

Received 25 December 2023, accepted 1 January 2024, date of publication 3 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349423



RESEARCH ARTICLE

An Improved Face Image Restoration Method Based on Denoising Diffusion Probabilistic Models

YUN PANG^{ID}¹, JIAWEI MAO¹, LIBO HE², HONG LIN¹, AND ZHENPING QIANG^{ID}¹

¹College of Big Data and Intelligence Engineering, Southwest Forestry University, Kunming 650224, China

²Information Security College, Yunnan Police College, Kunming 650221, China

Corresponding author: Zhenping Qiang (qzp@swfu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 12163004; in part by the Yunnan Fundamental Research Program of Agricultural Special Projects, China, under Grant 202301BD070001-008 and Grant 202101BD070001-05; and in part by the Fundamental Research Projects of Yunnan Provincial Department of Education, China, under Grant 2022J0496.

ABSTRACT Image restoration is a crucial task in computer vision, aiming to fill in missing areas within an image to restore its integrity. Traditional methods fall short when dealing with intricate facial image restoration, often failing to produce high-quality results. Denoising Diffusion Probabilistic Models(DDPM), characterized by its diversity and stability, plays a significant role in the domain of facial image restoration. This study aims to explore a facial image restoration method based on DDPM, utilizing a pre-trained unconditional DDPM model to achieve more flexible facial image restoration. At the same time, this study found that when the total number of iterations in the resampling process is relatively low, the quality of the restored image is poor. Therefore, we propose a method to optimize the inversion process by combining progressive sampling with sample scheduling to improve the quality of the restored images, and conduct extensive experiments on the CelebA-HQ and FFHQ datasets. Comparisons with other methods demonstrate that our approach yields higher-quality results in facial image restoration. Our method achieved the best results in terms of PSNR and LPIPS metrics. For random masks, the accuracy of face recognition increased by 15.7% after the restoration of facial images. For central masks, the accuracy improved by 26%.

INDEX TERMS Face image restoration, DDPM, progressive sampling, sampling scheduling.

I. INTRODUCTION

Image restoration, also known as image inpainting, aims to fill in missing regions within an image. Early tasks in facial image restoration relied predominantly on mathematical methods and patch-based approaches. For mathematical methods, the majority are based on partial differential equation (PDE) [1], [2] and interpolation-based approaches for image restoration [3].

For facial image restoration, mathematical methods often struggle to achieve high-quality results. Patch-based approaches [4], [5], fundamentally entail choosing suitable blocks (patches) from different areas of the image and copying or transforming them to fill in the missing regions.

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar .

These methods can achieve partial restoration of facial images in simple cases. For extensive missing or damaged areas, we typically employ deep learning-based facial image restoration techniques. Leveraging deep neural networks allows automatic learning of facial image features, leading to higher-quality restoration results.

Generative Adversarial Networks (GANs) [6], as a powerful model in the field of deep learning, exert significant influence through their unique adversarial training approach. Not only do they play a crucial role in various domains such as image generation [7], [8], [9], [10], [11], video synthesis [12], [13], and natural language generation [14], [15], but they also drive advancements in the field of image restoration.

The generative capabilities of GANs extend beyond generating realistic images and exhibit excellent performance in face image restoration tasks. GANs can produce high-quality

restored images while effectively preserving the structure and semantic information of the images. This characteristic is particularly crucial for face images with high-dimensional attributes and complex textures.

Despite the enormous potential of GANs in image restoration tasks, their training process poses challenges in terms of stability. Common issues such as mode collapse [16] and mode oscillation can lead to a lack of diversity in generated samples. Face images generated by GANs may sometimes exhibit instability, artifacts, or distortion issues. Additionally, when dealing with large missing regions, generators in GANs often struggle to acquire sufficient global information, making it difficult to accurately restore the content of the missing areas. Overcoming these challenges requires further research and improvement in the field [17].

Compared to GANs, DDPM [18] has certain advantages in several aspects. Firstly, DDPM is capable of generating diverse and higher-quality samples, indicating that the generated images are closer to real data and less prone to issues like artifacts or distortion. Secondly, the training process of DDPM is generally more stable, and it is less susceptible to problems such as mode collapse or mode oscillation. This stability makes DDPM a reliable choice for handling complex image data [19], [20], [21], [22], [23], [24], [25].

Additionally, DDPM adopts a progressive generation approach, allowing it to better maintain the global consistency and structure of images. Existing studies have confirmed that DDPM exhibits stronger capabilities compared to GANs [26], [27]. Therefore, applying DDPM to face image restoration tasks holds immense potential and promising prospects [28], [29], [30], [31].

As a generative model, DDPM's reverse sampling strategy often generates images with matching textures, but these textures may not align with the semantic information of the image. Therefore, directly applying DDPM to facial image restoration tasks is not feasible. Maintaining image consistency after DDPM-based restoration becomes a challenge in facial image restoration tasks.

RePaint [28] proposed a method called Resample to enhance the reverse sampling strategy of DDPM, aiming to generate more semantically aligned image restoration results. Although this new sampling method significantly improves the quality of restored images, it introduces a potential issue by increasing the number of iterations. The increased number of iterations results in higher computational resource requirements, thereby escalating the computational cost.

It is worth noting that our study also identified a limitation in RePaint's resampling strategy when there are fewer sampling steps and smaller jump lengths, corresponding to a lower number of iterations. In such cases, the quality of the restored images is compromised.

We propose an improved face image restoration method based on DDPM, aiming to utilize a pre-trained unconditional DDPM model for facial image restoration. In contrast to traditional training with specific masks, we opt not to train

DDPM with specific masks. The advantage of this decision lies in the adaptability of our network during the reverse sampling process to various mask inputs, enhancing its flexibility. In response to the issue of poorer image quality in the restoration process with fewer iterations in RePaint, we propose a method that combines progressive sampling with sampling scheduling to optimize the reverse sampling strategy based on Resample. The aim is to enhance the quality of restored images with a reduced number of iterations. Through this improvement, we aim to offer a more effective solution for facial image restoration tasks.

We conducted experiments on the CelebA-HQ [32] and FFHQ [9] datasets, comparing our method with other approaches. Through qualitative and quantitative analyses, we compared the face restoration images using SSIM [33], PSNR [34], and LPIPS [35] metrics. The research results indicate that our proposed method exhibits superior generalization capabilities and achieves significant improvements in both PSNR and LPIPS metrics. The generated images possess richer semantic information. Additionally, we conducted an application analysis, and the results demonstrate a significant improvement in face recognition accuracy after face image restoration.

In the second section, we will conduct an in-depth review of pertinent literature, examining prior works relevant to our study to establish a comprehensive background and theoretical framework. The third section will meticulously detail our research methodologies, encompassing progressive sampling and sampling scheduling strategies. The fourth section will undertake a rigorous analysis of experimental results, drawing comparisons with alternative methods and incorporating insightful ablation experiments. The fifth section will delve into a comprehensive discussion of the experimental outcomes. Finally, the sixth section will encapsulate our conclusions, summarizing the pivotal findings of our study and accentuating its contributions.

II. RELATED WORK

A. DDPM

This paper employs DDPM [18] as the generative method. DDPM, a probabilistic distribution-based generative model, refines the process of sample generation by progressively handling noise, aiming to closely match the probability distribution of the training data. The DDPM model comprises two key processes: the forward process (also known as the diffusion process) and the reverse process.

The forward process (1) of DDPM involves gradually introducing Gaussian noise $\varepsilon \sim N(0, I)$ to the initial data $x_0 \sim q(x_0)$, transforming the original data progressively into random noise. This process establishes a Markov chain (2). The diffusion process of DDPM possesses a crucial characteristic: through the use of reparameterization techniques, it allows for the direct generation of data x_t at any time step t based on the initial data x_0 (3). Through this reparameterization technique, x_t can be regarded as the linear

combination of the initial data x_0 and random noise ϵ .

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \quad (1)$$

The sample x_t is obtained by adding *i.i.d.* Gaussian noise with variance β_t at timestep t and scaling the previous sample x_{t-1} with $\sqrt{1 - \beta_t}$ according to a variance schedule.

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \quad (2)$$

$$X_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \quad (3)$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Here, $1 - \bar{\alpha}_t$ tells us the variance of the noise for an arbitrary timestep, and we could equivalently use this to define the noise schedule instead of β_t .

The primary objective of the forward process is to gradually introduce noise to the original data, while the reverse process aims to achieve the opposite goal: step-by-step denoising to generate a real sample. In the reverse process (4), we start with an initial random noise sample $x_t \sim N(0, \mathbf{I})$. By progressively reducing the level of noise, we gradually restore a sample that closely resembles real data. Similar to the forward process, the reverse process is defined as a Markov chain (5) to ensure the continuity and consistency of the denoising process.

