Dual-path Image Inpainting with Auxiliary GAN Inversion

Wentao Wang[1], Li Niu[1*], Jianfu Zhang[2], Xue Yang[1], Liqing Zhang[1*]
[1] Department of Computer Science and Engineering, MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
[2] Tensor Learning Team, RIKEN AIP
{wwt117,ustcnewly,yangxue-2019-sjtu,lqzhang}@sjtu.edu.cn, jianfu.zhang@riken.jp

Abstract

*Deep image inpainting can inpaint a corrupted image using a feed-forward inference, but still fails to handle large missing area or complex semantics. Recently, GAN inversion based inpainting methods propose to leverage semantic information in pretrained generator (*e.g.*, StyleGAN) to solve the above issues. Different from feed-forward methods, they seek for a closest latent code to the corrupted image and feed it to a pretrained generator. However, inferring the latent code is either time-consuming or inaccurate. In this paper, we develop a dual-path inpainting network with inversion path and feed-forward path, in which inversion path provides auxiliary information to help feed-forward path. We also design a novel deformable fusion module to align the feature maps in two paths. Experiments on FFHQ and LSUN demonstrate that our method is effective in solving the aforementioned problems while producing more realistic results than state-of-the-art methods.*

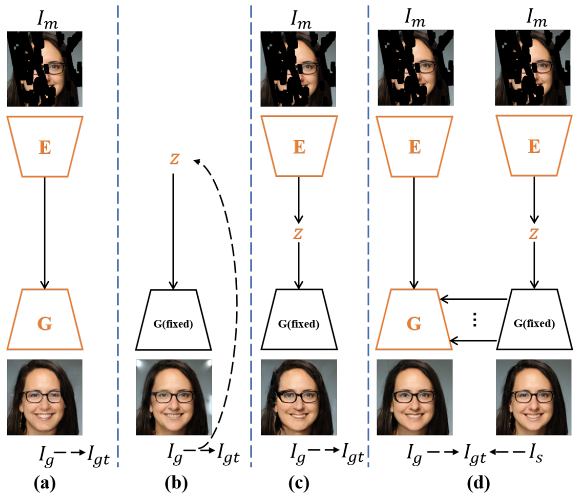
I_m
E
G
$I_g \dashrightarrow I_{gt}$
(a)
z
G(fixed)
$I_g \dashrightarrow I_{gt}$
(b)
I_m
E
z
G(fixed)
$I_g \dashrightarrow I_{gt}$
(c)
I_m
I_m
E
E
z
G
G(fixed)
$I_g \dashrightarrow I_{gt} \dashleftarrow I_s$
(d)

Figure 1. Illustration of four types of inpainting methods. (a) Feed-forward inpainting method. (b) Optimization-based GAN inversion inpainting method. [25] (c) Learning-based GAN inversion inpainting method. [12] (d) Ours. The modules marked in orange are optimized during training.

1. Introduction

Image inpainting [3] aims to fill the semantically appropriate and visually faithful contents in the missing regions of corrupted images. Numerous applications such as image editing, missing region repairing in image/video, and object removing benefit from the advance of image inpainting.

In recent years, deep neural networks have brought breakthroughs in image inpainting. Most deep inpainting methods [8, 16, 18, 20, 22, 24, 28–31] are single feed-forward methods. As shown in Figure 1 (a), they usually train a convolutional neural network with adversarial training and inpaint the corrupted image by feed-forward inference. Although great progress has been achieved in inpainting quality and speed compared with conventional methods, they still fail to process images with large missing areas or generate the results with good semantics [12,25]. As an alternative, GAN inversion [23] inpainting methods [12,25], which are rarely studied, have shown promising results for solving the above problems. They usually utilize a pretrained generative model (*e.g.*, StyleGAN [9, 10] and BigGAN [4]) to provide semantic prior. Differing from feed-forward inpainting methods, they first seek for the closest latent code in the latent space (input to the pretrained generator) for the corrupted image and then invert the latent code back to a complete image using the pretrained generator. There are two strategies to obtain the latent code: optimization-based strategy [25] and learning-based strategy [12]. For optimization-based strategy (see Figure 1(b)), optimization algorithm is adopted to iteratively update the latent code z which minimizes the reconstruction loss between generated image I_g and ground truth I_{gt}. For learning-based strategy (see Figure 1(c)), instead of directly optimizing the latent code z, an encoder network is used to infer the latent

[*]Corresponding author.

code $\boldsymbol{z}$ based on the masked image I_m. With extra semantic prior from the pretrained generative model, GAN inversion inpainting methods are not restricted to limited information when processing large missing scenarios. Nevertheless, two drawbacks of GAN inversion methods limit their applications. Specifically, optimization-based strategy [25] suffers from long inference time. Learning-based strategy [12] has mediocre restoration results because a single inference can hardly find the accurate latent code compared with multiple inferences in optimization-based strategy.

Considering the advantages and drawbacks of feed-forward inpainting methods and GAN inversion inpainting methods, we attempt to apply GAN inversion inpainting to assist feed-forward inpainting. As shown in Figure 1 (d) and detailed in Figure 2, we propose a hybrid inpainting framework which consists of two paths: feed-forward path and inversion path. In the inversion path, we search in the latent space to find a latent code whose inverted image has the closest distance to the observed region in the corrupted image. To maintain the efficiency of inference, we adopt learning-based strategy to infer the latent code using an encoder. To guarantee the inpainting quality, we do not directly use the results from the inversion path. Instead, we transfer useful knowledge from pretrained GAN in the inversion path to the feed-forward path. In the feed-forward path, we use auto-encoder network to inpaint the corrupted image, during which multi-scale features from the inversion path are integrated into the decoder. Besides, there may exist misalignment issue (shown in Figure 9) when fusing the features from two paths, so we propose a novel deformable fusion module in the generator to align the features from two paths. In this way, we alleviate the severe issues in inversion inpainting method and use its rich semantic knowledge to promote feed-forward inpainting method. Experiments on three datasets show that our method is superior to the state-of-the-art approaches. In summary, the main contributions of our work are summarized as follows,

- We propose a hybrid dual-path inpainting framework which assists in feed-forward inpainting with GAN inversion;
- We propose a novel deformable fusion module in the generator to solve the misalignment issue when fusing the features from two paths;
- Extensive experiments prove that our method can produce more semantically reasonable and high-fidelity results than other state-of-the-art methods.

