# Learning High-Fidelity Face Texture Completion without Complete Face Texture

Jongyoo Kim, Jiaolong Yang, Xin Tong
Microsoft Research Asia
{jongk,jiaoyan,xtong}@microsoft.com

(a) Input (b) Inferred texture (c) Zoom-in of (b) (d) Rendered image (e) Zoom-in of (d)

Figure 1: Examples of single-image face texture completion results. Our neural network, trained in an unsupervised manner without any complete face texture, can generate high-fidelity results with natural skin details and facial hair.

## Abstract

*For face texture completion, previous methods typically use some complete textures captured by multiview imaging systems or 3D scanners for supervised learning. This paper deals with a new challenging problem – learning to complete invisible texture in a single face image without using any complete texture. We simply leverage a large corpus of face images of different subjects (e. g., FFHQ) to train a texture completion model in an unsupervised manner. To achieve this, we propose DSD-GAN, a novel deep neural network based method that applies two discriminators in UV map space and image space. These two discriminators work in a complementary manner to learn both facial structures and texture details. We show that their combination is essential to obtain high-fidelity results. Despite the network never sees any complete facial appearance, it is able to generate compelling full textures from single images.*

## 1. Introduction

Human face analysis and digitization are among the most popular topics in computer vision and graphics. Most face photos we take do not represent a full view of a face due to occlusion by the face itself or other objects. In fact, self-occlusion is ubiquitous for face images in the wild as shown by the examples in Fig. 1, leading to invisible texture content. The task of face texture completion is to infer the invisible face content and recover full-face appearance. It has a wide variety of applications ranging from 3D avatar creation [14, 20], 3D morphable model construction [2, 31], and face image manipulation [13, 36], to high-level vision tasks such as face recognition [13, 6]. This work is devoted to texture[1] completion learning using deep neural networks for single face images.

However, learning face texture completion is not

[1]Following [6], we use texture to refer to a facial appearance on an image, not an albedo or intrinsic image.

straightforward due to the difficulty of collecting labeled training data. For single images, obtaining complete texture by manually labeling or painting is not a viable option. Using multi-view images to obtain high-resolution and high-quality textures is also not a trivial task, which necessitates sophisticated face image capture and processing pipelines. Previous works often use some special devices placed in controlled environments for training data capture, such as a multi-view DSLR capturing system or a 3D scanning system [4, 37, 6, 20]. Most datasets from these works are not publicly available. Note that most of them capture face albedo which is different from the texture we consider in this paper, but the requirements for face scanning, registration, and stitching are similar for obtaining both.

To step aside from the effort for labeled data collection, we propose using a large collection of high-resolution face photos captured in unconstrained settings to train a face texture completion model. Although this would eliminate the need for complete texture acquisition, it poses new challenges for the learning task, since, for each input face image, there's no image of the same person that can be used for supervision.

To this end, we propose a novel generative adversarial learning method called Dual-Space-Discriminator Generative Adversarial Network (*DSD-GAN*) to learn face texture completion in an unsupervised fashion. To make full use of the partially visible face appearance, we apply two discriminators designed differently in the UV texture space and image space. Our key observation is that the former is more suitable to learn texture details, whereas the latter is more important to learn facial structure.

In the UV texture space, a discriminator takes small local patches as input to focus on detailed textural patterns. Real and fake patches are obtained on one image according to their visibility. Since the discriminator focuses on local patches, it may get stuck into local minima and ignore global textural consistency. Therefore, we make the local discriminator conditioned on the patch coordinates and train it to regress these coordinates. On the other hand, we employ a differentiable mesh renderer [10] to convert the generated UV texture into the image space and apply a discriminator taking the full image as input. The goal here is to capture the general face structure, core semantic components, and overall color gradient caused by different illumination conditions. The raw face photos are used as real samples, which naturally provide reliable training signals. To make the discriminator focus more on missing regions and avoid confusing it when missing regions are small, we apply spatially-varying labels for more effective training. We show that our dual space discriminators are very effective for the unsupervised learning task and the trained generator can produce high-fidelity face texture completion results.

## 2. Related Work

**Deep image completion.** Image completion with a deep neural network has been actively studied in the past few years. Pathak *et al.* proposed a context-encoder architecture, which employs adversarial and pixel-wise reconstruction losses [29]. To efficiently enlarge receptive field, Iizuka *et al*. [15] deployed a dilated convolution for image completion. Later, to deal with irregular hole shapes, a partial convolutional (PCONV) was proposed by Liu *et al*. [25] where a mask is used to consider only valid pixels and valid regions are gradually grown through a network.