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (4)$$

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \quad (5)$$

If we consider the intermediate generated variables as latent variables, then DDPM can be categorized as a type of latent variable model. This categorization allows us to leverage variational inference to maximize the optimization objective (6). By introducing the framework of this latent variable model, we can more effectively learn and infer model parameters, thereby enhancing the performance and data generation capability of the model.

$$\begin{aligned} L = -L_{VLB} &= \mathbb{E}_{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right] \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \end{aligned} \quad (6)$$

Through a thorough analysis and decomposition by Ho et al [18], we further derive (7).

$$\begin{aligned} &\mathbb{E}_q \underbrace{D_{KL}(q(\mathbf{x}_T \mid \mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} \\ &+ \sum_{t>1} \underbrace{D_{KL}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{L_{t-1}} \\ &- \underbrace{\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0} \end{aligned} \quad (7)$$

Finally, DDPM simplified the aforementioned objectives, removing weighting coefficients in the derivation process,

resulting in a more straightforward training objective (8).

$$L_{\text{simple}} = E_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right] \quad (8)$$

This objective can be seen as a reweighted form of L_{VLB} (without the terms affecting Σ_θ). The authors found that optimizing this reweighted objective resulted in much better sample quality than optimizing L_{VLB} directly. They explain this improvement by drawing a connection to generative score matching.

B. REPAINT

RePaint [28] applies DDPM to image restoration tasks, with its core idea being to utilize the model's predicted results at each step of the sampling process to handle regions within the mask, while areas outside the mask use a noisy version of the real image. However, this straightforward approach has a significant drawback: the information within the mask is nearly entirely lost, making it challenging for the content inside and outside the mask to semantically match.

While the model attempts to coordinate the results from the previous steps at each iteration, this often introduces new disharmonies, making the problem challenging to resolve entirely. Additionally, as the reverse process progresses, the variance gradually decreases, making it more difficult to maintain image consistency. This is particularly crucial in tasks like facial image restoration, where more steps are needed to preserve image consistency.

To address the issue of semantic inconsistency mentioned above, as shown in Fig 1, RePaint introduces the Resample technique to optimize the DDPM reverse process. The key to this technique is to rediffuse the generated image x_{t-1} back to x_t (1), achieving a “repeated hopping” effect on the Markov chain. This process aims to assist the model in mitigating inconsistencies in the restored images. Through the Resample technique, RePaint can better address semantic inconsistencies, thereby enhancing the effectiveness of face image restoration. This innovative approach opens up new possibilities for face image restoration tasks.

The Resample technique proposed by the RePaint network plays a crucial role in enhancing the sampling strategy of the DDPM reverse process, successfully overcoming issues related to poor semantic information. Furthermore, subsequent research [36] provides a detailed theoretical explanation of the Resample technique in the RePaint algorithm, demonstrating its correctness and effectiveness from a rigorous theoretical perspective.

The Resample technique introduces two key hyperparameters: i and j , representing the number of sampling steps and the jump lengths, respectively. The selection of these parameters influences the total number of iterations, depending on different the number of sampling steps and jump lengths. Table 1 illustrates the total number of iterations corresponding to various combinations of sampling steps and jump lengths.

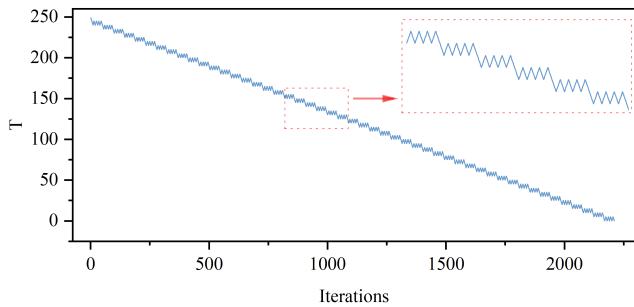


FIGURE 1. Iterations during inference. The diffusion time T that a sample x_t undergoes in the inference process with a jump length $j = 5$ and resampling $i = 5$.

However, it is important to note that through experimentation, we have observed that with small values of i and j , the image quality of RePaint-restored images tends to be lower. Additionally, the Resample technique significantly increases the total number of iterations, leading to increased computation time and costs. Therefore, when employing the Resample technique, a balance between image quality and computational cost needs to be struck, and appropriate hyperparameter values should be chosen to meet the specific requirements of the task.

TABLE 1. Total number of iterations corresponding to different resampling counts and jump lengths.

T	i	j	Iterations
250	2	2	746
250	3	3	1246
250	5	5	2210
250	10	10	4570

C. IMAGE RESTORATION WITH DDPM

Many researchers believe that diffusion models have the potential to represent the next generation of image generation models. In the closely related field of image inpainting, some studies [19], [20], [21], [22], [23], [24], [25], [26], [28], [31], [37], [38], [39], [40], [41] have begun to explore the use of diffusion models for image inpainting.

Kawar et al. [38] employed unsupervised posterior estimation, demonstrating the potential of diffusion models for image inpainting. Theis et al. [39] utilized an unconditional generation method, encoding images to be restored with diffusion models and showcasing their potential in lossy image compression. Dhariwal and Nichol [26] verified that the performance of diffusion models surpassed that of GANs and further refined image inpainting based on diffusion models. Lugmayr et al. [28] sampled from undamaged areas of an image to replace the reverse diffusion process in the diffusion model. This model can handle irregular and free-form damage, representing a relatively successful modification of the diffusion model used in image inpainting tasks, outperforming both GANs and VAEs.

III. METHOD

In this chapter, we first introduce, in Section III-A, a method using progressive sampling in the reverse process of unconditional DDPM to generate the regions to be restored. Subsequently, in Section III-B, we present an improved Resample method, employing a sampling scheduling strategy to enhance the quality of facial image restoration. Our overall approach is illustrated in Figure 2.

At the beginning, our inputs include damaged images and a mask. The known regions, after undergoing the forward noisy process of DDPM (1), are element-wise multiplied by the mask (top). The initial input x_t for the unknown regions is random noise. Before $I \times 0.9$ (I : Iterations), the standard DDPM reverse sampling strategy is applied (4), followed by element-wise multiplication with (1-mask). When λ is less than 1, a progressive sampling strategy (Section III-A) is employed to correct the noise, then entering the next iterations. When λ equals 1, the known regions are directly added to the unknown regions for the next iterations (11). After $I \times 0.9$, the reverse process adopts a sampling schedule strategy (Section III-B) for optimization (13) (bottom).

A. PROGRESSIVE SAMPLING

In this paper, we first train an unconditional DDPM. In this model, we define the original image as x_0 , the known region as $mask \odot x_0$, and the unknown region as $(1 - mask) \odot x_0$. Since the forward process of DDPM forms a Markov chain (2), we can sample an image x_t at any time t based on x_0 (3). This allows us to sample the known region $mask \odot x_0$ at any time t .

In the reverse process of DDPM, the generated image depends solely on x_t (4). Therefore, in the reverse process, we can also sample the unknown region $(1 - mask) \odot x_t$. Thus, we can express this as follows:

$$x_{t-1}^{\text{known}} \sim \mathcal{N} \left(\sqrt{\bar{\alpha}_t} x_0, \sqrt{1 - \bar{\alpha}_t} \mathbf{I} \right) \quad (9)$$

$$x_{t-1}^{\text{unknown}} \sim \mathcal{N} \left(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t) \right) \quad (10)$$

$$x_{t-1} = mask \odot x_{t-1}^{\text{known}} + (1 - mask) \odot x_{t-1}^{\text{unknown}} \quad (11)$$

This expression delineates the approach we employ in the DDPM model to generate unknown regions utilizing known regions and a reverse process. Through this method, we enhance our ability to address image restoration problems, achieving restoration of the unknown regions more effectively.

As the noise in DDPM's reverse sampling is randomly generated, its probability distribution also follows a random pattern. Correcting this initial noise from the outset contributes to a more effective achievement of restoration results. Experimental evidence demonstrates that, at the initiation of reverse sampling, gradually reducing the weight of the unknown region x_{t-1}^{unknown} , as opposed to the standard DDPM sampling steps, results in better overall coherence in our outcomes.

