

Building Bridge Across the Time: Disruption and Restoration of Murals In the Wild (Supplementary Material)

Paper ID 9367

A. Conditional Denoising Diffusion Model

In this section, we will give more details about Conditional Denoising Diffusion Model. Given a damaged mural image \mathbf{x}^d and heatmap \mathbf{x}^a , we want to learn a parametric approximation to $p(\mathbf{x}|\mathbf{x}^d, \mathbf{x}^a)$ through a stochastic iterative process. In other words, $p(\mathbf{x}|\mathbf{x}^d, \mathbf{x}^a)$ indicates a mapping from damaged image \mathbf{x}^d to restored image \mathbf{x} .

Forward process. Let $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ be the distribution of intermediate process in the forward chain. We define the forward diffusion process $(\mathbf{x}_0, \dots, \mathbf{x}_t, \dots, \mathbf{x}_T)$ as iteratively adds Gaussian noise to the image via a fixed Markov chain (as shown in Fig. 6). Then, we give the definition of forward process:

$$\begin{aligned} \mathbf{x}_t &= \alpha_t \mathbf{x}_{t-1} + \sqrt{\beta_t} \boldsymbol{\epsilon}_t \\ &= \alpha_t (\alpha_{t-1} \mathbf{x}_{t-2} + \sqrt{\beta_{t-1}} \boldsymbol{\epsilon}_{t-1}) + \sqrt{\beta_t} \boldsymbol{\epsilon}_t \\ &= \alpha_t \alpha_{t-1} \mathbf{x}_{t-2} + \sqrt{(1 - \beta_{t-1}) \beta_t} \boldsymbol{\epsilon}_{t-1} + \sqrt{\beta_t} \boldsymbol{\epsilon}_t \\ &= \alpha_t \alpha_{t-1} \mathbf{x}_{t-2} + \sqrt{(1 - \beta_t \beta_{t-1})} \boldsymbol{\epsilon} \\ &= \dots \\ &= (\alpha_t \dots \alpha_1) \mathbf{x}_0 + \sqrt{1 - (\alpha_t \dots \alpha_1)^2} \boldsymbol{\epsilon} \\ &= \bar{\alpha}_t \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t^2} \boldsymbol{\epsilon}, \end{aligned} \quad (7)$$

Following the result in [22], if T is large enough, the reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a)$ has the same distribution of forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Then, we give the definition of the forward process:

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t|\sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I}) \\ q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t|\sqrt{\gamma_t} \mathbf{x}_0, (1 - \gamma_t) \mathbf{I}), \end{aligned} \quad (8)$$

where the diffusion parameter $0 < \alpha_t < 1$, $\alpha_t = \sqrt{1 - \beta_t}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\gamma_t = \bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. By rearranging the terms, we have:

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}|\boldsymbol{\mu}_t, \sigma_t^2 \mathbf{I}) \\ \boldsymbol{\mu}_t &= \frac{\sqrt{\gamma_{t-1}}(1 - \alpha_t)}{1 - \gamma_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \gamma_{t-1})}{1 - \gamma_t} \mathbf{x}_t \\ \sigma_t^2 &= \frac{(1 - \gamma_{t-1})(1 - \alpha_t)}{1 - \gamma_t} \end{aligned} \quad (9)$$

Optimizing the Denoising Model. Benefiting from the additivity of Gaussian distribution, given step t , we can es-

timate the noisy target image \mathbf{x}_t :

$$\mathbf{x}_t = \sqrt{\gamma_t} \mathbf{x}_0 + \sqrt{1 - \gamma_t} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (10)$$

The goal of CDDPM is to recover the target image \mathbf{x}_0 . Let $f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t)$ be denoising model. Following [43], the objective function is minimizing the error between $\boldsymbol{\epsilon}$ and model output:

$$\mathbb{E}_{(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t)} \mathbb{E}_{\boldsymbol{\epsilon}, \gamma_t} \|f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t) - \boldsymbol{\epsilon}\|_2^2. \quad (11)$$

Reverse process. Furthermore, for reverse process of CDDPM, starting from Gaussian noise \mathbf{x}_T , we have:

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}|\mathbf{x}^d, \mathbf{x}^a) &= p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a) \\ p(\mathbf{x}_T) &= \mathcal{N}(\mathbf{x}_T|\mathbf{0}, \mathbf{I}) \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a) &= \mathcal{N}(\mathbf{x}_{t-1}|\mu_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t), \sigma_t^2 \mathbf{I}). \end{aligned}$$

Reverse step $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a)$ aims to denoise the noisy image \mathbf{x}_t for given damaged image \mathbf{x}^d and heatmap \mathbf{x}^a . According to the above reverse process, given \mathbf{x}_t , we can calculate the approximated $\hat{\mathbf{x}}_0$ by:

$$\hat{\mathbf{x}}_0 = \frac{1}{\sqrt{\gamma_t}} \left(\mathbf{x}_t - \sqrt{1 - \gamma_t} f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t) \right), \quad (12)$$

where $f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t)$ is used to estimate $\boldsymbol{\epsilon}$ at iteration t . Then, we can estimate the mean of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a)$:

$$\mu_\theta(\mathbf{x}^d, \mathbf{x}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \gamma_t}} f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t) \right). \quad (13)$$

Following the formulation of [22], we can estimate the mean of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}^d, \mathbf{x}^a)$:

$$\mu_\theta(\mathbf{x}^d, \mathbf{x}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \gamma_t}} f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t) \right). \quad (14)$$

Then we can use the above approximation to estimate the \mathbf{x}_{t-1} :

$$\begin{aligned} \mathbf{x}_{t-1} &\leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \gamma_t}} f_\theta(\mathbf{x}^d, \mathbf{x}^a, \mathbf{x}_t, \gamma_t) \right) \\ &\quad + \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}_t, \end{aligned} \quad (15)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