On the other hand, many methods adopt both a local and global discriminator to enhance image quality [15, 41, 6]. The local discriminator is applied for some fixed regions like facial center [6] or regions with generated content [15, 41]. Compared to these methods, our discriminators are designed in two different spaces to learn texture completion without any complete texture.

**3D face geometry and texture.** Face UV texture is closely related to the 3D geometry of human faces. The 3D Morphable Model (3DMM) [2, 30] has been one of the most popular tool for 3D face reconstruction . The model is generally built based on a set of real captured data, then PCA is applied to form a parametric model. Many works employed the analysis-by-synthesis approach for 3D reconstruction. Recently, deep neural networks are employed for 3D face reconstruction using synthetic data for training [8, 12, 32] or trained in a weakly-supervised way [7, 10, 34].

Parametric facial texture models were also proposed in the past [3, 9]. Booth *et al*. [3] used a robust principal component analysis to build a model from incomplete "in-the-wild" textures. This method has an implicit texture completion during the construction of a texture model but loses fine details. In [9], a GAN-based texture model is proposed where 10,000 high-resolution complete UV data were used. However, for texture inference, the generated results are limited to their parametric model space. Contrarily, our approach generates a complete texture while preserving the visible pixels and details from the input. Lin *et al*. [24] proposed to generate fine-detail-textures based on a reconstructed texture using a parametric model. Their model only focuses on the visible part and does not handle the naturalness of the self-occluded textures.

Similar to our approach, there have been a few methods proposed for texture completion based on 3D geometry [33, 37, 6]. Saito *et al*. [33] proposed a DNN to extract features from the partial images and used multi-scale detail analysis. Though it could generate plausible results, it requires heavy computation resources. Yamaguchi *et al*. [37] employed GAN for albedo prediction, face texture completion, and super-resolution of face texture. Deng et al.[6] proposed a GAN-based UV texture completion using hy-

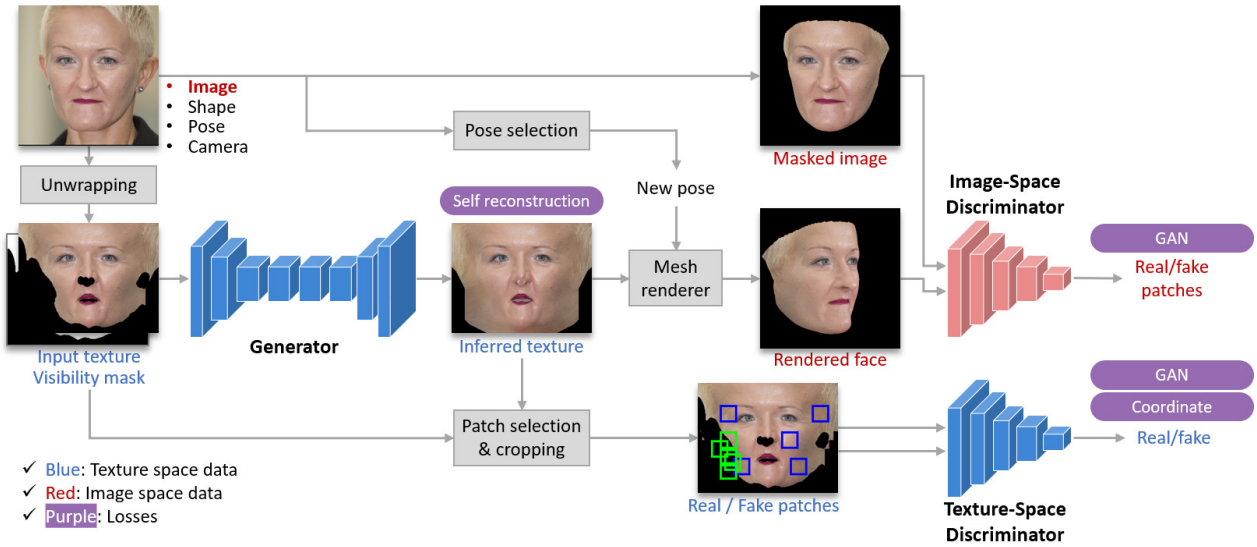



Figure 2: Overview of the proposed framework. The image-space discriminator is trained to learn coarse structure while the texture-space discriminator learns detailed textures.

brid discriminators. However, both methods heavily depend on high-quality and high-resolution training data.