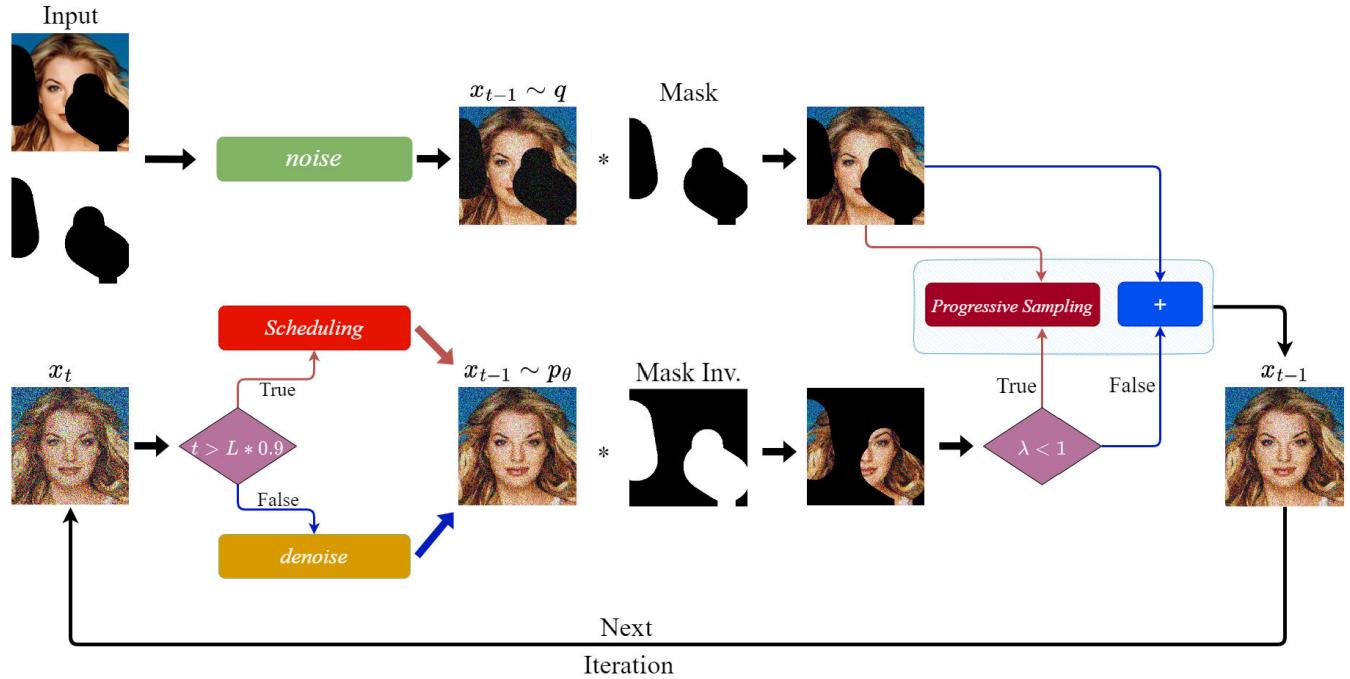


FIGURE 2. Overview of our approach. In each step, we sample the known region (top) from the input and the inpainted part from the DDPM output (bottom).

In specific terms, the initial x_t in the reverse process of DDPM is a random Gaussian noise with a stochastic probability distribution, which is inconsistent with the content of the region we intend to restore. Therefore, at the initiation of the reverse process in DDPM, we initially introduce a randomly generated Gaussian noise of the same size as the image to be restored. Subsequently, we gradually decrease the weight of the unknown region x_{t-1}^{unknown} . The purpose of this approach is to continuously refine the initial x_t at the early stages of the reverse process, thereby enhancing the overall quality of the restored image.

We employed a simple mathematical formula to gradually reduce the weight of the unknown region (12). In this formula, λ is a variable with an initial value of 0.0, increasing by $1/\delta$ at each iteration but not exceeding 1.0. This ensures that the value of λ remains within the $[0, 1]$ range. Thus, we obtain the complete expression for x_{t-1} :

$$\lambda = \min\left(\lambda + \frac{1}{\delta}, 1.0\right) \quad (12)$$

$$x_{t-1} = \begin{cases} \text{mask} \odot x_{t-1}^{\text{known}} + (1 - \text{mask}) \odot (1 - \lambda) \odot x_{t-1}^{\text{unknown}}, \\ \text{if } \lambda < 1 \\ \text{mask} \odot x_{t-1}^{\text{known}} + (1 - \text{mask}) \odot x_{t-1}^{\text{unknown}}, \\ \text{if } \lambda = 1 \end{cases} \quad (13)$$

Through this progressive sampling strategy, our aim is to continuously refine the probability distribution of the initial x_t , thereby enhancing the overall quality of the restored

image. The effects of the progressive sampling are detailed in Section IV-D1.

B. SAMPLING SCHEDULING

Sampling scheduling [42] is a widely used training strategy in machine learning and deep learning. Its purpose is to optimize the training process of a model to enhance performance, generalization ability, and robustness. We expect the model to have the capability to learn and self-adjust its sampling strategy. In the reverse process of DDPM, the model continuously seeks the probability distribution, i.e., calculating the predicted mean and variance. In this scenario, we can schedule the handling of the model output.

The sampling scheduling we employ involves gradually reducing the model output based on the current sampling iteration count. The idea behind this setup is that DDPM may require a larger exploration space in the early stages of sampling, so we provide a wider sampling range. As the iterations progress, DDPM may have learned more useful information. Therefore, we limit the model output by reducing the sampling range, allowing it to focus more on the restoration of critical regions.

Through this sampling scheduling strategy, we can enhance the model's learning capability and predictive performance. It not only allows the model to focus more on crucial regions but also helps prevent the model from getting stuck in local optima prematurely, leading to better restoration results.

The key aspects of the sampling scheduling strategy include three components: the choice of the sampling stage,

the selection of the lower bound for sampling scheduling, and the choice of the decay rate for sampling scheduling. For the first question (Q1), we conducted a series of experiments exploring the effects of different sampling stages under the same lower bound and decay rate conditions and conducted a detailed analysis of the results.

From the results in Figure 3, it can be observed that the restoration performance is not ideal when sampling scheduling is performed in the first 80% of the total iterations. This is because the early selection of sampling scheduling significantly reduces the model output in the early sampling iterations, making denoising ineffective. Based on empirical evidence, it has been shown that the sampling scheduling strategy is most effective when applied after 90% of the total iterations.

For the second question (Q2), we conducted a series of experiments based on 0.9 Iterations, exploring the impact of different lower bound selections under the same decay rate conditions, and analyzed the results. From the results in Figure 4, it can be seen that when the lower bound is below 0.85, the noise is not completely removed. This is because the model output is too small, making the denoising process ineffective. Through comparative experiments, we ultimately chose to set the scheduling lower bound to 0.9 to achieve better restoration results.

For the third question (Q3), we analyzed that the decay rate of sampling scheduling should start relatively small and gradually increase to ensure the quality of the restoration results. To address this, we employed a specific decay function described by (14) to depict the variation of the decay rate. The form of this decay function is shown in Figure 5.

$$f(t) = \text{min_sch} + (\text{initial_sch} - \text{min_sch}) \cdot e^{-\gamma \cdot t} \quad (14)$$

In this function, we set initial_schedule to 1, the min_schedule to 0.9, and experimented with three different decay rates (γ): 0.005, 0.01, and 0.02. The specific decay functions are illustrated in Figure 5. We conducted experiments with each decay rate, and detailed results can be found in Section IV-D2.

IV. EXPERIMENT

We conducted extensive experiments on the CelebA-HQ [32] and FFHQ [9] test sets. The CelebA-HQ dataset is an extension of the CelebA [43] dataset, containing high-quality facial images with higher resolution. Similarly, the FFHQ dataset is a collection of high-quality facial images sourced from users on Flickr, known for its high image quality. Both datasets are suitable for deep learning tasks related to facial analysis. Image preprocessing includes cropping, face alignment, denoising, and other techniques. In Section IV-C, we analyze the restoration results of our network and compare them with other methods. In Section IV-D1, we provide a detailed analysis of our progressive sampling, and in Section IV-D2, we conduct a thorough analysis of the sampling schedule decay rate.

A. IMPLEMENTATION DETAILS

We validated our method using the RTX 4090 on the CelebA-HQ and FFHQ datasets. We set $T = 250$ timesteps. We conducted experiments for four types of resampling cases in RePaint: $i = 2, j = 2$, iterations = 746, $i = 3, j = 3$, iterations = 1246, $i = 5, j = 5$, iterations = 2210, and $i = 10, j = 10$, iterations = 4570. Two types of masks, random masks, and central masks, were selected. For random masks, we used different proportions (30%-40%, 40%-50%, 50%-60%, 60%-70%, 70%-80%) and irregular-shaped masks. We used 256x256 crops in three batches on the RTX 4090 GPU each. The CelebA-HQ model is only trained for 250,000 iterations during roughly nine days. All our qualitative and quantitative results are based on images with a resolution of 256×256 . The experimental process in this paper is implemented under the PyTorch framework. For the experiment, 80% of the images from the CelebA-HQ dataset are selected as the training set, and the remaining 20% are designated as the test set. We randomly selected 50 images from the CelebA-HQ 20% test set and another 50 images from the FFHQ dataset, making a total of 100 test images for the experiment.

B. EVALUATION METRICS

We utilized 100 images from the CelebA-HQ and FFHQ test sets, each with a size of 256×256 . We calculated the SSIM [33], PSNR [34], and LPIPS [35] metrics.

SSIM measures the similarity between images, with higher values indicating greater similarity. PSNR evaluates the level of distortion in images, where higher values signify less distortion. LPIPS integrates deep learning and human perceptual aspects, considering factors like image structure, color, and texture. Smaller LPIPS values are preferable.