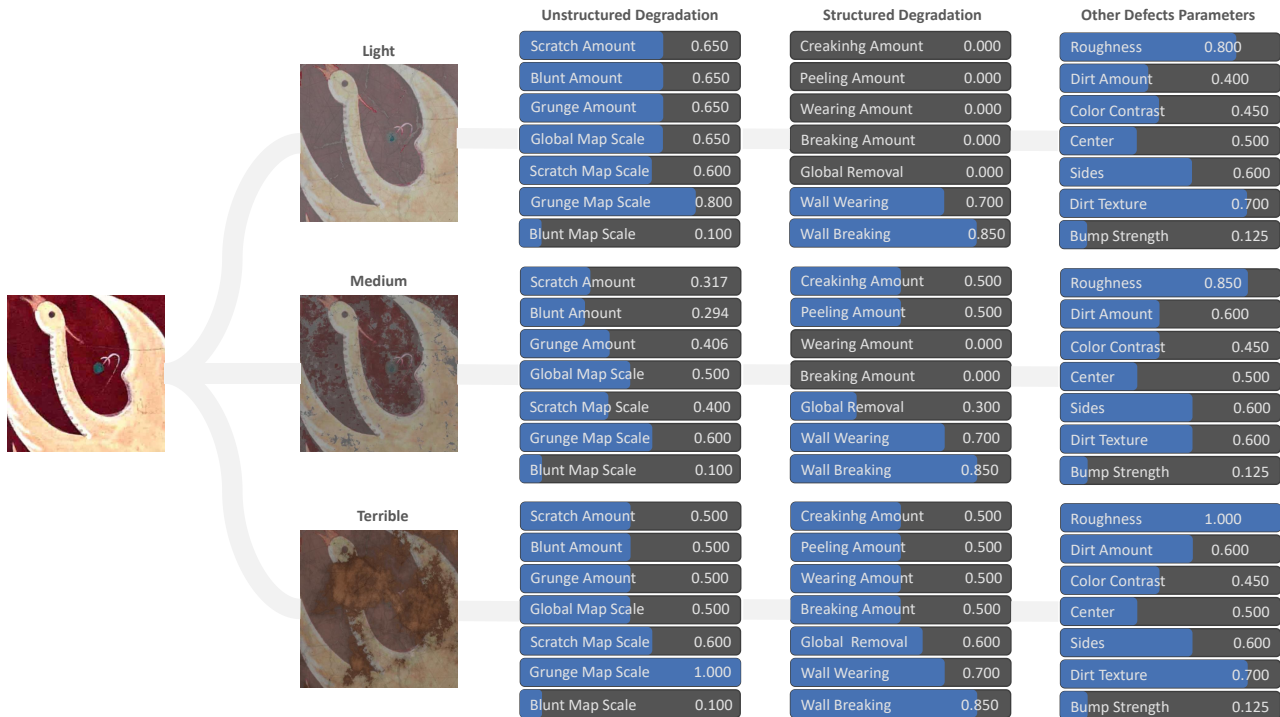


Figure 11: An overview of parameters of subsets.

B. Dataset Details

B.1. Subset Details

By controlling different parameters of degradation, we roughly construct three subsets of data with different degrees of damage, including Light, Medium and Terrible Damage Subset.

- **Light Damage Subset.** The first dataset we built prefers to damage mural images with unstructured degradation. We take color degradation and scratches as the main damage methods and use a small amount of grunge to block the texture of the mural. For structured degradation, we take a little wearing defect to the edge of the plane model. The light damage dataset's defect is simple but harder than the degradation in the old photo.
- **Medium Damage Subset.** The second dataset we built considers both structured and unstructured degradation. The amount of color degradation, scratch, and grunge will be taken randomly within a high interval. Meanwhile, structured degradation like cracking and peeling are added and set at a low value. Compared to the light damage subset, this subset contains more missing parts that need to be repainted.
- **Terrible Damage Subset.** The third dataset we built contains amounts of all types of structured and unstructured degradation. Benefiting from `blender`'s

powerful texture engine, defects and image contents were perfectly blended together. Cracking, peeling, wearing, and breaking will cause extensive irregular holes. Unlike the previous two subsets, this subset is more closely related to the real-world damaged mural.

The control parameters of three subset (Light, Medium, Terrible) are shown in Fig. 11. We present the effects of each parameter in the framework for generating damage mural (Fig. 16, Fig. 17, Fig. 18). All control parameters are divided into three categories: Unstructured Degradation, Structured Degradation and Other Defects Parameters.

- **Unstructured Degradation.** This set of parameters control some unstructured degradation. The amount parameters, *e.g.*, *Scratch Amount*, *Blunt Amount* and *Grunge Amount* control the number of areas for degradation (row 1-3 in Fig. 16, higher value stands for more severe). The scale parameters, *e.g.*, *Global Map Scale*, *Scratch Map Scale*, *Grunge Map Scale* and *Blunt Map Scale* control the scale of degradation (row 4-7).
- **Structured Degradation.** This set of parameters control some structured degradation. The amount parameters, *e.g.*, *Creaking Amount*, *Peeling Amount*, *Wearing Amount* and *Breaking Amount* control the damage extent of 3d model (row 1-4 in Fig. 17). *Global Removal* defects will randomly delete pieces from mural plane (fifth row). *Wall Wearing* and *Wall Cracking* can damage the base wall model (row 6-7).

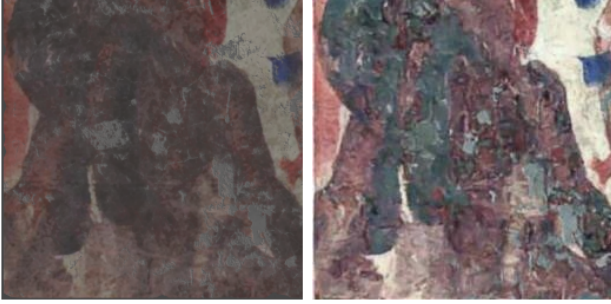


Figure 12: **The effect of Light base parameter on real-world mural restoration.** The left image is a real-world damaged mural and the right is the restoration result.