On the other hand, SG-NN proposed a self-supervised completion model for 3D structure [5] without complete data. However, while SG-NN utilizes direct supervision by imposing artificial masks, DSD-GAN learns the distribution of the partial textural information using discriminators.

## 3. Method

We have two key goals: (1) learning face texture completion without complete texture data and (2) generating high-resolution and high-quality textures. To achieve both, we carefully designed our framework to fully make use of visible pixels in the training data. The overview of the framework is demonstrated in Fig. 2. A generator infers complete texture from an incomplete input texture and its visibility mask. Then, the texture is fed into two discriminators working in two different spaces: UV-atlas texture space and rendered image space. Since there is no full ground-truth in texture space, we crop the inferred textures to form several real and fake patches, which are fed into the texture-space discriminator. In the case of image space, the original image itself is regarded as real data while the rendered image is fake data as the inputs to the discriminator. The detailed processes are described in the following sections.

### 3.1. Data Preparation

In this work, we leverage a large collection of face images for the texture completion training task. We use two face image datasets: Flickr-Faces-HQ (FFHQ) [18] and CelebA-HQ [17]. FFHQ consists of 70K high-resolution face images and CelebA-HQ has about 30K.

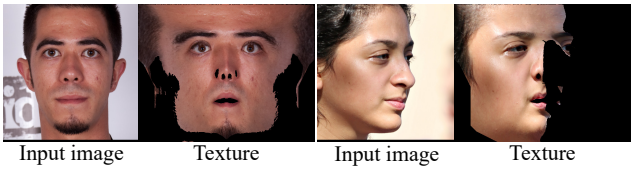



Figure 3: Samples in the generated training dataset.

To obtain UV textures and generate our training data, we developed an automated pipeline which consists of three steps: (1) DNN-based 3D face reconstruction, (2) optimization-based refinement, and (3) data cleaning. For the first step, we employ an off-the-shelf algorithm from [7] to obtain the initial 3D reconstruction results. To further improve the alignment accuracy, we then apply an offline optimization with losses similar to [7]. With the recovered 3D geometry, face textures and visibility maps cab be extracted. Finally, we refine the textures to remove bad samples: we first simplify the visibility mask boundary by using simple morphological operations, and then remove background (non-face) pixels introduced in texture space using the face parsing information [23]. If there are too many background pixels in a texture, we simply discard it.

Some example textures generated from our pipeline are presented in Fig. 3. In our experiment, the resolutions of UV texture data and image space data are $512\times512$ and $448\times448$, respectively.

### 3.2. Generator

Motivated from [35], our generator consists of downsampling, residual blocks, and upsampling layers. Between each residual block, we add dilated convolutions [39] to enlarge the receptive field, which we found to be important

especially for high-resolution images. For the input of the generator, we first concatenate a texture with a Gaussian random noise image where noise appears only in the hole regions of the texture map, then we flip the data horizontally and concatenate it with the original one to impose a weak symmetric consistency as used in previous studies [6, 37].

We denote the input texture as $T_{\text{inc}}$, visibility mask as $M_{\text{tex}}$, and the inferred texture as $T_{\text{pred}}$. For $M_{\text{tex}}$, valid and hole regions are indicated with 1 and 0 respectively. The output of the generator is used to calculate the self-reconstruction loss for the valid pixels as

$$\mathcal{L}_{\text{rec}} = \frac{1}{\sum_{(i,j)} M'_{\text{tex}}} \sum_{(i,j)} |(T_{\text{inc}} - T_{\text{pred}}) \odot M'_{\text{tex}}| \quad (1)$$

where $M'_{\text{tex}}$ is the inverse of $M_{\text{tex}}$ and $\odot$ is a Hadamard product. Detailed network architectures are explained in the *suppl. material*.

### 3.3. Discriminators

**Texture-space discriminator.** From the inferred face texture $T_{\text{pred}}$, we extract local patches to represent real and fake labels for the discriminator. To make the selection process simpler and faster, we first define discrete candidate locations with the stride of $stride_c$ and cropping size $width_c$. Then, using each visibility mask, we calculate the ratio of the number of valid pixels over that of total pixels in each cropping region and classify them into three categories: a hole patch (ratio$< 0.65$), a valid patch (ratio$> 0.9$), and the rest. Among them, only the hole patches $C_{\text{hol}}$ and valid patches $C_{\text{val}}$ are used for training. The optimal cropping size, $width_c$ is chosen based on experiments. If $width_c$ is too small, the model is less likely to catch important textural patterns, while large cropping results in fewer possible candidates. In the experiment, we set $stride_c = width_I/32$ and $width_c = stride_c \times 2$.