This comprehensive evaluation with various metrics provides insights into the performance of different methods in facial image restoration tasks, guiding further optimization efforts.

C. COMPREHENSIVE EXPERIMENTS AND COMPARATIVE ANALYSIS

When adjusting the parameter γ (Section IV-D2), we observed generally poor image restoration results when $\gamma = 0.02$. Therefore, we further conducted a detailed analysis with random masks and central masks at $\gamma = 0.005$ and $\gamma = 0.01$, respectively, considering different values of δ .

For $\gamma = 0.005$, we performed a quantitative analysis, and the results are shown in Table 3. From the table, it can be observed that for scenarios like 746RAND ($i=2, j=2$, iteration=746, random mask; the same applies below) and 746CENT ($i=2, j=2$, iteration=746, central mask; the same applies below), the restoration quality is optimal at $\delta = 20$, while it degrades at $\delta = 40$ and 50. For 1246RAND and 1246CENT, good restoration quality is achieved at $\delta = 20$ and 30. In the case of 2210RAND and 2210CENT, good restoration quality is observed at $\delta = 35$ and 40. For

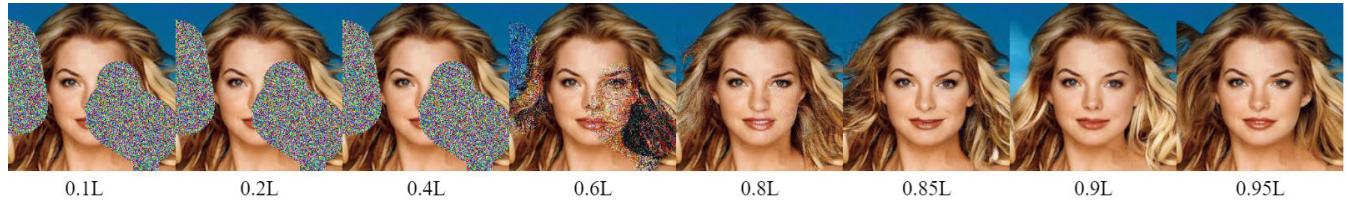


FIGURE 3. Comparison of restoration effects at different sampling stages.

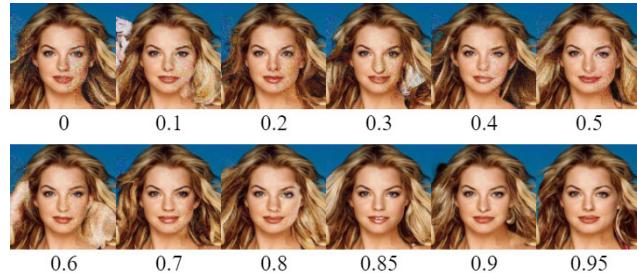


FIGURE 4. Impact of different lower bound selections on restoration effects.

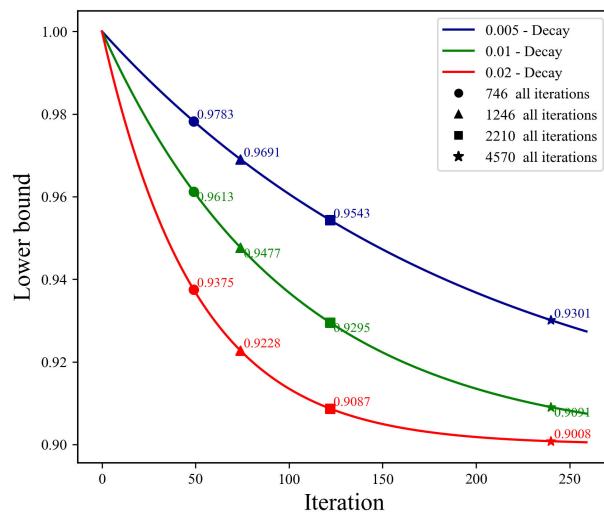


FIGURE 5. Impact of different lower bound selections on restoration effects.

4570RAND and 4570CENT, optimal restoration quality is achieved at $\delta = 70$ and 80.

At $\gamma = 0.001$, we observed a restoration quality trend similar to that at $\gamma = 0.005$. The specific quantitative analysis results are presented in Table 4. Additionally, we compared the best results of our method with RePaint under different tasks, and the relevant results are shown in Figure 7. For detailed conclusions and analysis, please refer to Section IV-D1 and Section IV-D2.

We compared our proposed improved face image restoration method based on DDPM with other types of face restoration methods, including GAN-based EC [44], CTSDG [45], DeepFill v2 [46], Yohan et al. [47], Transformer-based ICT [48], and DDPM-based RePaint [28]. Qualitative

analysis is shown in Figure 6. We observed that, compared to other methods, our restoration results are more natural and closer to the ground truth.

For example, in the first image restoration, we found that the restoration quality of EC and CTSDG is poor. In contrast, our method is closer to the ground truth, and it outperforms DeepFill v2, ICT, Yohan et al., and RePaint. In the third image restoration, EC, CTSDG, Yohan et al., and DeepFill v2 exhibit poor restoration quality, while ICT has defects in the left hair restoration. Our restoration results, especially in the right eyeglass hair region, are more natural compared to RePaint. In the fifth image restoration, EC, CTSDG, and DeepFill v2 still show poor restoration quality, and RePaint performs poorly in the restoration of the right eye. Our restoration results, on the other hand, are relatively more natural compared to ICT. In the seventh image restoration, EC and CTSDG show poor restoration quality, DeepFill v2 produces inconsistent eyebrow restoration, ICT's restoration results are semantically inconsistent, while our restoration results are more natural compared to Yohan et al. and Repaint.

The quantitative analysis results are presented in Table 2. From the table, it can be observed that our method has slightly lower SSIM compared to Yohan et al. but achieves the best results in terms of PSNR and LPIPS indicators.

TABLE 2. Quantitative Analysis Comparison. We compared our proposed method with EC [44], CTSDG [45], DeepFill v2 [46], ICT [48], RePaint [28] and Yohan et al. [47] using SSIM, PSNR, and LPIPS metrics.

RAND	SSIM	PSNR	LPIPS
EC	0.8589	31.9037	0.1037
CTSDG	0.8522	32.1764	0.1270
DeepFillv2	0.8530	32.7273	0.1380
ICT	0.8695	33.0768	0.0974
RePaint	0.8636	33.1785	0.0973
Yohan et al.	0.8715	33.2090	0.0962
Ours	0.8667	33.2155	0.0960
CENT	SSIM	PSNR	LPIPS
EC	0.9139	34.9211	0.0653
CTSDG	0.9072	34.9165	0.1010
DeepFillv2	0.9252	34.8973	0.0574
ICT	0.9269	34.9087	0.0503
RePaint	0.9256	34.9590	0.0515
Yohan et al.	0.9285	34.9695	0.0502
Ours	0.9263	34.9713	0.0501

D. ABLATION EXPERIMENT

1) PROGRESSIVE SAMPLING

In this section, we will conduct a detailed study on the impact of progressive sampling on restoration results. We will



FIGURE 6. Qualitative Results: Comparison with other types of facial image restoration methods. In the figures, (a) represents EC [44], (b) represents CTSRG [45], (c) represents DeepFill v2 [46], (d) represents ICT [48], (e) represents RePaint [28], and (f) represents Yohan et al. [47].

analyze the effects of progressive sampling under different restoration tasks and conditions using visual qualitative results in Figure 8 and quantitative data in Table 5.

From the qualitative results, it becomes evident that not all parameter selections contribute to enhancing the final restoration quality. For example, for 746RAND, the overall restoration results of RePaint are relatively poor. Our method outperforms RePaint for δ values of 10, 20, and 30, while our network's restoration results are worse for δ values of 40 and 50.

In the case of 746CENT, the visual consistency of our restoration results is superior to RePaint for δ values of 10, 20, and 30. However, for $\delta = 50$, our network's restoration results do not align with the semantic content.

For 1246CENT, the eyes restored by RePaint are clearly inconsistent with semantic information. Our network performs well in restoration results for δ values of 20, 30, and

40, but for $\delta = 10$, the restored right cheek does not align with semantic information.

Concerning 2210RAND, RePaint does not align with semantic information when restoring hair on the forehead. Our network achieves good restoration results for δ values of 20, 35, 40, and 50, but the restoration of hair becomes less effective for $\delta = 70$.

In the case of 2210CENT, the pupils restored by RePaint are inconsistent. Our network produces good restoration results for δ values of 35, 40, 50, and 70, but for $\delta = 20$, the right bangs are somewhat blurry.

For 4570RAND, RePaint introduces a discontinuity in the left hair and yellow spots on the right hair. Our network performs well in restoration results for δ values of 60, 70, and 80, but for $\delta = 40$, there is a discontinuity in the left hair. For $\delta = 100$, the left hair clearly does not align with semantic information.