2. Related Work

In this section, we will briefly introduce image inpainting methods and the applications of GAN inversion technique.

2.1. Image Inpainting

Recently, deep learning has brought remarkable progress in image inpainting. The mainstream deep image inpainting methods are based on single feed-forward inference. Pathak *et al*. [18] proposed context-encoder which first applies deep learning into image inpainting. Yu *et al*. [28] proposed a coarse-to-fine inpainting framework with contextual attention module. Designed for the irregular regions completion, partial convolution was proposed in [14] and gated convolution was proposed in [29]. Zheng *et al*. [33] realized diverse image inpainting by a VAE [11] based probabilistic network. Nazeri *et al*. [16] utilized edge as auxiliary information to improve image inpainting. Yi *et al*. [26] proposed an ultra-high-resolution image inpainting network with a contextual residual aggregated technique. Zhao *et al*. [32] proposed a conditional and unconditional modulated generative architectures for image inpainting. Although these feed-forward inpainting methods generate incredible results, they still fail to generate semantic results and most of them struggle with large missing scenarios.

GAN inversion inpainting methods [12, 25] with rare attention provide another perspective for solving the problems in feed-forward inpainting. Instead of directly inpainting the corrupted image, they search in the latent space of pretrained GAN to find a latent code to represent the corrupted image. Then the reconstructed image can be obtained by inverting the latent code back to the image through the pretrained GAN. Yeh *et al*. [25] proposed an optimization-based GAN inversion inpainting method that seeks the closest latent code by iterative inference. Even with high accuracy, it suffers from a long inference time due to multiple optimizations. To speed up the inference time, Lahiri *et al*. [12] proposed an learning-based GAN inversion inpainting method that adopts a learnable encoder to infer the latent code. Nevertheless, a single inference may not find the accurate latent code and generate color discrepancy results without post-processing. To solve the above problems, we propose a hybrid dual-path inpainting framework combining the advantages of GAN inversion inpainting and feed-forward inpainting.

2.2. GAN Inversion

GAN inversion aims to find the closest latent code based on a pretrained GAN model to recover the input image. It enables real image editing from the latent space. It can be mainly divided into two categories, learning-based and optimization-based. The learning-based strategy trains an encoder that can infer the latent code from a given image, and the optimization-based strategy directly optimizes the latent code by minimizing the pixel-wise reconstruction loss. Moreover, some works take advantage of both categories by combining these two ideas. They first use the encoder to infer a relatively accurate latent code and then

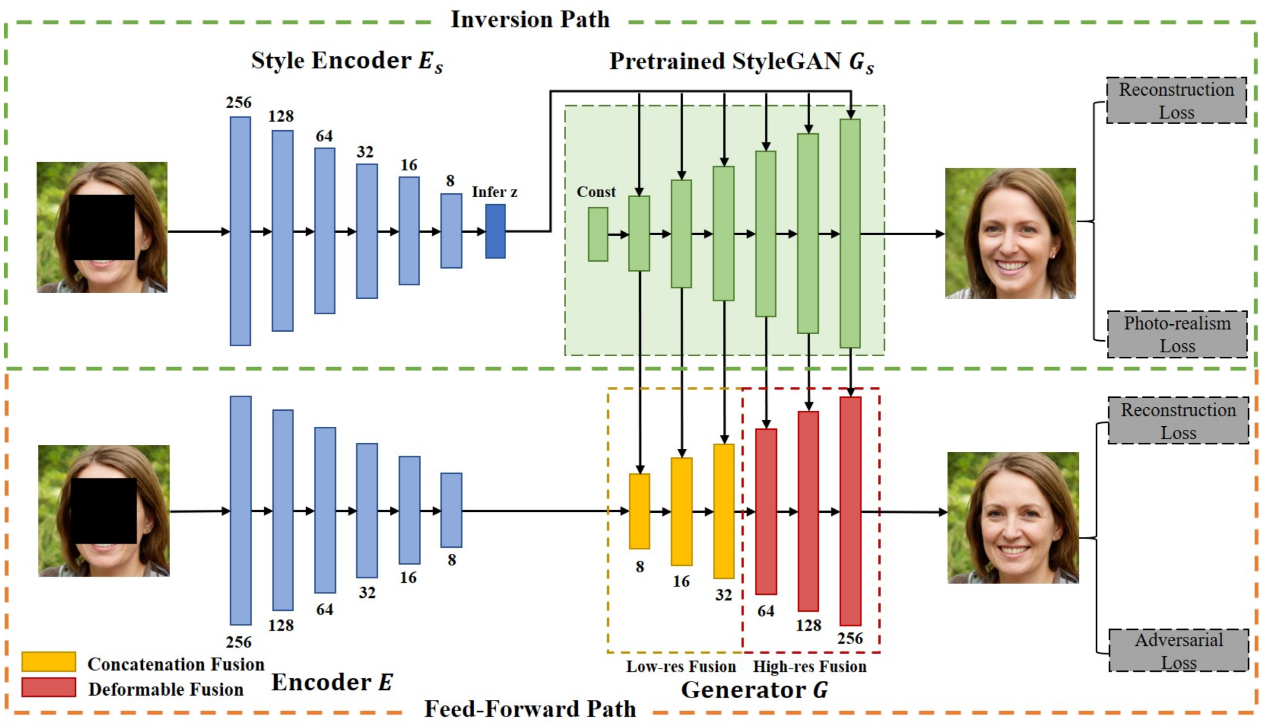

Figure 2. Hybrid Dual-path Inpainting Network. In inversion path, we find the close latent code of the corrupted image $\boldsymbol{I}_m$ and extract the corresponding semantic intermediate features from the pretrained GAN. In feed-forward path, we inpaint the corrupted image $\boldsymbol{I}_m$ to a complete image $\boldsymbol{I}_{out}$ with extra semantic prior from inversion path.

optimize it using the optimization-based strategy.