- **Other Defects Parameters.** This set of parameters controls some dirt texture. Roughness describes the variety of dirt texture (first row in Fig. 18). *Dirt Amount* controls the number of areas for dirt (second row). *Color Contrast* controls the contrast between dirt and wall (third row). *Center/Sides* describe the amount of dirt in top/sides of the base wall model and mural plane (row 4-5). *Dirt Texture* can change the color and shape of dirt (sixth row). *Bump Strength* controls the depth of dirt (last row).

We set parameters in Fig. 11 as **mean** value (e.g., Scratch Amount:0.5). Given a restored image, all control parameters are randomly tuned with a uniform distribution $U(\mathbf{mean}-0.15, \mathbf{mean}+0.15)$.

We compared the generated mural with the actually damaged mural. As shown in Fig. 15, we have implemented most of the damaging effects. For unstructured defects, scratch and blunt results are satisfactory. Especially for the color degradation, almost the same as the actual damaged mural effect. However, blunt defects are hard to simulate the real damage. For structured defects, our proposed framework perfectly mimics the physical damage of a real mural. Influenced by wearing and breaking effects, the generated mural image contains many missing areas.

Nonetheless, our generated damaged mural image still has a gap from the actually damaged mural. Some defects such as bacteria, wind erosion, and water stains are still difficult to simulate in the existing framework. Moreover, some dirt defects still need to design carefully to achieve a more realistic effect. Hence, in future work, we will pay more attention to improving the reality of generated images to reduce the gap between synthesis and reality.

B.2. Hyper-parameter Experiments

For the construction of synthetic datasets, a critical thing is how the proposed simulation with these adjustable affects the final restoration. It's important to choose the correct degradation hyper-parameters to build datasets for better

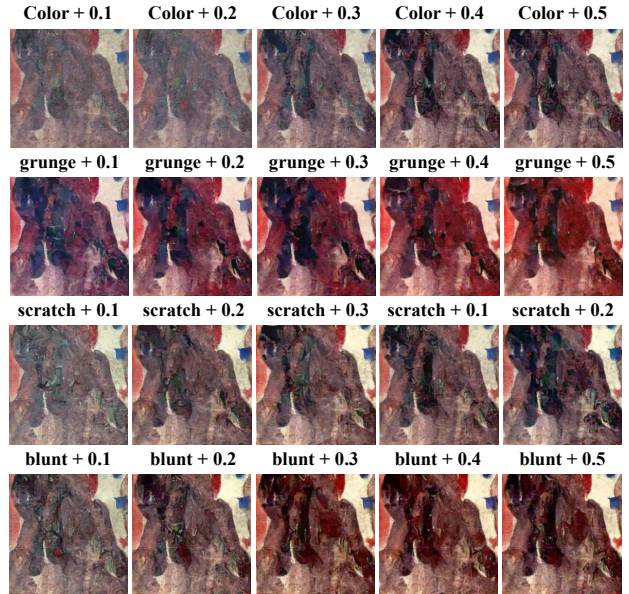


Figure 13: The effect of differently unstructured degradation hyper-parameters on real-world mural restoration.

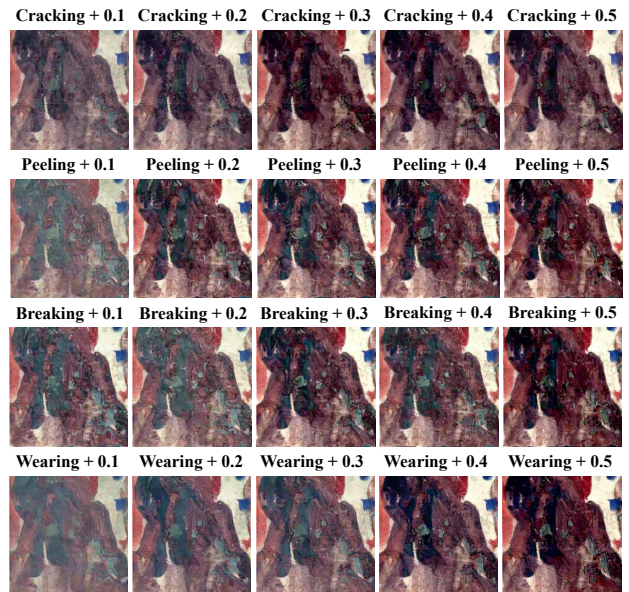


Figure 14: The effect of differently structured degradation hyper-parameters on real-world mural restoration.

real-world mural restoration performance. In this subsection, we conduct some experiments to explore the parameters of the optimal defect. Following Fig. 11, we set Light as the base degradation hyper-parameter. We constructed different datasets by offsetting the mean values of each of the eight underlying parameters (e.g., color + 0.1 means adding 0.1 from the mean value of all color-related parameters in Light). We train the ADF model on these datasets and evaluate the performance on real-world dataset.