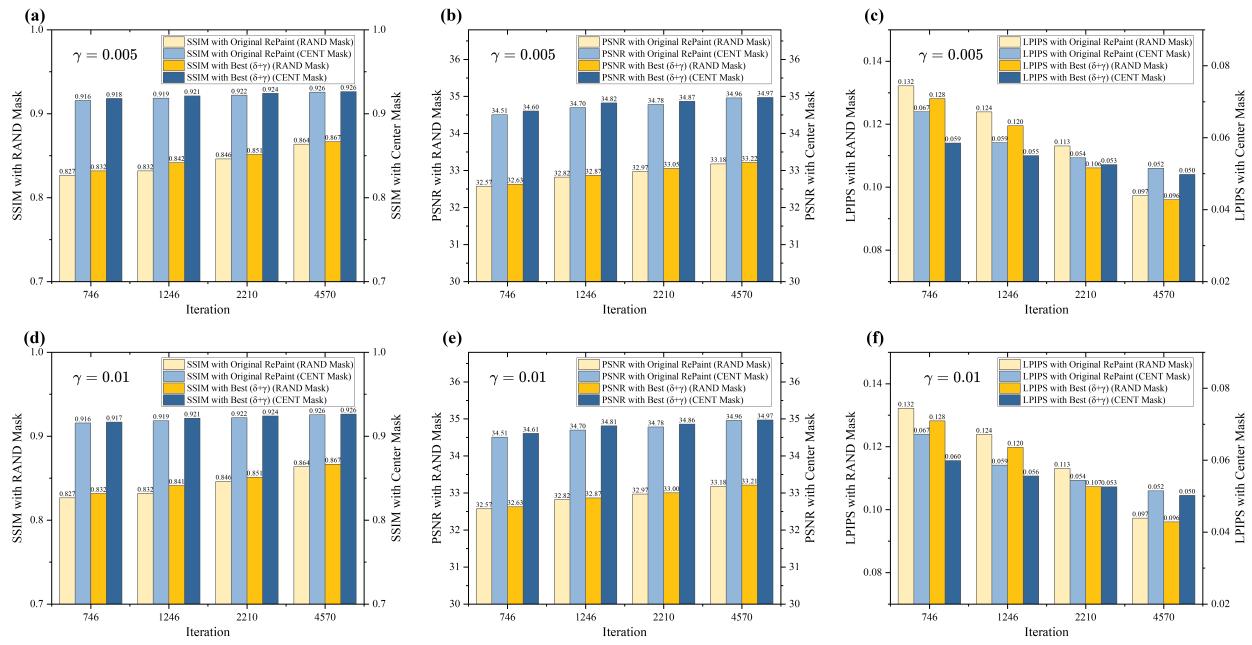


FIGURE 7. We employed three commonly used image restoration evaluation metrics, including Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Perceptual Loss Index (LPIPS), to assess our restoration results. We compared the optimal restoration outcomes under different δ values with the results of the RePaint model. The following figures illustrate the comparisons: (a) Compares the best SSIM results under different tasks with RePaint at $\gamma = 0.005$; (b) Compares the best PSNR results under different tasks with RePaint at $\gamma = 0.005$; (c) Compares the best LPIPS results under different tasks with RePaint at $\gamma = 0.005$; (d) Compares the best SSIM results under different tasks with RePaint at $\gamma = 0.01$; (e) Compares the best PSNR results under different tasks with RePaint at $\gamma = 0.01$; (f) Compares the best LPIPS results under different tasks with RePaint at $\gamma = 0.01$.

TABLE 3. The quantitative analysis results for various restoration tasks under different δ values are presented when γ is set to 0.005, including SSIM, PSNR, and LPIPS. These findings contribute to a comprehensive understanding of the impact of different δ values on restoration outcomes, aiding in the selection of the most suitable δ value for task requirements.

746 RAND	SSIM	PSNR	LPIPS	746 CENT	SSIM	PSNR	LPIPS
RePaint	0.8266	32.5728	0.1322	RePaint	0.9157	34.5070	0.0672
$\delta=10$	0.8296	32.5848	0.1305	$\delta=10$	0.9164	34.5364	0.0585
$\delta=20$	0.8319	32.6275	0.1281	$\delta=20$	0.9181	34.6037	0.0597
$\delta=30$	0.8287	32.5777	0.1301	$\delta=30$	0.9166	34.5598	0.0609
$\delta=40$	0.8209	32.5513	0.1338	$\delta=40$	0.9133	34.5274	0.0612
$\delta=50$	0.8184	32.5656	0.1372	$\delta=50$	0.9127	34.4992	0.0627
1246 RAND	SSIM	PSNR	LPIPS	1246 CENT	SSIM	PSNR	LPIPS
RePaint	0.8318	32.8188	0.1239	RePaint	0.9185	34.6970	0.0586
$\delta=10$	0.8355	32.8395	0.1217	$\delta=10$	0.9193	34.7563	0.0565
$\delta=20$	0.8417	32.8691	0.1195	$\delta=20$	0.9212	34.8237	0.0553
$\delta=30$	0.8356	32.8398	0.1208	$\delta=30$	0.9199	34.7995	0.0550
$\delta=40$	0.8359	32.8183	0.1221	$\delta=40$	0.9184	34.6816	0.0574
$\delta=50$	0.8342	32.7879	0.1241	$\delta=50$	0.9168	34.6645	0.0591
2210 RAND	SSIM	PSNR	LPIPS	2210 CENT	SSIM	PSNR	LPIPS
RePaint	0.8462	32.9689	0.1131	RePaint	0.9219	34.7816	0.0544
$\delta=20$	0.8493	32.9811	0.1108	$\delta=20$	0.9235	34.7963	0.0532
$\delta=35$	0.8514	33.0176	0.1061	$\delta=35$	0.9242	34.8581	0.0525
$\delta=40$	0.8512	33.0550	0.1075	$\delta=40$	0.9238	34.8671	0.0527
$\delta=50$	0.8472	32.9736	0.1109	$\delta=50$	0.9219	34.8196	0.0524
$\delta=70$	0.8399	32.9656	0.1145	$\delta=70$	0.9202	34.7807	0.0548
4570 RAND	SSIM	PSNR	LPIPS	4570 CENT	SSIM	PSNR	LPIPS
RePaint	0.8636	33.1785	0.0973	RePaint	0.9256	34.9590	0.0515
$\delta=40$	0.8618	33.1712	0.0978	$\delta=40$	0.9237	34.9332	0.0525
$\delta=60$	0.8643	33.1803	0.0967	$\delta=60$	0.9256	34.9680	0.0514
$\delta=70$	0.8662	33.2010	0.0962	$\delta=70$	0.9263	34.9713	0.0501
$\delta=80$	0.8667	33.2155	0.0960	$\delta=80$	0.9260	34.9684	0.0498
$\delta=100$	0.8632	33.1754	0.0975	$\delta=100$	0.9251	34.9472	0.0521

In the case of 4570CENT, the restoration results of RePaint do not align with semantic information. Our network

performs well in semantic aspects for δ values of 60, 70, and 80, but for $\delta = 40$ and 100, the semantic effect decreases.

TABLE 4. The quantitative analysis results for various restoration tasks under different δ values are presented when γ is set to 0.01, including SSIM, PSNR, and LPIPS. These findings contribute to a comprehensive understanding of the impact of different δ values on restoration outcomes, aiding in the selection of the most suitable δ value for task requirements.