Apart from image inpainting [12,25], GAN inversion has also been applied to a wide range of computer vision tasks, like image editing [17,21,35], super-resolution [5,15], colorization [13], style transfer [1,2], *etc.* There are three reasons for integrating GAN inversion into image inpainting: 1) Since pretrained GAN model contains rich semantic prior [9,10,23], it can provide helpful knowledge especially required in the cases with the large missing areas; 2) If the images in the training set are highly diverse or with complex scenes, the feed-forward inpainting method may have difficulty in capturing the features of all images; 3) GAN inversion experts in exploiting the internal information of the image, which makes up the shortage of feed-forward inpainting. GAN inversion inpainting method is effective in producing semantic results and capable of powering our method to exploit more information from the corrupted image itself. By taking advantage of GAN inversion inpainting and feed-forward inpainting, our method can generate results with high fidelity and reasonable semantics.

3. Method

In this section, we will first present the overall network architecture. Then, we will discuss the misalignment issue when fusing the feature maps from two paths. Finally, we will describe the loss functions used in our network.

We denote the ground-truth complete image as $\boldsymbol{I}_{gt}$. $\boldsymbol{M}$ denotes the mask of corrupted image, which is a binary matrix with 0 indicating the missing region and 1 indicating the observed region. $\boldsymbol{I}_m = \boldsymbol{I}_{gt} \odot \boldsymbol{M}$ represents the masked corrupted image.

3.1. Hybrid Two-path Inpainting Network

The semantic issue in image inpainting results is important while most existing methods cannot address this issue very well. Recent studies have shown that GANs are able to effectively encode rich semantic information in the intermediate features and latent space [9, 10, 23]. By GAN inversion, we can obtain close semantic intermediate features to a corrupted image, which are used to compensate for the missing content in corrupted images. With this consideration, we propose a hybrid dual-path inpainting network which uses GAN inversion inpainting to enhance feed-forward inpainting. The overview of our network is shown in Figure 2, which contains two paths: inversion path and feed-forward path.

3.1.1 Inversion Path

In the inversion path, we aim to find the closest latent code of the corrupted image $\boldsymbol{I}_m$ and extract the corresponding semantic intermediate features $\boldsymbol{F}_s$ from the pretrained GAN. Then, the extracted semantic intermediate features $\boldsymbol{F}_s$ are incorporated into feed-forward path to provide extra semantic prior. The pretrained GAN model used in this work is StyleGAN [9, 10] due to its excellent generative performance.

Although optimization-based strategy used in [25] is able to find an accurate latent code, it suffers from intolerable inference time. To maintain the inference efficiency, we opt for learning-based strategy that uses an encoder E_s to infer the closest latent code $\boldsymbol{z}$ of corrupted image $\boldsymbol{I}_m$:

$$\boldsymbol{z} = E_s(\boldsymbol{I}_m). \tag{1}$$

Then the latent code $\boldsymbol{z}$ is sent into each layer of pretrained StyleGAN G_s:

$$\boldsymbol{F}_s, \boldsymbol{I}_s = G_s(\boldsymbol{z}), \tag{2}$$

where $\boldsymbol{F}_s = \{\boldsymbol{f}_i\}, i \in \{0, 1, .., N_s\}$ contains the multi-scale intermediate semantic features with N_s being the number of scales and $\boldsymbol{I}_s$ is the output of StyleGAN.

We do not directly use the output $\boldsymbol{I}_s$ from inversion path as the final inpainted result since the style encoder E_s may not find the closest latent code by a single inference. Thus, the results from inversion path may have location misalignment or color discrepancy issues without post-processing.

3.1.2 Feed-Forward Path

In feed-forward path, we utilize an auto-encoder architecture as our inpainting network, which contains an encoder E and a generator G. For the detailed architecture of encoder and generator, please refer to the generative path in [33].

The corrupted image $\boldsymbol{I}_m$ is fed into the encoder E to get the encoded feature $\boldsymbol{f}_m$:

$$\boldsymbol{f}_m = E(\boldsymbol{I}_m). \tag{3}$$

Then, the encoded feature $\boldsymbol{f}_m$ and semantic intermediate features $\boldsymbol{F}_s = \{\boldsymbol{f}_i\}, i \in \{0, 1, .., N_s\}$ from inversion path are both delivered to generator G to produce the inpainted result $\boldsymbol{I}_g$.

$$\boldsymbol{I}_g = G(\boldsymbol{f}_m, \boldsymbol{F}_s). \tag{4}$$

Specifically, we use inversion path to help feed-forward path by integrating the intermediate features from two paths. The most straightforward integration strategy is concatenating the feature maps with the same resolution from two paths. However, there may exist location misalignment issue if we directly fuse the features from two paths. One reason is that the inpainted result in the inversion path may be inaccurate with a single inference of latent code. Another reason is that one corrupted image has multiple possible inpainted results and the inpainted results from two paths could be different. In view of this, we propose a novel deformable feature fusion module in the generator to align the features from two paths. More details are illustrated in Section 3.2.