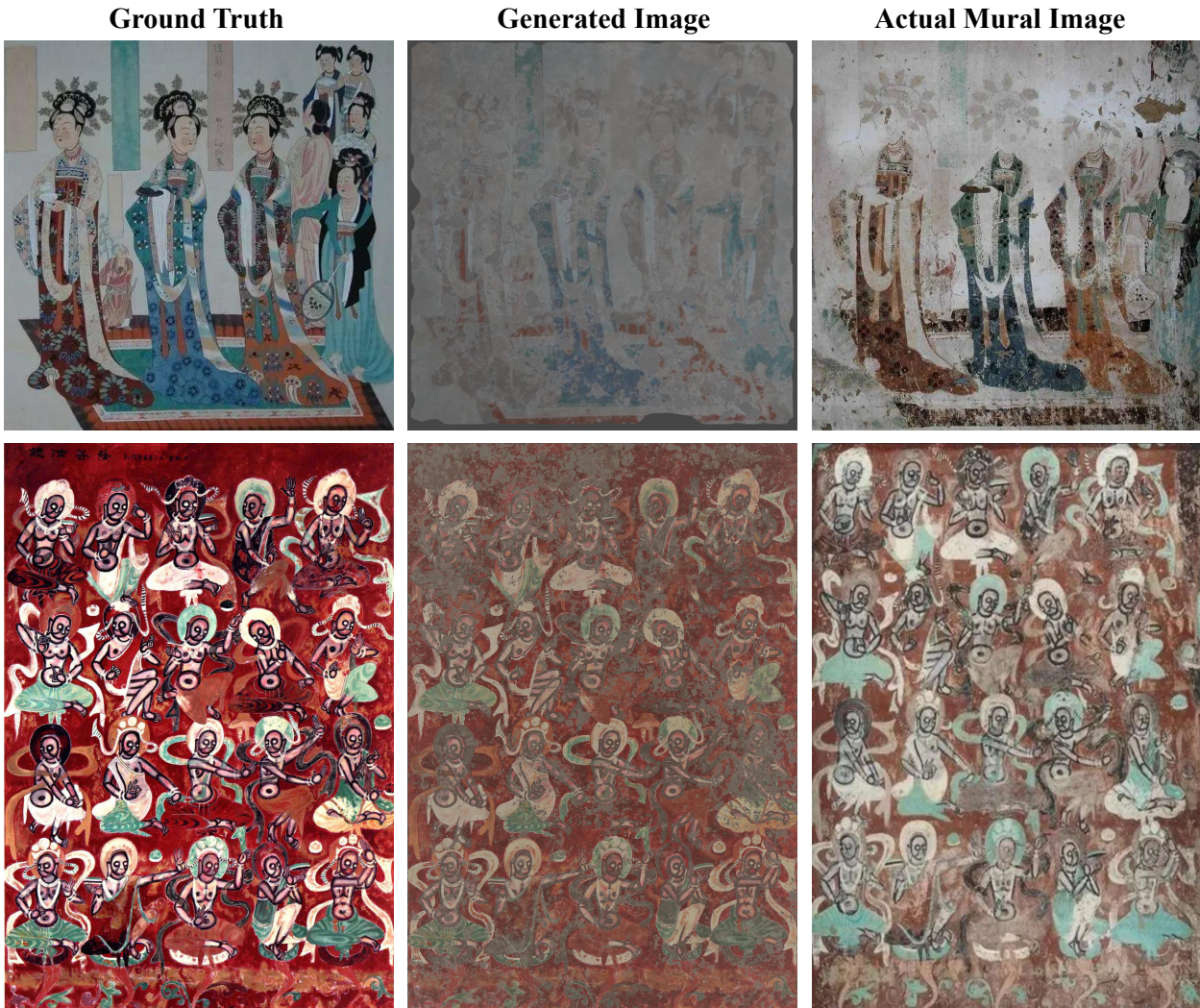


Figure 15: The comparison between the generated damage and the real damage image.

For comparison, we give the real-world results obtained after restoration training of the dataset generated by the Light base parameter (shown in Fig. 12). Since the light parameters in each degradation are not high, it does not match the real-world situation of mural damage, we can find that the model does not generate satisfactory results. Then we tune all types of decay parameters. The experiment result is shown in Fig. 13 and Fig. 14. We can make the following remarks:

(1) for unstructured degradation in Fig. 13, as the degradation parameter increases, we find that color decay and grunge can significantly improve the color correction and sharpness of the results, respectively. When the weight of color decay and grunge in the training image increases, the model can obtain better robustness and thus generate stable and high-quality restoration results. Hence, we infer that color decay and grunge parameters are important to restoration.

(2) for structured degradation in Fig. 14, when cracking

and peeling parameters is small, the model can not process the terrible damage region in the mural image. However, as the parameter values grow, the quality of the generated images rapidly improves. As the weight of structured degradation parameters rises in the training images, the model can better learn how to locate broken areas for repair. This is the reason why the overall result becomes better when the structured degradation parameters are raised. Hence, higher cracking and peeling can help models significantly improve their ability to repaint patterns.

(3) for scratch, blunt, breaking, and wearing, these degradation can help the model focus on processing noisy detail which is more suitable for the restoration of the complex damaged murals. Therefore, setting these parameters at a medium value can effectively improve the detail of the restoration results (e.g., cloth texture, hair, and color) in most situations. In real-world mural restoration tasks, these parameters can be specifically set to bring the training images closer to reality by analyzing them according to the

analysis of mural damage.

C. Experiment Details

C.1. Competitors.

We compare ADF with recent SOTA methods, including Old Photo Restoration [46], DDRM [27], SwinIR [32] and Real-ESRGAN [43].

- **Old Photo Restoration** [46] is a method that restores old photos that suffer from multiple types of degradations through training variational autoencoders;
- **DDRM** [27] proposes a practical framework to complete conditional image-to-image task, *e.g.*, image restoration, haze and noise removal;
- **SwinIR** [32] is a strong baseline model which aims to restore high-quality images from damaged images;
- **Real-ESRGAN** [43] introduce a blind super-resolution framework to remove unknown and complex degradations based on a generative adversarial network.
- **Restormer** [58] presents an efficient method that can capture long-range pixel interactions for large image restoration task.
- **All-In-One** [29] presents an all-in-one method that could recover images from a variety of unknown corruption types and levels.

C.2. Experiment Setup

In this subsection, we add some details about the experiment. In ablation studies about backbone, we train and evaluate the both types of origin model on our dataset. The CGAN model is trained on the same environment with all other baselines (software and hardware). We train the CGAN generator and discriminator 50000 iterations with a batch size of 64 (8 per GPU). For other hyper parameters, we following the origin paper [37].