746 RAND	SSIM	PSNR	LPIPS	746 CENT	SSIM	PSNR	LPIPS
RePaint	0.8266	32.5728	0.1322	RePaint	0.9157	34.5070	0.0672
$\delta=10$	0.8295	32.5955	0.1308	$\delta=10$	0.9163	34.5575	0.0626
$\delta=20$	0.8317	32.6276	0.1282	$\delta=20$	0.9168	34.6087	0.0599
$\delta=30$	0.8272	32.6201	0.1300	$\delta=30$	0.9164	34.5667	0.0611
$\delta=40$	0.8238	32.5590	0.1329	$\delta=40$	0.9133	34.5257	0.0642
$\delta=50$	0.8198	32.5630	0.1352	$\delta=50$	0.9125	34.4816	0.0685
1246 RAND	SSIM	PSNR	LPIPS	1246 CENT	SSIM	PSNR	LPIPS
RePaint	0.8318	32.8188	0.1239	RePaint	0.9185	34.6970	0.0586
$\delta=10$	0.8342	32.8322	0.1221	$\delta=10$	0.9190	34.7486	0.0571
$\delta=20$	0.8412	32.8670	0.1198	$\delta=20$	0.9214	34.8114	0.0556
$\delta=30$	0.8355	32.8386	0.1209	$\delta=30$	0.9196	34.7879	0.0560
$\delta=40$	0.8351	32.8181	0.1223	$\delta=40$	0.9184	34.6836	0.0574
$\delta=50$	0.8339	32.7570	0.1244	$\delta=50$	0.9163	34.6558	0.0598
2210 RAND	SSIM	PSNR	LPIPS	2210 CENT	SSIM	PSNR	LPIPS
RePaint	0.8462	32.9689	0.1131	RePaint	0.9219	34.7816	0.0544
$\delta=20$	0.8482	32.9858	0.1100	$\delta=20$	0.9231	34.7891	0.0538
$\delta=35$	0.8511	33.0040	0.1073	$\delta=35$	0.9239	34.8584	0.0526
$\delta=40$	0.8509	32.9995	0.1078	$\delta=40$	0.9240	34.8490	0.0528
$\delta=50$	0.8470	32.9693	0.1126	$\delta=50$	0.9219	34.8076	0.0531
$\delta=70$	0.8400	32.9561	0.1182	$\delta=70$	0.9199	34.7781	0.0552
4570 RAND	SSIM	PSNR	LPIPS	4570 CENT	SSIM	PSNR	LPIPS
RePaint	0.8636	33.1785	0.0973	RePaint	0.9256	34.9590	0.0515
$\delta=40$	0.8616	33.1698	0.0982	$\delta=40$	0.9232	34.9327	0.0528
$\delta=60$	0.8645	33.1808	0.0965	$\delta=60$	0.9256	34.9677	0.0513
$\delta=70$	0.8661	33.2109	0.0963	$\delta=70$	0.9262	34.9711	0.0502
$\delta=80$	0.8665	33.2044	0.0961	$\delta=80$	0.9263	34.9668	0.0510
$\delta=100$	0.8631	33.1680	0.0981	$\delta=100$	0.9250	34.9474	0.0530

TABLE 5. The quantitative analysis results for different restoration tasks under various δ values are presented. These results contribute to a comprehensive understanding of the impact of different δ values on restoration outcomes, aiding in the selection of the most suitable δ value for task requirements.

746 RAND	SSIM	PSNR	LPIPS	746 CENT	SSIM	PSNR	LPIPS
RePaint	0.8266	32.5728	0.1322	RePaint	0.9157	34.5070	0.0672
$\delta=10$	0.8285	32.5921	0.1307	$\delta=10$	0.9154	34.5001	0.0609
$\delta=20$	0.8320	32.6086	0.1284	$\delta=20$	0.9167	34.5281	0.0601
$\delta=30$	0.8292	32.5625	0.1303	$\delta=30$	0.9166	34.5537	0.0598
$\delta=40$	0.8262	32.6071	0.1339	$\delta=40$	0.9160	34.5217	0.0602
$\delta=50$	0.8234	32.5877	0.1355	$\delta=50$	0.9156	34.5064	0.0622
1246 RAND	SSIM	PSNR	LPIPS	1246 CENT	SSIM	PSNR	LPIPS
RePaint	0.8318	32.8188	0.1239	RePaint	0.9185	34.6970	0.0586
$\delta=10$	0.8355	32.8190	0.1230	$\delta=10$	0.9197	34.7060	0.0567
$\delta=20$	0.8414	32.8684	0.1198	$\delta=20$	0.9208	34.7372	0.0552
$\delta=30$	0.8405	32.8341	0.1202	$\delta=30$	0.9196	34.7259	0.0557
$\delta=40$	0.8381	32.8214	0.1224	$\delta=40$	0.9172	34.6895	0.0575
$\delta=50$	0.8372	32.7896	0.1245	$\delta=50$	0.9168	34.6776	0.0577
2210 RAND	SSIM	PSNR	LPIPS	2210 CENT	SSIM	PSNR	LPIPS
RePaint	0.8462	32.9689	0.1131	RePaint	0.9219	34.7816	0.0544
$\delta=20$	0.8472	32.9815	0.1071	$\delta=20$	0.9227	34.7911	0.0541
$\delta=35$	0.8508	33.0322	0.1065	$\delta=35$	0.9241	34.8405	0.0529
$\delta=40$	0.8502	33.0258	0.1084	$\delta=40$	0.9237	34.8643	0.0543
$\delta=50$	0.8499	32.9804	0.1103	$\delta=50$	0.9223	34.8170	0.0554
$\delta=70$	0.8493	32.9644	0.1110	$\delta=70$	0.9212	34.7944	0.0564
4570 RAND	SSIM	PSNR	LPIPS	4570 CENT	SSIM	PSNR	LPIPS
RePaint	0.8636	33.1785	0.0973	RePaint	0.9256	34.9590	0.0515
$\delta=40$	0.8611	33.1724	0.0975	$\delta=40$	0.9231	34.9483	0.0518
$\delta=60$	0.8636	33.1800	0.0968	$\delta=60$	0.9254	34.9674	0.0508
$\delta=70$	0.8665	33.2023	0.0965	$\delta=70$	0.9263	34.9711	0.0502
$\delta=80$	0.8671	33.2144	0.0956	$\delta=80$	0.9257	34.9696	0.0513
$\delta=100$	0.8638	33.1784	0.0969	$\delta=100$	0.9244	34.9436	0.0518

We employed three evaluation metrics (SSIM, PSNR, LPIPS) to comprehensively assess our restoration results, comparing them with those of RePaint, as shown in Table 5.

Additionally, we compared the best results with RePaint, as illustrated in Figure 9. Analysis of the data in Table 5 reveals the following observations and conclusions:



FIGURE 8. The qualitative analysis of the impact on different image restoration tasks under different δ values is presented for images (a), (b), (c), (e), (f) with respect to 746CENT(RAND) and 1246CENT(RAND) at $\delta = 10, \delta = 20, \delta = 30, \delta = 40, \delta = 50$. For 2210CENT(RAND), the analysis is provided for $\delta = 20, \delta = 35, \delta = 40, \delta = 50, \delta = 70$. Similarly, for 4570CENT(RAND), the analysis covers $\delta = 40, \delta = 60, \delta = 70, \delta = 80, \delta = 100$.

In some cases, when the total number of iterations is relatively small, such as for 746CENT(RAND), 1246CENT(RAND), and 2210CENT(RAND), our restoration performance improved in terms of SSIM, PSNR, and LPIPS by choosing smaller δ values. However, as we selected larger δ values, the restoration performance began to decline. For the case of 4570CENT(RAND), when δ is 40 and 100, our restoration results show a decreasing trend, consistent with our qualitative analysis. Across the three evaluation metrics, our restoration method outperforms RePaint, indicating that our approach provides higher restoration quality and better performance in image restoration tasks. These results further validate the effectiveness and superiority of the proposed restoration strategy.

Overall, these data suggest that when selecting δ values, there is a need to balance the relationship between the total number of iterations and the restoration results. Smaller δ values may contribute to improving the quality of restoration results for tasks with fewer total iterations, while larger δ values may decrease restoration quality. Therefore, choosing the appropriate δ values requires careful optimization and adjustment for specific tasks to achieve the best restoration results.

We further analyzed the reasons mentioned above. If the value of δ is too small, there are fewer steps to correct x_t . If the value of δ is too large, there are more steps to correct x_t . Through qualitative and quantitative analysis of δ , we found that a larger δ value is not necessarily better.

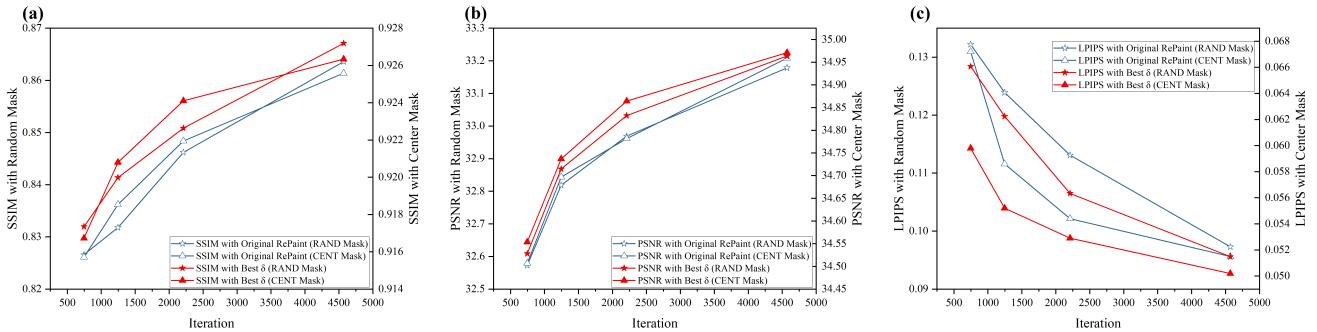


FIGURE 9. We used three commonly used image restoration evaluation metrics, including SSIM, PSNR, and LPIPS, to assess our restoration results. The optimal restoration results under different δ values were compared with the results of the RePaint model. The figures below show the comparisons: (a) compares the best SSIM results under different tasks with RePaint; (b) compares the best PSNR results under different tasks with RePaint; (c) compares the best LPIPS results under different tasks with RePaint.