The final inpainted result $\boldsymbol{I}_{out}$ can be obtained by:

$$\boldsymbol{I}_{out} = \boldsymbol{I}_m + (1 - \boldsymbol{M}) \odot \boldsymbol{I}_g. \tag{5}$$

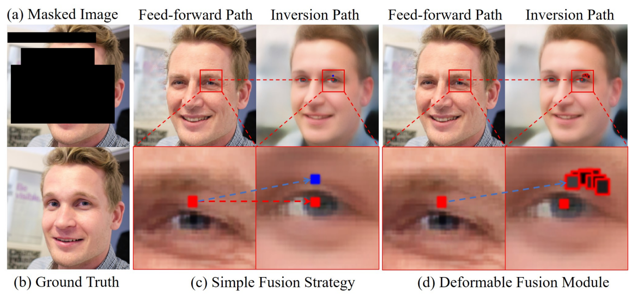

Figure 3. A comparison between simple fusion strategy and our deformable fusion module. In (c)(d), we draw the red points at the same location in two paths. In (c), simple fusion strategy (direct concatenation) will misalign the eyelid feature with eye feature and the blue point should be the ideal point to be aligned. In (d), our deformable fusion module can attend eyelid information correctly. We visualize the deformed sampling points with the corresponding modulations (marked with grey values) in the deformable convolution kernel centering at the red point.

3.2. Deformable Feature Fusion Module

Due to the inaccurate inference in the inversion path or the uncertainty of inpainted results, the results from the inversion path may be spatially misaligned with the feed-forward path results. This issue also exists in semantic intermediate features which are fed into generator G in the feed-forward path and would degrade the final inpainting results. This misalignment issue can not be well addressed by simply concatenating features from the inversion path and the feed-forward path.

Inspired by deformable convolution [6, 36], we propose a deformable fusion module to solve this issue. Similar to [6, 36], we visualize the outputs at the largest resolution (256×256) from two paths. In Figure 3, we compare simple fusion strategy with our proposed deformable fusion module. We select a point on the eyelid in the feed-forward path as the target point (red point). We also draw the target point (red point) at the same location in the inversion path. In Figure 3 (c), simple fusion strategy (Figure 4 (a)) directly concatenates the eyelid feature (red point) in the feed-forward path with misaligned eye feature (red point) in the inversion path. The blue point shows the ideal location to be concatenated. As shown in Figure 3 (d), our deformable fusion module (Figure 4 (b)) can attend correct eyelid information in the inversion path to match the eyelid in the feed-forward

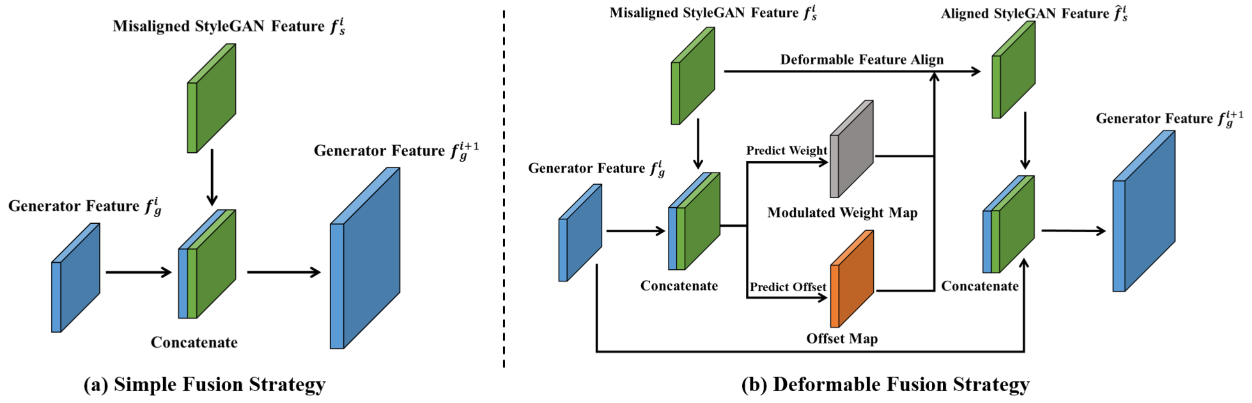
Misaligned StyleGAN Feature f_s^i
Generator Feature f_g^{i+1}
Generator Feature f_g^i
Concatenate
(a) Simple Fusion Strategy
Misaligned StyleGAN Feature f_s^i
Aligned StyleGAN Feature $\hat{f}_s^i$
Deformable Feature Align
Generator Feature f_g^{i+1}
Predict Weight
Generator Feature f_g^i
Modulated Weight Map
Predict Offset
Concatenate
Concatenate
Offset Map
(b) Deformable Fusion Strategy

Figure 4. Two fusion strategies. (a) Simple fusion strategy. (b) Our deformable fusion module. In (a), simply concatenate features from two path will cause misalignment. In (b), our deformable fusion module can align the inversion path feature and solve the misalignment.

path. In deformable fusion module, we predict the offsets of the target point in the inversion path, which can attend more similar information to the target point in the feed-forward path. In this way, the local features in the inversion path can be better aligned with those in the feed-forward path.

Specially, in low-resolution, the misalignment issue is not obvious because structures or semantic contents in low-resolution features are vague. The misaligned contents in the high-resolution feature map of two path are usually at the same location in the low-resolution feature map. Thus, considering the model complexity and inference efficiency, we simply concatenate the features $\boldsymbol{f}_s^i$ in the inversion path and the features $\boldsymbol{f}_g^i$ in the feed-forward path at low resolution:

$$\boldsymbol{f}_g^{i+1} = L^i(Concat(\boldsymbol{f}_s^i, \boldsymbol{f}_g^i)), \tag{6}$$

where $i = 0, 1, 2$ is the layer of the generator G corresponding to low-resolution feature $8\times8, 16\times16, 32\times32$. L^i is the i-th convolution block in generator. A detailed illustration is shown in Figure 4(a).