C.3. Additional Experiment.

The qualitative results of mural-restoration experiments are shown in Fig. 19, Fig. 20 and Fig. 21. From the results of the figure, we can conclude the following remarks: Our algorithm generated a continuous stroke in the heavily damaged area and naturally connected it to the other less damaged content (*e.g.*, first and third row in Fig. 19). However, in some special cases, compared with the manually restored murals, the content generated by our method is only style matched, not semantic matched (*e.g.*, eighth row). Since the input image is only a small part of the overall mural, it is difficult for even humans to understand the semantics of the

contents in the image. In some damaged images with relatively complete semantic information (*e.g.*, the last row), our algorithm can accurately fill in the missing content in terms of style and semantics.

(2) In some damaged images with relatively complete semantic information (*e.g.*, the last row in Fig. 20), our algorithm can accurately fill in the missing content in terms of pixels and semantics. However, in the same situation, some of the generated areas still have some distorted or cluttered content (*e.g.*, second and fourth row). This demonstrates that the location of the damaged area and the degree to which it is critical to the overall semantic information will affect the final repair result.

(3) The guided diffusion model (*e.g.*, the last column in Fig. 21) prefers to restore the missing content with black pixels. This situation means that the model only learns how to automatically locate the damaged areas, but not how to fill in the missing areas based on the surrounding content (*e.g.*, first and second row). Moreover, in some cases, the tone of the restored images is far from the ground truth (*e.g.*, fifth and sixth row). This demonstrates that the model misinterpreted the correct pixel in the damaged image as the noise pixel. Hence, comparing our algorithm with original guided diffusion model, it is clear that our proposed components can address these problems.

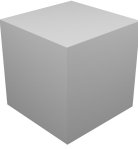


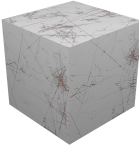
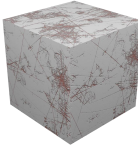
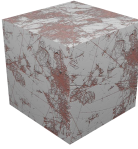
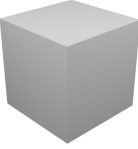


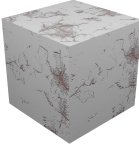

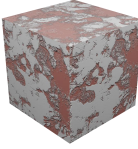
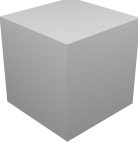
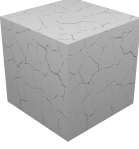
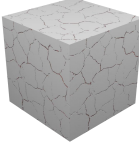
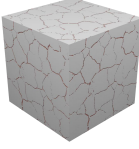
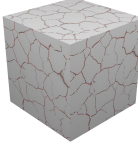
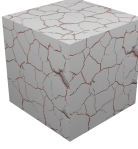
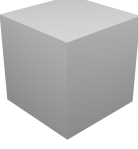



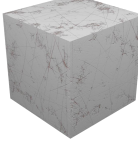
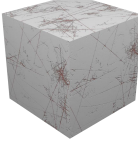
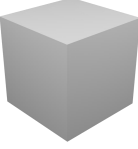
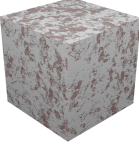


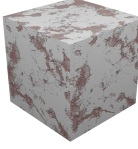
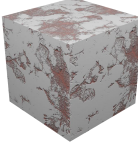
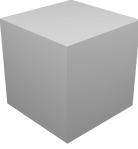

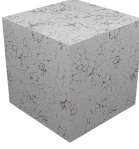
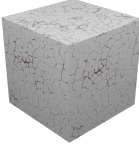
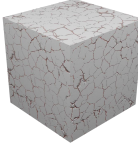
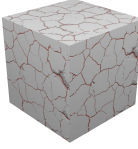
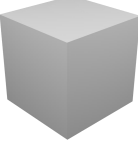

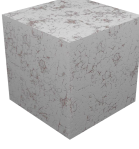
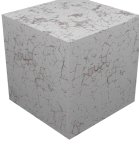
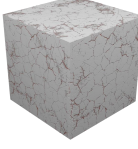
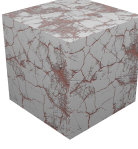
Parameter	0.00	0.20	0.40	0.60	0.80	1.00
Scratch Amount						
Grunge Amount						
Blunt Amount						
Scratch Map Scale						
Grunge Map Scale						
Blunt Map Scale						
Global Map Scale						

Figure 16: An overview of control parameters.