For 746CENT(RAND) and 1246CENT(RAND), we found that a δ value of 20 produces better restoration results. For 2210CENT(RAND), a δ value of 35 leads to improved restoration results. For 4570CENT, a δ value of 70 yields better restoration results. Lastly, for 4570RAND, a δ value of 80 produces better restoration results.

2) SAMPLING SCHEDULING

In Section III-B, we conducted a detailed analysis of the choice of sampling stages and the selection of the sampling schedule lower bound. In this subsection, we delve into the impact of the sampling schedule decay rate on the restoration results, providing a comprehensive display of the influence of the decay rate γ on the restoration results through qualitative and quantitative results. Qualitative results are shown in Figure 10, while quantitative results are presented in Table 6.

From the qualitative results, we observe an improvement in restoration quality when γ is set to 0.005 and 0.01. However, the restoration quality degrades when γ is set to 0.02. For 746RAND, RePaint's attempt to restore glasses is semantically incorrect, but when the decay rate γ is 0.005, the glasses are successfully restored; when γ is 0.01, the repaired glasses still have defects; and when γ is 0.02, the hair on the right side is semantically incorrect. For 746CENT, the results repaired by RePaint are semantically incorrect, but with γ set to 0.005 and 0.01, the restoration results of our network are good; when γ is 0.02, the hair restored on the right side is semantically incorrect.

For 1246RAND, RePaint's attempt to restore teeth is unsuccessful, while our network performs well with γ set to 0.005, but when γ is 0.01, there are some flaws in the upper lip; when γ is 0.02, the restoration results are poor for the upper lip, right hair, and earring. For 1246CENT, RePaint's attempt to restore glasses is semantically incorrect, but our network performs well with γ set to 0.005; when γ is 0.01, the results show aging and wrinkles; when γ is 0.02, semantic quality decreases.

For 2210RAND, the background restored by RePaint is semantically incorrect, and our network performs well with γ set to 0.005 and 0.01, but when γ is 0.02, the restoration

of the hair on the right side fails. For 2210CENT, RePaint's attempt to restore pupils is inconsistent, while our network performs well with γ set to 0.005 and 0.01; when γ is 0.02, the facial restoration quality decreases.

For 4570RAND, RePaint's attempt to restore hair is semantically incorrect, but our network performs well with γ set to 0.005 and 0.01, while the restoration fails when γ is 0.02. For 4570CENT, our network performs better than RePaint in terms of image consistency with γ set to 0.005 and 0.01, but the restoration is poor when γ is 0.02.

We employed three evaluation metrics (SSIM, PSNR, LPIPS) to assess our restoration results, comparing our best outcomes with those of RePaint, as depicted in Figure 11. Across the three evaluation metrics, our restoration method exhibits higher performance compared to RePaint. This suggests that our approach offers superior restoration quality and better overall performance in image restoration tasks. These results further validate the effectiveness and superiority of the proposed restoration strategy.

Our analysis indicates that as the value of γ increases, i.e., a larger decay rate, the model's output sharply decreases, leading to a decline in the restoration capability of our network. As shown in Table 6, for the 8 RAND/CENT scenarios, $\gamma = 0.005$ generally achieves the most significant improvement. However, when γ is set to 0.02, the metrics almost universally decline. This aligns with our qualitative analysis, suggesting that the restoration quality tends to be poorer when γ is set to 0.02.

In summary, in this section, we delved into the impact of the decay rate γ in the sampling schedule strategy on image restoration effects. Through qualitative and quantitative results, we have clarified the following three key points:

Firstly, different image restoration tasks exhibit varying sensitivities to the decay rate γ , necessitating a task-specific selection. For instance, in the 2210RAND task, $\gamma = 0.005$ consistently yielded the best improvement.

Secondly, the choice of the decay rate γ directly influences the model's output, with larger γ values causing a sharp reduction in output and, consequently, a decrease in restoration capability. Therefore, adjusting γ requires a balance between restoration effectiveness and output stability.



FIGURE 10. These visual analyses illustrate the qualitative results for different γ values across various image restoration tasks. The aim is to showcase the impact of the decay rate γ on different image restoration tasks. Through these visual examples, one can clearly observe the visual characteristics and differences in the restoration results under different γ values. These visual illustrations provide an intuitive understanding, aiding in a more comprehensive evaluation and comparison of the restoration effects under different γ values.

Lastly, the concordance between qualitative observations and quantitative evaluations underscores the critical role of

γ selection in image restoration tasks. Considering these factors collectively allows for a more precise adjustment of

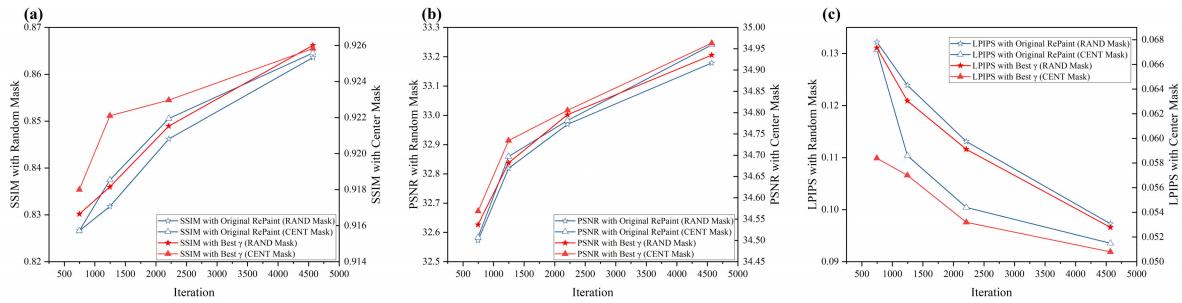


FIGURE 11. We utilized three common image restoration evaluation metrics, including SSIM, PSNR, and LPIPS, to assess our restoration results. The best restoration outcomes under different γ values were compared with the results from the RePaint model, as illustrated in Figure 8. Subfigures (a), (b), and (c) present the best SSIM, PSNR, and LPIPS comparisons with RePaint across various restoration tasks, respectively. The results consistently demonstrate that our restoration method outperforms RePaint in terms of these evaluation metrics, emphasizing the effectiveness of our proposed restoration strategy.

TABLE 6. We presented a detailed quantitative analysis of the impact of different decay rates (γ) on restoration results. We utilized three common image quality assessment metrics: SSIM, PSNR, and LPIPS. These metrics allowed us to objectively evaluate the quality of restoration results under different γ values. The data in the tables clearly demonstrate how the choice of γ influences the restoration outcomes. It is evident that different γ values lead to distinct restoration effects in various tasks. Through careful experimentation and analysis, we ultimately selected the optimal γ values to achieve the best restoration performance.

746 RAND	SSIM	PSNR	LPIPS	746 CENT	SSIM	PSNR	LPIPS
RePaint	0.8266	32.5728	0.1322	RePaint	0.9157	34.5070	0.0672
$\gamma=0.005$	0.8279	32.6264	0.1319	$\gamma=0.005$	0.9161	34.5408	0.0622
$\gamma=0.01$	0.8302	32.5768	0.1311	$\gamma=0.01$	0.9180	34.5686	0.0584
$\gamma=0.02$	0.8229	32.5600	0.1336	$\gamma=0.02$	0.9135	34.5191	0.0651
1246 RAND	SSIM	PSNR	LPIPS	1246 CENT	SSIM	PSNR	LPIPS
RePaint	0.8318	32.8188	0.1239	RePaint	0.9185	34.6970	0.0586
$\gamma=0.005$	0.8347	32.8375	0.1209	$\gamma=0.005$	0.9205	34.7056	0.0570
$\gamma=0.01$	0.8360	32.8332	0.1216	$\gamma=0.01$	0.9221	34.7345	0.0578
$\gamma=0.02$	0.8305	32.7731	0.1231	$\gamma=0.02$	0.9153	34.6620	0.0600
2210 RAND	SSIM	PSNR	LPIPS	2210 CENT	SSIM	PSNR	LPIPS
RePaint	0.8462	32.9689	0.1131	RePaint	0.9219	34.7816	0.0544
$\gamma=0.005$	0.8490	33.0020	0.1116	$\gamma=0.005$	0.9230	34.8054	0.0541
$\gamma=0.01$	0.8472	32.9835	0.1126	$\gamma=0.01$	0.9223	34.7974	0.0532
$\gamma=0.02$	0.8456	32.9589	0.1142	$\gamma=0.02$	0.9167	34.7437	0.0586
4570 RAND	SSIM	PSNR	LPIPS	4570 CENT	SSIM	PSNR	LPIPS
RePaint	0.8636	33.1785	0.0973	RePaint	0.9256	34.9590	0.0515
$\gamma=0.005$	0.8662	33.2055	0.0966	$\gamma=0.005$	0.9258	34.9631	0.0508
$\gamma=0.01$	0.8641	33.1849	0.0972	$\gamma=0.01$	0.9256	34.9611	0.0512
$\gamma=0.02$	0.8604	33.0305	0.1112	$\gamma=0.02$	0.9126	34.7940	0.0753

TABLE 7. Comparison of Face Recognition Accuracy Before and After Face Image Restoration.