In high-resolution feature map, we replace simple fusion with deformable fusion, which is based on the modulated deformable convolution in DCNv2 [36]. As shown in Figure 4(b), we first concatenate the feature maps $\boldsymbol{f}_s^i$ and $\boldsymbol{f}_g^i$ from two paths, based on which the location offsets and modulation weights for each point in $\boldsymbol{f}_s^i$ are predicted. For each point, the location offsets indicate the sampling points in $\boldsymbol{f}_s^i$ which align with the point in $\boldsymbol{f}_g^i$ and the modulation weights imply the importance of different sampling points. The location offsets and modulation weights for all points form the location offset map $\boldsymbol{m}_o^i$ and modulation weight map $\boldsymbol{m}_w^i$ respectively. Then, $\boldsymbol{m}_o^i$, $\boldsymbol{m}_w^i$, and $\boldsymbol{f}_s^i$ are fed into modulated deformable convolution $DFConv$ to get the aligned feature map $\hat{\boldsymbol{f}_s}^i$. For the details of modulated deformable convolution, please refer to [36]. Finally, the aligned feature map $\hat{\boldsymbol{f}_s}^i$ is concatenated with $\boldsymbol{f}_g^i$ to predict the feature map for the next resolution:

$$\begin{aligned} \boldsymbol{m}_o^i, \boldsymbol{m}_w^i &= P^i(Concat(\boldsymbol{f}_s^i, \boldsymbol{f}_g^i)), \\ \hat{\boldsymbol{f}_s}^i &= DFConv(\boldsymbol{m}_o^i, \boldsymbol{m}_w^i, \boldsymbol{f}_s^i), \\ \boldsymbol{f}_g^{i+1} &= L^i(Concat(\hat{\boldsymbol{f}_s}', \boldsymbol{f}_g^i)), \end{aligned} \tag{7}$$

where $i = 3, 4, 5$ is the layer of the generator G corresponding to high-resolution feature $64\times64, 128\times128, 256\times256$. P^i is the convolution layer to predict location offset map $\boldsymbol{m}_o^i$ and modulation weight map $\boldsymbol{m}_w^i$.

3.3. Loss Function

In the inversion path, we follow [12] to use photo-realism loss $\mathcal{L}_{sp}$ and reconstruction loss $\mathcal{L}_{sr}$ to train style encoder E_s:

$$\mathcal{L}_{sp} = log(1 - D_s(G_s(\boldsymbol{z}))), \tag{8}$$

where D_s is the pretrained discriminator used in inversion path.

$$\mathcal{L}_{sr} = \|\boldsymbol{I}_{gt} - \boldsymbol{I}_s\|_1. \tag{9}$$

The overall loss function $\mathcal{L}_{inv}$ in the inversion path can be summarized as

$$\mathcal{L}_{inv} = \mathcal{L}_{sp} + \lambda_{sr}\mathcal{L}_{sr}, \tag{10}$$

where the hyper-parameter λ_{sr} is empirically set as 20.

In the feed-forward path, we use the reconstruction loss $\mathcal{L}_r$ for the final output $\boldsymbol{I}_g$ with the ground-truth image $\boldsymbol{I}_{gt}$:

$$\mathcal{L}_r = \|\boldsymbol{I}_{gt} - \boldsymbol{I}_g\|_1. \tag{11}$$

To encourage the final result $\boldsymbol{I}_g$ to be realistic, we add an adversarial loss [7] with a discriminator D. The adversarial loss for D is defined as:

$$\mathcal{L}_{adv}^D = -\mathbb{E}_{\boldsymbol{I}_{gt}}[logD(\boldsymbol{I}_{gt})] - \mathbb{E}_{\boldsymbol{I}_g}log[1 - D(\boldsymbol{I}_g)], \tag{12}$$

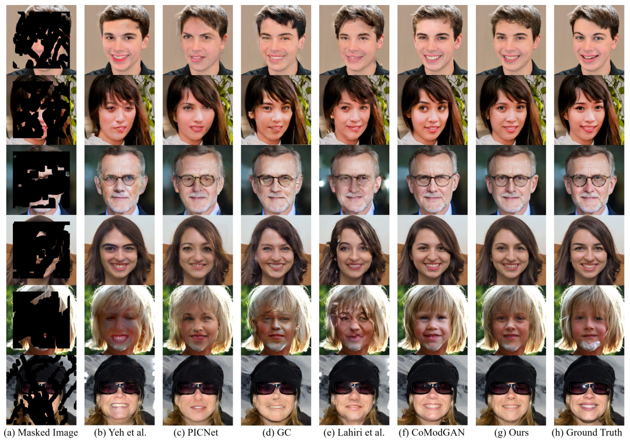

Figure 5. The visual comparison results on FFHQ [9]. Best viewed by zooming in.

while the adversarial for the generator is defined as

$$\mathcal{L}_{adv}^{G} = -\mathbb{E}_{\boldsymbol{I}_g}[logD(\boldsymbol{I}_g)]. \tag{13}$$

Then, the overall loss function $\mathcal{L}_{ff}$ in the feed-forward path can be summarized as

$$\mathcal{L}_{ff} = \lambda_r \mathcal{L}_r + \mathcal{L}_{adv}, \tag{14}$$

where the hyper-parameter λ_r is empirically set as 20. $\mathcal{L}_{adv}$ stands for $\mathcal{L}_{adv}^{G}$ (*resp.*, $\mathcal{L}_{adv}^{D}$) when optimizing the generator (*resp.*, discriminator).