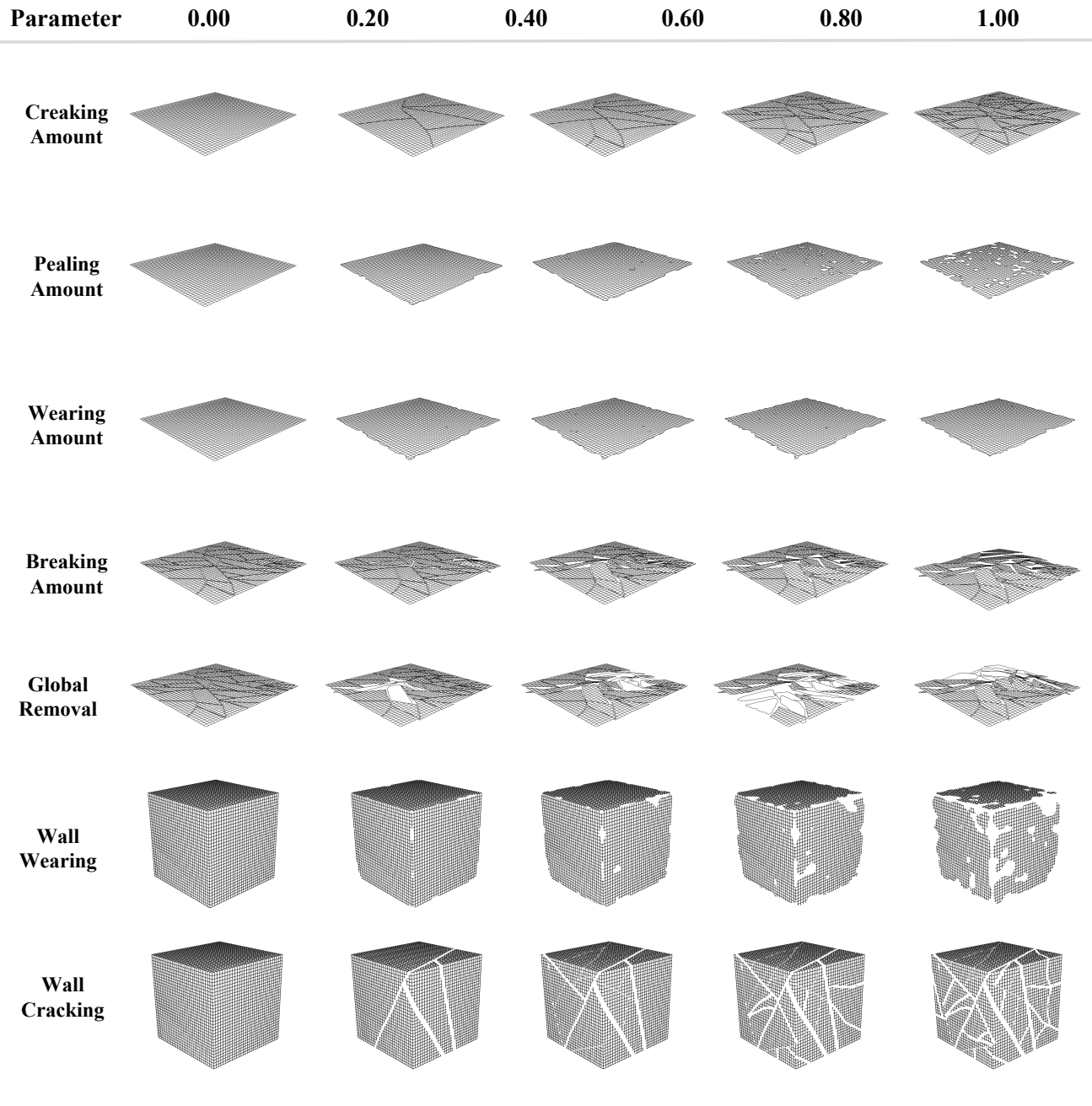


Figure 17: An overview of control parameters.

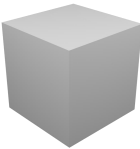


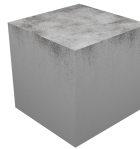


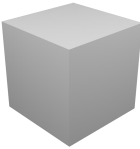

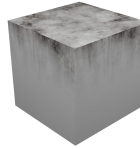
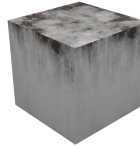
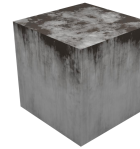
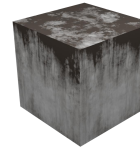
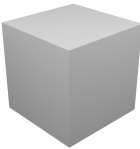

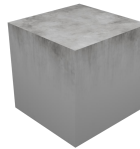
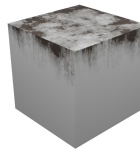
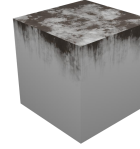
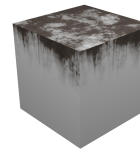
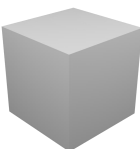
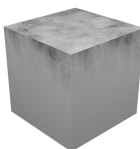
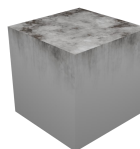
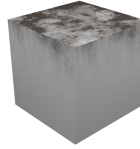
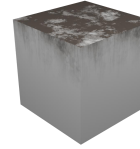
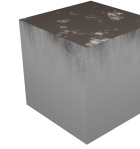
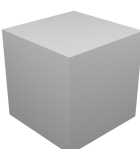
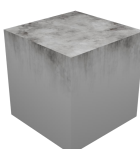
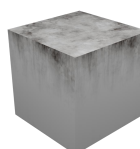
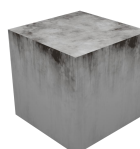

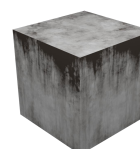
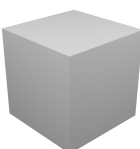
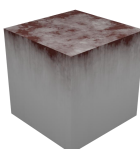
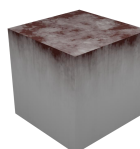
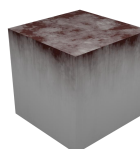
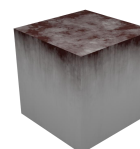
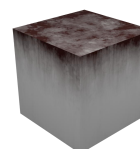
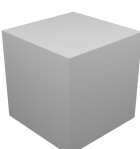
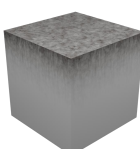
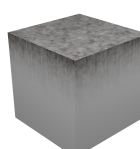
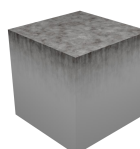
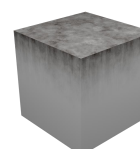
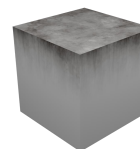
Parameter	0.00	0.20	0.40	0.60	0.80	1.00
Roughness						
Dirt Amount						
Color Contrast						
Center						
Sides						
Dirt Texture						
Bump Strength						

Figure 18: An overview of control parameters.