Task	Accuracy	Task	Accuracy
Rand Masked	81.3%	Cent Masked	0%
746 RAND	92%	746 CENT	16%
1246 RAND	94%	1246 CENT	22%
2210 RAND	96%	2210 CENT	20%
4570 RAND	97%	4570 CENT	26%

γ , achieving optimal restoration results and enhancing the successful application of deep learning in image restoration tasks.

E. APPLICATION ANALYSIS

We conducted facial recognition on images before and after restoration, calculating accuracy as presented in Table 7. It is evident from the table that the post-restoration facial images exhibit a higher recognition accuracy, indicating a significant improvement.

V. DISCUSSION

Through the exploration of face image restoration methods based on diffusion models, our study contributes novel insights to the field of image restoration. Experimental results demonstrate the excellent performance of DDPM in facial image restoration, particularly in effectively addressing extensive missing areas. By enhancing the sampling strategy through the optimized reverse process of progressive sampling and sampling scheduling, we have improved the quality of image restoration using DDPM.

However, the study has its limitations. Our method still requires considerable computational resources, potentially limiting its practical application. Future research could explore more efficient computing methods [49], [50]. Future research directions include optimizing the decay rate (γ) to enhance image restoration quality and computational efficiency. Additionally, further investigation is needed on adapting to diverse facial features and scenarios, aiming for more intelligent facial image restoration.

VI. CONCLUSION

We propose an enhanced facial image restoration method based on DDPM. This approach introduces the Resample sampling strategy, combining progressive sampling with sampling scheduling to optimize the reverse process of DDPM and improve the quality of restored images. Additionally, our method is not specifically trained for a particular mask, making it adaptable to any mask, thereby enhancing the flexibility of facial image restoration methods. The results show improvements in both PSNR and LPIPS metrics. Furthermore, an application analysis indicates a significant enhancement in face recognition accuracy after the restoration of facial images.

REFERENCES

- [1] Z. Xu, X. Lian, and L. Feng, "Image inpainting algorithm based on partial differential equation," in *Proc. ISECS Int. Colloq. Comput., Commun., Control, Manag.*, 2008, pp. 120–124.
- [2] C.-B. Schönlieb, "Applying modern PDE techniques to digital image restoration," 2012. [Online]. Available: <https://www.mathworks.com/company/newsletters/articles/applying-modern-pde-techniques-to-digital-image-restoration.html>
- [3] Z. Xie, F. Zhang, and C. Zhang, "An adaptive matching algorithm for image inpainting," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Sep. 2011, pp. 1293–1296.
- [4] Y.-T. Zhuang, Y.-S. Wang, T. K. Shih, and N. C. Tang, "Patch-guided facial image inpainting by shape propagation," *J. Zhejiang Univ.-Sci. A*, vol. 10, no. 2, pp. 232–238, Feb. 2009.
- [5] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," 2018, *arXiv:1803.07422*.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. ICLR*, 2016, pp. 1–16.
- [8] K. Qiao, J. Chen, L. Wang, C. Zhang, L. Tong, and B. Yan, "BigGAN-based Bayesian reconstruction of natural images from human brain activity," *Neuroscience*, vol. 444, pp. 92–105, Sep. 2020.
- [9] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.
- [10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
- [11] T.-D. Nguyen, B. Kim, and M.-C. Hong, "New hole-filling method using extrapolated spatio-temporal background information for a synthesized free-view," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1345–1358, Jun. 2019.
- [12] M.-Y. Liu, X. Huang, J. Yu, T.-C. Wang, and A. Mallya, "Generative adversarial networks for image and video synthesis: Algorithms and applications," *Proc. IEEE*, vol. 109, no. 5, pp. 839–862, May 2021.
- [13] Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva, "ImaGINator: Conditional spatio-temporal GAN for video generation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1149–1158.
- [14] M. A. Haidar and M. Rezagholizadeh, "Textkd-GAN: Text generation using knowledge distillation and generative adversarial networks," in *Advances in Artificial Intelligence*. Kingston, ON, Canada: Springer, 2019, pp. 107–118.
- [15] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: Better text generation via filling in the _," 2018, *arXiv:1801.07736*.
- [16] R. Durall, A. Chatzimichailidis, P. Labus, and J. Keuper, "Combating mode collapse in GAN training: An empirical analysis using Hessian eigenvalues," 2020, *arXiv:2012.09673*.
- [17] Y. Yu, L. Zhang, H. Fan, and T. Luo, "High-fidelity image inpainting with GAN inversion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 242–258.
- [18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.
- [19] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.
- [20] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv:2010.02502*.
- [21] G. Kim, T. Kwon, and J. C. Ye, "DiffusionCLIP: Text-guided diffusion models for robust image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2416–2425.
- [22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.
- [23] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "ILVR: Conditioning method for denoising diffusion probabilistic models," 2021, *arXiv:2108.02938*.
- [24] Z. Dorjsembe, S. Odonchimed, and F. Xiao, "Three-dimensional medical image synthesis with denoising diffusion probabilistic models," in *Proc. Med. Imag. Deep Learn.*, 2022.
- [25] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-Prompt image editing with cross attention control," 2022, *arXiv:2208.01626*.
- [26] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.
- [27] M. Styulkowski, K. Vougioukas, S. He, M. Zieba, S. Petridis, and M. Pantic, "Diffused heads: Diffusion models beat GANs on talking-face generation," 2023, *arXiv:2301.03396*.
- [28] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using denoising diffusion probabilistic models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11451–11461.
- [29] H. Hukkelås, F. Lindseth, and R. Mester, "Image inpainting with learnable feature imputation," in *Pattern Recognition*. Tübingen, Germany: Springer, 2021, pp. 388–403.
- [30] T. Auciukevicius, Z. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, and P. Guerrero, "RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12608–12618.
- [31] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Guided image synthesis and editing with stochastic differential equations," 2021, *arXiv:2108.01073*.
- [32] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
- [33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [34] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
- [35] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision—ECCV*. Amsterdam, The Netherlands: Springer, 2016, pp. 694–711.
- [36] L. Rout, A. Parulekar, C. Caramanis, and S. Shakkottai, "A theoretical justification for image inpainting using denoising diffusion probabilistic models," 2023, *arXiv:2302.01217*.
- [37] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," 2021, *arXiv:2112.10741*.
- [38] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 23593–23606.
- [39] L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer, "Lossy compression with Gaussian diffusion," 2022, *arXiv:2206.08889*.
- [40] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *Proc. ACM SIGGRAPH 2022 Conf.*, 2022, pp. 1–10.
- [41] B. Li, K. Xue, B. Liu, and Y.-K. Lai, "BBDM: Image-to-image translation with Brownian bridge diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1952–1961.
- [42] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

- [43] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (CelebA) dataset," *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.
- [44] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "EdgeConnect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*.
- [45] X. Guo, H. Yang, and D. Huang, "Image inpainting via conditional texture and structure dual generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14114–14123.
- [46] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.
- [47] Y. Poirier-Ginter and J.-F. Lalonde, "Robust unsupervised StyleGAN image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22292–22301.
- [48] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4672–4681.
- [49] E. Luhman and T. Luhman, "Denoising synthesis: A module for fast image synthesis using denoising-based models," *Softw. Impacts*, vol. 9, Aug. 2021, Art. no. 100076.
- [50] E. Luhman and T. Luhman, "Knowledge distillation in iterative generative models for improved sampling speed," 2021, *arXiv:2101.02388*.



LIBO HE received the master's degree in computer science and the Ph.D. degree from Yunnan University, in 2006 and June 2018, respectively. She is currently a Lecturer with the Information Network Security College and the Yunnan Police Officer College, China. She has performed research in distributed task scheduling and video processing. Her current research interests include vision computing and machine learning.



HONG LIN received the Ph.D. degree from Mocam Polytechnic University, in June 2023.

She is currently an Associate Professor with the College of Big Data and Intelligence Engineering, Southwest Forestry University, China. Her research interests include computer vision, deep learning, and few-shot learning.



YUN PANG was born in Sichuan, China, in 1999. He is currently with the College of Big Data and Intelligence Engineering, Southwest Forestry University. His research interests include the field of vision computing and image processing.



JIAWEI MAO was born in Henan, China, in 1999. He is currently with the College of Big Data and Intelligence Engineering, Southwest Forestry University. His research interests include the field of image processing and NLP.



ZHENPING QIANG received the master's degree in computer science from the Kunming University of Science and Technology, in 2006, and the Ph.D. degree from Yunnan University, in June 2018. He is currently a Professor with the College of Big Data and Intelligence Engineering, Southwest Forestry University, China. He has performed research in image processing and artificial intelligence. His current research interests include vision computing and big data processing.

• • •