4. Experiment

4.1. Datasets and Implementation Details

We implement our model using Pytorch 1.5.0 and the details of our model architecture are illustrated in the supplementary. We train the model on a single NVIDIA TITAN RTX GPU (24GB) with a batch size of 8, optimized by Adam optimizer with learning rate 0.0001, $\beta_1 = 0.001$, and $\beta_2 = 0.99$. Note that the pretrained StyleGAN model used in our network is StyleGAN2 [10]. Two datasets FFHQ [9] and LSUN [27] are utilized to validate our model since StyleGAN2 provides pretrained models for these two datasets. FFHQ dataset contains 70,000 face images and LSUN consists of 10 scene categories and 20 object categories. For LSUN dataset, we only show the results on the "church" and "cat" category due to limited computational resources. We follow StyleGAN2 [10] to pre-process the LSUN images, after which 100,000 church images and 200,000 cat images are used in training. For fair comparison with previous inpainting methods, all the images are resized to 256×256 with regular holes or irregular holes in random positions.

	Mask	Yeh *et al.* [25]	Lahiri *et al.* [12]	GC [29]	PICNet [33]	CoModGAN [32]	Ours
ℓ_1 (%)$^{\downarrow}$	0-10%	0.94	1.27	0.73	0.74	0.64	**0.62**
	10-20%	1.59	1.83	1.23	1.23	1.11	**1.06**
	20-30%	2.53	2.38	1.97	1.95	1.75	**1.41**
	30-40%	3.64	4.05	2.83	2.79	2.61	**2.16**
	40-50%	5.06	5.94	3.90	3.84	3.69	**3.21**
	50-60%	7.73	9.21	5.73	5.76	5.62	**4.59**
	Ave%	3.58	4.18	2.73	2.71	2.54	**2.17**
SSIM$^{\uparrow}$	0-10%	0.969	0.911	0.974	0.973	0.978	**0.979**
	10-20%	0.932	0.876	0.941	0.939	0.948	**0.951**
	20-30%	0.881	0.827	0.895	0.893	0.905	**0.914**
	30-40%	0.827	0.774	0.845	0.843	0.857	**0.879**
	40-50%	0.767	0.711	0.789	0.785	0.802	**0.827**
	50-60%	0.688	0.621	0.713	0.702	0.727	**0.743**
	Ave%	0.844	0.787	0.859	0.856	0.870	**0.882**
PSNR$^{\uparrow}$	0-10%	33.576	31.994	35.600	35.726	36.211	**36.342**
	10-20%	28.937	27.311	30.807	31.053	31.209	**31.607**
	20-30%	25.714	24.729	27.467	27.813	27.703	**28.365**
	30-40%	23.281	22.512	25.113	25.474	25.214	**26.251**
	40-50%	23.282	20.201	23.093	23.462	23.069	**24.155**
	50-60%	21.087	17.039	20.625	20.804	20.490	**21.751**
	Ave%	25.152	23.964	27.117	27.388	27.366	**28.078**
FID$^{\downarrow}$	0-10%	1.83	1.97	1.50	1.57	1.31	**1.20**
	10-20%	3.33	3.85	2.40	2.71	2.14	**2.00**
	20-30%	5.42	6.70	3.93	4.55	3.86	**2.99**
	30-40%	7.92	10.01	6.25	6.90	5.56	**4.13**
	40-50%	11.04	14.42	9.69	10.64	6.25	**5.67**
	50-60%	13.89	21.73	15.91	16.71	9.08	**8.13**
	Ave%	7.24	9.77	6.61	7.18	4.75	**4.02**

Table 1. Quantitative results of different methods on FFHQ [9].

4.2. Quantitative Comparisons

To quantitatively evaluate the inpainted results, we compare our model with two GAN inversion inpainting methods: Yeh *et al.* [25], Lahiri *et al.* [12], and three feed-forward inpainting methods PICNet [33], GC [29], CoModGAN [32]. Note that the pretrained GAN model used in Yeh *et al.* [25] is DCGAN [19] and used in Lahiri *et al.* [12] is BigGAN [4]. It is unfair to directly compare with these methods since StyleGAN2 outperforms DCGAN and BigGAN. For fairness, we reimplement Yeh *et al.* [25], Lahiri *et al.* [12] based on StyleGAN2. Since CoModGAN is trained with an image size of 512 or 1024, we resize the CoModGAN results to 256 for comparison. And CoModGAN is trained on FFHQ and Places2 [34], we omit the comparison with CoModGAN on LSUN dataset. PICNet is a diverse inpainting method, we set the sample number for each image to 20. The method used to test its metrics is the same as [33]. In FFHQ, the first 75,000 images are used as the training set and the last 5,000 images are used as the test set. For FFHQ test set, we use irregular masks provided by [14], in which the masks are split into several groups according to the relative masked area ratio: 0$\sim$10%, 10%$\sim$20%, 20%$\sim$30%, 30%$\sim$40%, 40%$\sim$50%, 50%$\sim$60%. We adopt the following four evaluation metrics: relative l_1, Structural Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Frechet Inception Distance (FID). The evaluation results on FFHQ are shown in Table 1. Note that since the code of Lahiri *et al.* [12] is not released, there may exist difference between the reimplemented results shown in Table 1 with original paper. PICNet is a diverse inpainting method, the way to test its metrics is the same as the original paper. It can be seen that our method outperforms all existing methods, especially in

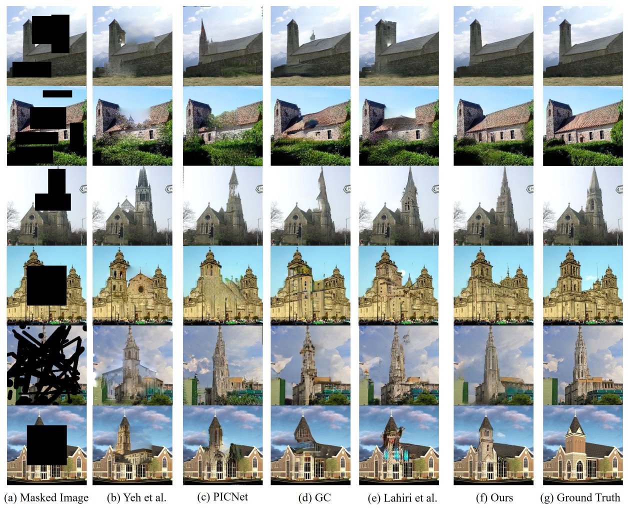

Figure 6. The visual comparison results on LSUN church [27]. Best viewed by zooming in.