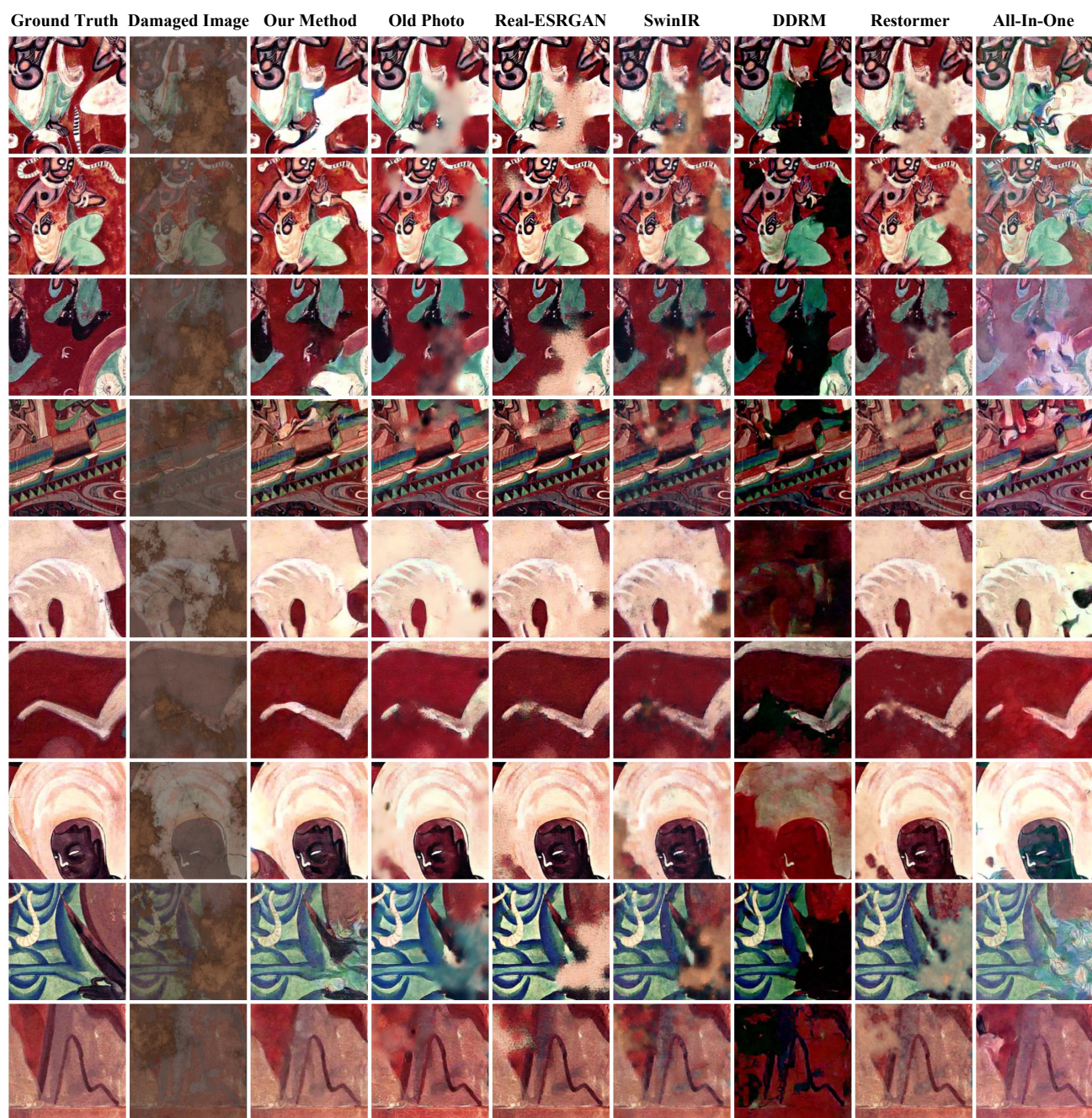


Figure 19: The qualitative results of mural-restoration experiments.

