steps on the FFHQ dataset.

DiffGAR exhibits better image restoration capabilities compared to other models in all three kinds of generative artifacts.

5.2 Ablation Study
We conduct ablation studies on the FFHQ artifact dataset.

Blind / Non Blind Image Restoration. We investigate how different artifacts restoration methods affect the restoration performance. As shown in Tab. 3, blind image restoration performs the worst in terms of reconstruction quality. The non-blind image restoration strategy recovers most of the details of the target image with the
smallest MSE value. We find that the classifier-free guidance clearly improves the image quality in terms of FID, at the cost of reduced reconstruction detail. Fig. 4a shows how the guidance scale s affects the restoration results. We find that when s increases, the FID score and ID consistency become better, while the MSE score decreases. We hypothesize that this may be because the classifier-free guidance pushes the samples toward the true data distribution while ignoring instance details. In this paper, we choose a guide scale of 3.0 to achieve a trade-off between image quality and reconstruction accuracy. However, it is off to the users to choose a proper setting according to their requirements.

DDIM sampling steps. We study the impact of the number of DDIM sampling steps on restoration performance. As shown in Fig. 4b, as the number of sampling steps increases, the FID score and ID consistency get better while the MSE score decreases. When the number of sampling steps increases to 30, the ID consistency score starts to drop. We assume that the more sampling steps, the better the image quality, but at the cost of changing more details of the image to get closer to the real manifold space. We use 30 DDIM sampling steps in our experiment because it provides a good trade-off between various evaluation metrics.

6 CONCLUSION

This paper introduces the generative artifacts restoration task along with two new generative artifact datasets. Furthermore, we design a general generative artifacts restoration framework DiffGAR based on a conditional diffusion model and demonstrate the great potential of restoring clean images from artifact images. We expect this work could attract more researchers from the community to further study the artifacts removal of generated images, including a better method of simulating generative artifacts and the development of new methods to improve the image generation quality.

REFERENCES

[1] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Invertible residual networks. ICML.
[2] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2021. WaveGrad: Estimating gradients for waveform generation. ICLR (2021).
[3] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sangroh Yoon. 2022. Perception Prioritized Training of Diffusion Models. CVPR.
[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. Stargan v2: Diverse image synthesis for multiple domains. CVPR.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
[6] Prafulla Dhariwal and Alex Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. NeurIPS (2021).
[7] Majed El Helou and Sabine Sulsstrunk. 2020. Blind universal Bayesian image denoising with Gaussian noise level learning. TIP 29 (2020).
[8] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. CVPR.
[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. NeurIPS 27 (2014).
[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two-time-scale update rule converge to a local nash equilibrium. NeurIPS (2017).
[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. NeurIPS 33 (2020).
[12] Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications.
[13] Thibault Izenourth, Ugo Tanetian, Jérémie Mary, and David Picard. 2021. EdiBERT, a generative model for image editing. ArXiv abs/2111.15264 (2021).
[14] Haedong Jeong, Jyeon Han, and Jaesik Choi. 2022. An Unsupervised Way to Understand Artifact Generating Internal Units in Generative Neural Networks. AAAI.
[15] Zahra Kadkhodaei and Eero P Simoncelli. 2020. Solving linear inverse problems using the prior implicit in a denoiser. arXiv preprint arXiv:2007.13640 (2020).
[16] Terō Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. CVPR.
[17] Terō Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. CVPR.
[18] Bahjat Kawar, Michael El, Stefano Ermon, and Juaming Song. 2022. Denoising Diffusion Restoration Models. arXiv preprint arXiv:2201.11793 (2022).
[19] Zhangheng Lin, Wei Ping, Jiayi Zhang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761 (2020).
[20] Chenlin Meng, Yang Song, Juangong Song, Juqian Wu, Jun-Yan Zhu, and Stefano Ermon. 2019. Sdeedit: Image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2010.01753 (2021).
[21] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021).
[22] Georg Ostrovski, Will Dabney, and Rémi Munos. 2018. Autoregressive quantile networks for generative modeling. ICLR.
[23] Nikif Parmar, Ashish Vaswani, Jakub Uskokovic, Lukasz Kaiser, Noam M. Shazeer, Alexander Ku, and Dustin Tran. 2018. Image Transformer. ICLR.
[24] Alex Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. ICLR.
[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjørn Ommer. 2022. High-resolution image synthesis with latent diffusion models. CVPR.
[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. 234–241.
[27] Chitwan Saharia, William Chan, Huwenn Chang, Chris A Lee, Jonathan Ho, Tim Salimans, David J Fleet, and Mohammad Norouzi. 2021. Palette: Image-to-image diffusion models. arXiv preprint arXiv:2111.05826 (2021).
[28] Assaf Shocher, Nadav Cohen, and Michal Irani. 2018. “zero-shot” super-resolution using deep internal learning. CVPR.
[29] Juaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. ICLR (2021).
[30] Way Tan, Bihan Wen, and Xulei Yang. 2021. Systematic analysis and removal of circular artifacts for stylegan. arXiv preprint arXiv:2103.01990 (2021).
[31] Ali Tousi, Haedong Jeong, Jyeon Han, Hwanil Choi, and Jaesik Choi. 2021. Automatic correction of internal units in generative neural networks. CVPR.
[32] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. NeurIPS (2017).
[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakub Uskokovic, Llion Jones, Aidan G N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. NeurIPS (2017).
[34] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. 2021. Towards real-world blind face restoration with generative facial prior. CVPR.
[35] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-resgan: Training real-world blind super-resolution with pure synthetic data. ICCV.
[36] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. 2019. Detecting and simulating artifacts in gan fake images. NeurIPS (2019).

Table 3: Ablation Study blind/non-blind image restoration.

|        | FID | MSE | ID consistency |
|--------|-----|-----|----------------|
| blind  | 17.45 | 0.0347 | 0.6295 |
| non-blind | 17.79 | 0.0258 | 0.6843 |
| guidance | 16.89 | 0.0316 | 0.6886 |