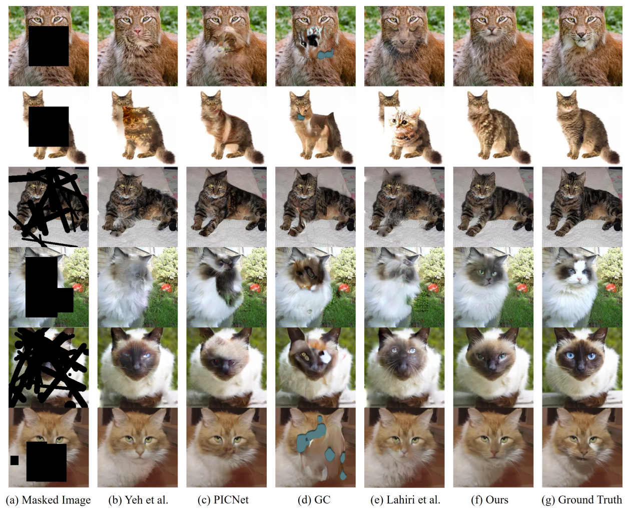

Figure 7. The visual comparison results on LSUN cat [27]. Best viewed by zooming in.

large missing area settings. Because PICNet picks each inpainted result based on the best metrics among 20 diverse results, it also achieves competitive quantitative results. Besides, extra comparisons conducted on the LSUN church are placed in the supplementary due to limited space.

4.3. Qualitative Comparisons

To qualitatively evaluate the inpainted results, we compare our model with other methods for both regular and irregular masks. In Figure 5, Figure 6, Figure 7, we provide visualization comparisons on FFHQ, LSUN church, LSUN cat, respectively. Due to space limitations, please zoom in to check the details and more comparisons are provided in the supplementary. In Figure 5, to prove the effectiveness of our methods under large missing scenarios, the test images are masked with 50%∼60% masks. It can be seen that the results produced by Yeh *et al.*, Lahiri *et al.*, PICNet, and GC contain distorted contents, artifacts or color discrepancies. The performance of CoModGAN is similar to ours, but it often generates inharmonious content with unmasked areas. (*e.g.*, row 1 and 5 in Figure 5 (f)). In Figure 6 and Figure 7, we can see that feed-forward inpainting methods (PICNet, GC) are good at inpainting irregular masks. However, GAN inversion inpainting methods (Yeh *et al.*, Lahiri *et al.*) produce more semantic results than feed-forward inpainting methods in regular mask setting. Because the image in LSUN cat and LSUN church dataset are highly diverse, despite containing the same type of objects, it is hard for feed-forward inpainting method to learn accurate features for each image. On the contrary, GAN inversion inpainting methods expert in exploiting information internally from each image and can generate more semantic results with the help of pre-trained model. Taking advantage of both methods, our method is good at dealing with different mask settings, producing more realistic and reasonable results.

Settings	ℓ_1 (%)$^\downarrow$	SSIM$^\uparrow$	PSNR$^\uparrow$	FID$^\downarrow$
w/o FF-Path	5.72	0.773	23.366	10.22
w/o Inv-Path	2.85	0.814	26.998	7.85
w/o DF	2.29	0.870	27.818	4.73
Ours	**2.17**	**0.882**	**28.078**	**4.02**

Table 2. Ablation studies for each module: (a) w/o inversion path (“w/o Inv-Path”); (b) w/o feed-forward path (“w/o FF-Path”); (c) w/o deformable fusion module (“w/o DF”); (d) Our method (“Ours”).

5. Ablation Study

In this section, we conduct ablation studies for our proposed hybrid dual-path inpainting network.

5.1. Effectiveness of Each Module

We investigate the effectiveness of our proposed modules by ablating each module: (a) w/o inversion path (“w/o Inv-Path”); (b) w/o feed-forward path (“w/o FF-Path”); (c) w/o deformable fusion module (“w/o DF”); (d) Our method (“Ours”). Note that in setting (c), we use concatenation to fuse the features from two paths. The results are shown in Table 2. All the results are tested on FFHQ datasets, and the test setting is the same as Section 4.2. Here only the averaged results over six mask groups are shown in Table 1. In these experiments, we find our inversion path (“w/o FF-Path”) sometimes generates blurry results because a single feed-forward inference can not find accurate latent code. This causes the “w/o FF-Path” setting to generate poor results in Table 2. By integrating the inversion path into the feed-forward path, our method outperforms other methods as shown in Table 1. Our proposed deformable fusion mod-

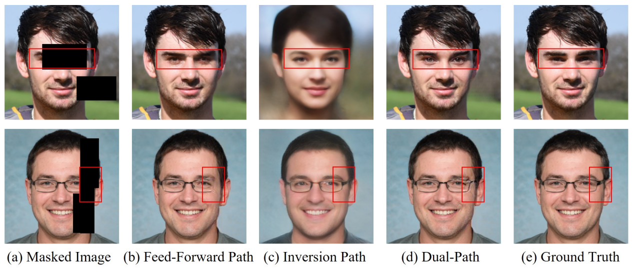

Figure 8. Visualization of Two Path Output. The differences are highlighted in red boxes. Best viewed by zooming in.

ule solves the misalignment issue, which further promotes the results.