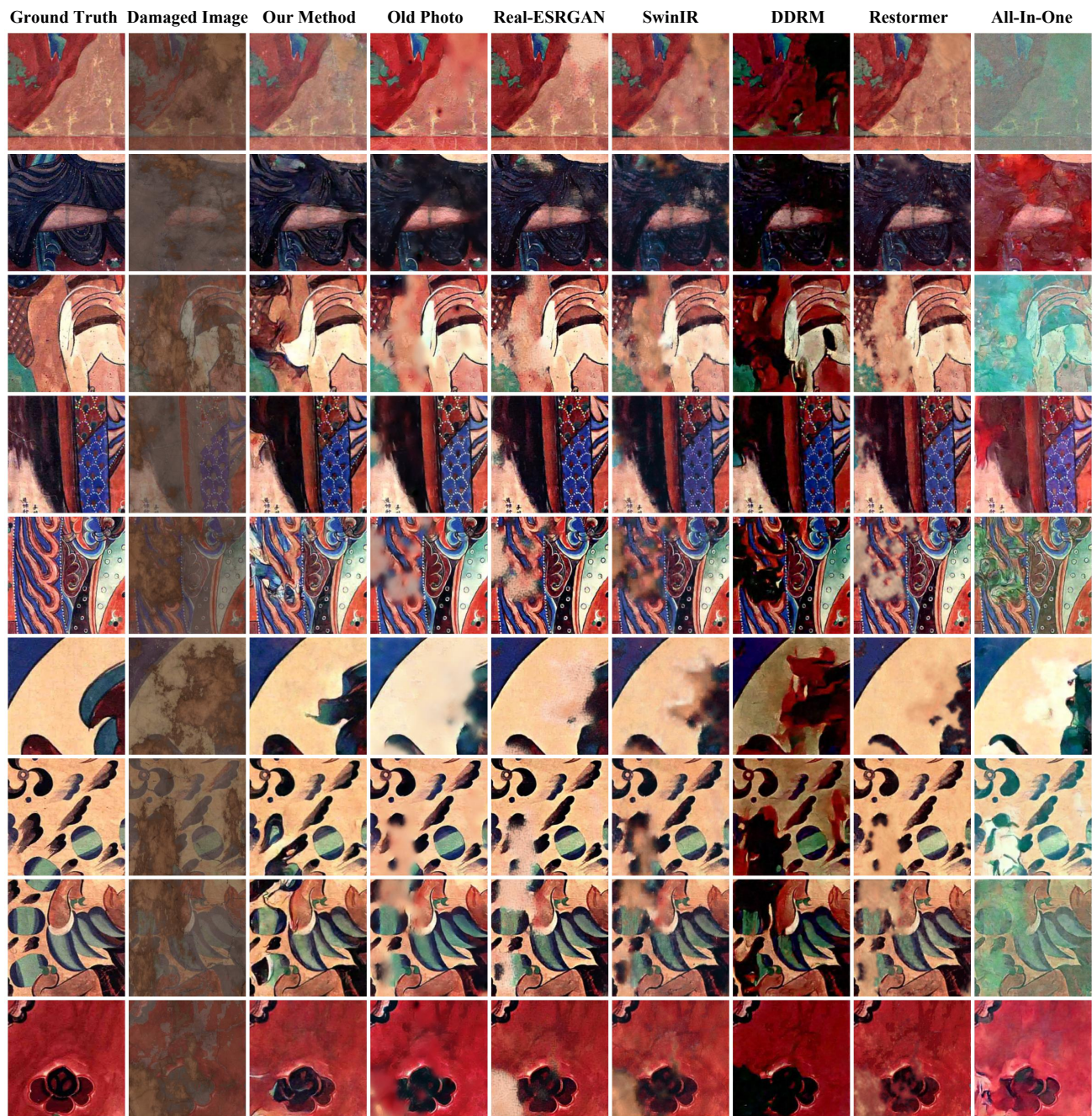


Figure 20: The qualitative results of mural-restoration experiments.

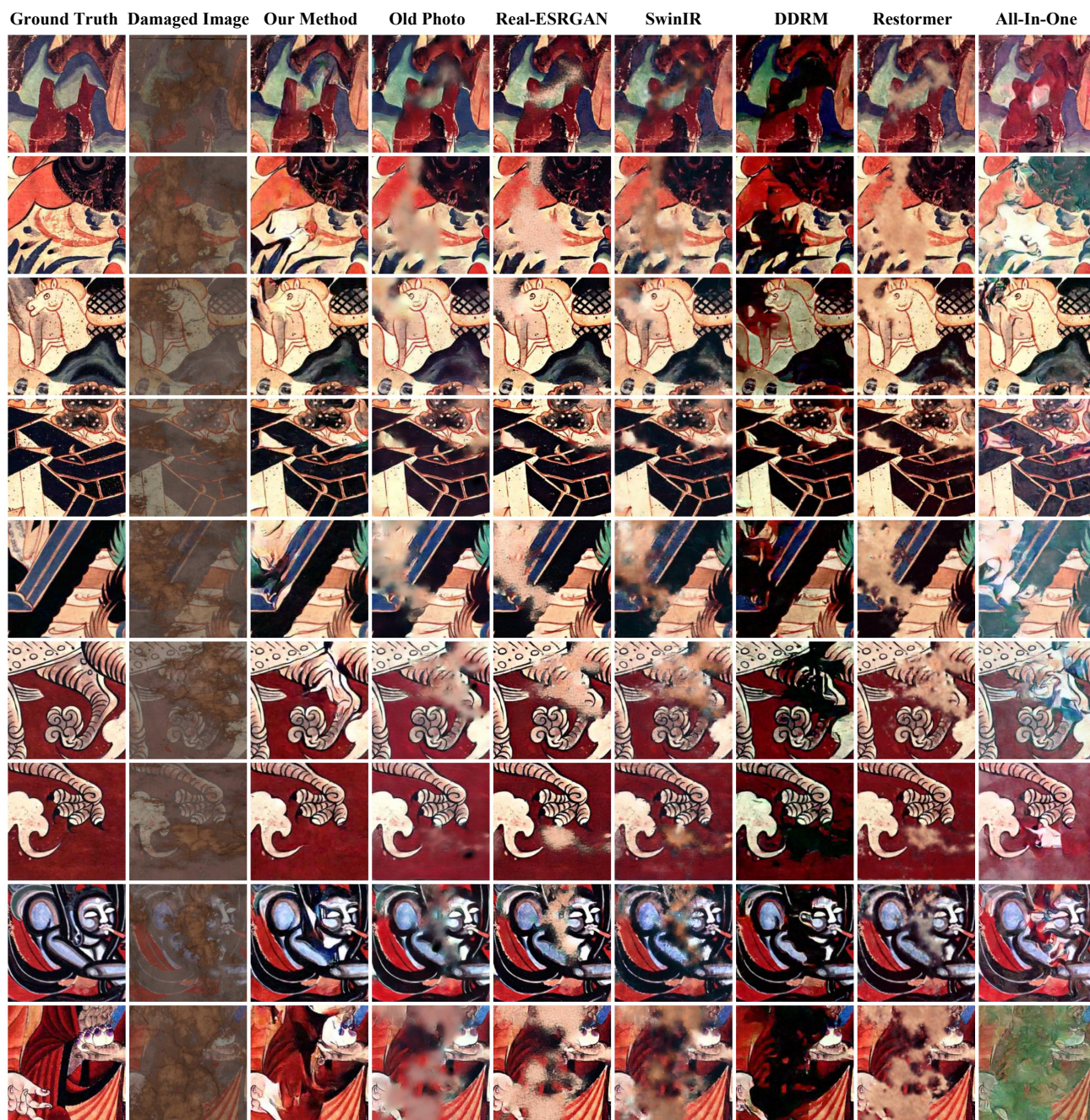


Figure 21: The qualitative results of mural-restoration experiments.