To calculate the adversarial loss, we deploy a least square GAN [26] as

$$\mathcal{L}_{\text{loc}} = \mathop{\mathbb{E}}_{C_{\text{val}}}[(D_T(C_{\text{val}}) - 1)^2] - \mathop{\mathbb{E}}_{C_{\text{hol}},z}[D_T(C_{\text{hol}})^2] \quad (2)$$

where $D_T(\cdot)$ indicates the texture-space discriminator, and $z$ is the random noise of the input.

Face texture has a canonical structure, and there is a strong relationship between texture patterns and locations. For example, beard exists only around the mouth and chin, while cheeks usually have homogeneous textures. Therefore, we consolidate the positional information of the patches into their textural information to generate semantically correct details. Specifically, we train a conditional discriminator by regressing the coordinates of the input patches in the original full texture. We add a subbranch at the end of the convolutional layers in the discriminator as a regressor like AC-GAN [22, 27]. The coordinate regression loss is calculated as

Figure 4: Illustration of local patch selection. Blue squares are real patches and green squares are fake patches.

$$\mathcal{L}_{\text{coord}} = \frac{1}{K}(\text{Reg}(C) - \text{pos}_C)^2 \quad (3)$$

where $\text{Reg}(C)$ indicates the predicted coordinate by the regressor, $\text{pos}_C$ is the normalized coordinate of $C$, and $K$ is the total number of patches. For the coordinate normalization, values are rescaled to range between $[0, 1]$.

**Image-space discriminator.** To learn the global structural information, we consider another discriminator defined in image space. Different from texture-space data, natural faces images in image space are faultless as themselves, that is, they can be regarded as real data for adversarial learning. Besides, image space data is less sensitive to alignment accuracy and it is easy to collect a large corpus of face images.

Using a differentiable mesh renderer [10], we render the inferred texture $T_{\text{pred}}$ to multiple face images $I_{\text{pred}}$. In this stage, choosing proper face poses to be rendered is important. We choose the new poses at random but ensure they are sufficiently far from the original face pose. For the real data, we simply apply the face boundary mask to the raw images, which we denote by $I_{\text{raw}}$.

Our image-space discriminator takes a whole image as input and outputs a 14×14 fake/real label predictions. The receptive field size for each output point is 286×286, which is large enough to capture the facial structural information. For the 14×14 predictions, we propose spatially-varying labels for the adversarial learning to deal with highly unbalanced real and fake pixels. Generally, in a rendered image $I_{\text{pred}}$, the number of hole pixels is smaller compared to the whole face region. The rest pixels are from the original valid face texture. If we naively handle these valid pixels as fake data, it could potentially confuse the discriminator. Therefore, different from the PatchGAN discriminator [16] where the whole patches in one image have a uniform label, our labels are spatially-varying. The individual labels are generated using the rendered visibility mask $M_{\text{img}} = \text{DR}(M_{\text{tex}}, \mathbf{p})$ where $\text{DR}(\cdot)$ is the differentiable mesh renderer, and $\mathbf{p}$ is the new pose. We resize $M_{\text{img}}$ to form the label map having the same size as the output of the image-space discriminator, and binarize it with the threshold of 0.9. More details can be found in the *suppl. material*.

The loss function for our image-space discriminator can

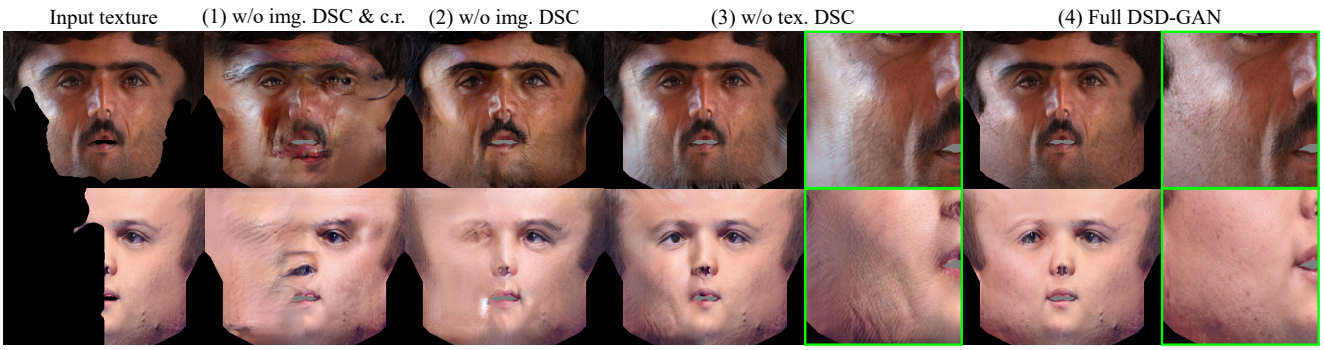
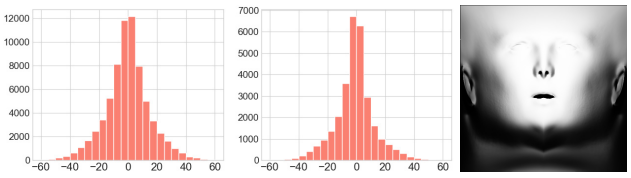