5.2. Visualization of Two Path Output

To further explore the effectiveness of our proposed dual-path inpainting network, we give a comparison between inversion path and feed-forward path. The inversion path network and feed-forward path network are trained separately. Figure 8 (b), (c), (d) are inversion path outputs, feed-forward path outputs, and whole network output (dual path), respectively. In the first row, although the output of the inversion path is not accurate enough, we still obtain a realistic result. And the result of the inversion path has influence on the final results that the man in Figure 8 (d) has bigger eyes than results in Figure 8 (b). In the second row, without the inversion path, the feed-forward path can not restore the eyeglasses shown in Figure 8 (b). On the contrary, our method restores the eyeglasses successfully with the aid of the inversion path.

5.3. Effectiveness of Deformable Fusion Module

We investigate the effectiveness of the proposed deformable fusion module. In Figure 9 (b), it can be seen the eyebrow in column 1 and nose in column 2 are misaligned using simple fusion strategy. Using deformable fusion module, we obtain coherent eyebrow and nose as shown in Figure 4(b). It proves the effectiveness of our proposed deformable feature fusion module.

6. Discussion and Limitation

Our method is able to generate more realistic and semantic results, with the ability to process large missing scenarios. It can also be used as a plug-in module that incorporates the inversion path network into other inpainting methods with slight modification. However, like other GAN inversion methods, it needs a pre-trained generative model. It fails to work on a new dataset without training a generative model in advance. Moreover, as mentioned in Section 1, although the learning-based GAN inversion used in our method has a high inference speed, it sometimes can not find the accurate latent code. The inaccurate latent code

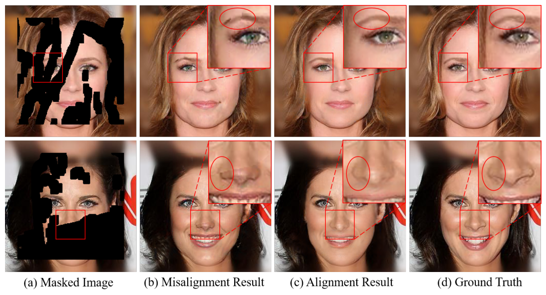

Figure 9. Misalignment issue when fuse features from two paths. The local magnification is in red box for better view. Best viewed by zooming in.

Settings	ℓ_1 (%)$^\downarrow$	SSIM$^\uparrow$	PSNR$^\uparrow$	FID$^\downarrow$	Speed (s/frame)$^\uparrow$
Optimization-based	2.02	0.884	28.191	3.82	45
Hybrid	2.00	0.889	28.135	3.83	45
Learning-based (Ours)	2.17	0.882	28.078	4.02	0.093

Table 3. Comparison of the influence between three GAN inversion strategies.

predicted by learning-based GAN inversion may influence the final inpainting result. Optimization-based GAN inversion and the hybrid strategy (combination of learning-based and optimization-based GAN inversion) could infer a more accurate latent code. We compare these three strategies to investigate their impact on the accuracy of the latent code. These three strategies are used to obtain the latent code in our inversion path, and the comparison is based on FFHQ dataset. In order to infer accurate latent code, optimization-based strategy and hybrid strategy all optimize 1,000 times. In Table 3, we can see that more accurate latent code further enhances our results. However, considering the slow inference speed, the promotion is not very significant. Taking accuracy and efficiency into account, it is wise to choose the learning-based GAN inversion in our inversion path.

7. Conclusion

In this paper, we propose a dual-path inpainting network with inversion path and feed-forward path. We use GAN inversion providing extra semantic information to assist feed-forward image inpainting. To solve the misalignment issue of fusing two path features, we also design a novel deformable fusion module. Experiments on FFHQ and LSUN demonstrate that our method is effective in producing more realistic and semantic results than state-of-the-art methods.

Acknowledgement

The work was supported by the National Science Foundation of China (62076162), and the Shanghai Municipal Science and Technology Major Project, China (2021SHZDZX0102, 20511100300). We thank Jiangtong Li for his valuable advice.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 3

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000. 1

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1, 6

[5] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14245–14254, 2021. 3

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5

[8] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):1–14, 2017. 1

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 3, 4, 6

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 3, 4, 6

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[12] Avisek Lahiri, Arnav Kumar Jain, Sanskar Agrawal, Pabitra Mitra, and Prabir Kumar Biswas. Prior guided gan based semantic inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13696–13705, 2020. 1, 2, 3, 5, 6

[13] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 3

[14] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision*, pages 85–100, 2018. 2, 6

[15] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020. 3

[16] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 1, 2

[17] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[18] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 1, 2

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[20] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 181–190, 2019. 1

[21] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 3

[22] Wentao Wang, Jianfu Zhang, Li Niu, Haoyu Ling, Xue Yang, and Liqing Zhang. Parallel multi-resolution fusion network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14559–14568, 2021. 1

[23] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *arXiv preprint arXiv:2101.05278*, 2021. 1, 3

[24] Jie Yang, Zhiquan Qi, and Yong Shi. Learning to incorporate structure knowledge for image inpainting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12605–12612, 2020. 1

[25] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017. 1, 2, 3, 4, 6

[26] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 2

[27] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6, 7

[28] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 1, 2

[29] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 1, 2, 6

[30] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *Proceedings of the European Conference on Computer Vision*, pages 1–17, 2020. 1

[31] Jianfu Zhang, Peiming Yang, Wentao Wang, Yan Hong, and Liqing Zhang. Image editing via segmentation guided self-attention network. *IEEE Signal Processing Letters*, 27:1605–1609, 2020. 1

[32] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 2, 6

[33] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. 2, 4, 6

[34] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 6

[35] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 3

[36] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 4, 5