Figure 5: Comparison of completion results with different modules. (**Best viewed with zoom**)



(a) FFHQ (b) CelebA-HQ (c)

Figure 6: (a) and (b) are the yaw angle distributions of FFHQ and CelebA-HQ; and (c) is the average of visibility maps (darker means less number of visible pixels).

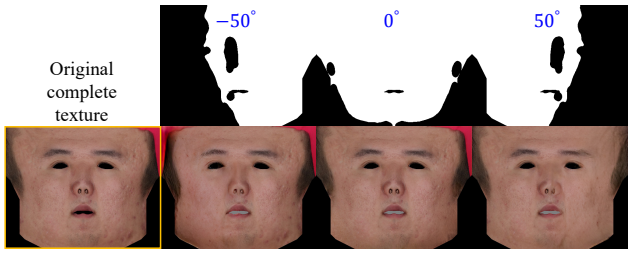



Figure 7: Pose invariance test. More visual results can be found in *suppl. material*.

be written as

$$\mathcal{L}_{\text{img}} = \mathop{\mathbb{E}}_{I_{\text{gt}}}[(D_I(I_{\text{gt}}) - \mathbf{1})^2] - \mathop{\mathbb{E}}_{I_{\text{pred}},z}[(D_I(I_{\text{pred}}) - \mathbf{l}_{\text{img}})^2] \quad (4)$$

$$I_{\text{pred}} = \text{DR}(T_{\text{pred}}, \mathbf{p}) \quad (5)$$

where $D_I(\cdot)$ is the image space discriminator, and $\mathbf{l}_{\text{img}}$ is the spacially-varying label.

### 3.4. Overcoming Data Bias

FFHQ and CelebA-HQ are highly biased to front faces as shown in Fig. 6 (a) and (b). To mitigate the bias issue, we applied several workarounds in data sampling.

**Sampling pose to render.** If we choose too far poses from its original, most of $I_{\text{pred}}$ become profile faces, while the majority of $I_{\text{raw}}$ is near-front faces. This can make the discriminator sensitive to pose rather than textural quality. Thus, the new pose was randomly sampled from the normal distribution with the estimated mean and covariance of FFHQ and CelebA-HQ. To guarantee the rendered face includes hole pixels, sampling was repeated until the new yaw meets two conditions: (1) $>20°$ difference from the original (2) opposite side based on front angle. This leads to slightly broader distribution then real, but the model less suffered from the bias issue. We will add details in the revision.

**Sampling local patches.** Due to the biased pose distribution, the distribution of valid/hole patches in the UV space along the vertical direction is highly biased to upper/lower regions, as depicted in Fig. 6 (c). Naive random patch sampling from the valid/hole patch pool will result in imbalanced training data. To avoid this issue, we first uniformly and independently sampling a row index, then uniformly sampled a column index. This way, the training path distribution along the vertical direction will be made uniform.

## 4. Experiments

**Implementation details.** Our method is implemented using Tensorflow [1]. We use Adam optimizer [19] and learning rate $1e{-}4$ to train all the models. We use four GPUs for training where the batch size of each GPU was 4. More details can be found in the *supp. material*.

### 4.1. Ablation Study

**Effect of each proposed module.** To validate the efficacy of each module, we conduct experiments with different submodules. We compare the four different settings: (1) DSD-GAN with only texture-space discriminator without coordinate regression, (2) DSD-GAN with only texture-space discriminator, (3) DSD-GAN with only image-space discriminator, and (1) full DSD-GAN. Each result is shown in Fig. 5. For models (1), (2), and (3), the self-reconstruction loss was used for training with a small weight.

As seen in Fig. 5, model (2) works well for near-front faces with natural textures. However, significant artifacts appear for a large-pose, especially when a large part of the core facial component is missing. The inferred texture has

| | SSIM | | | | | | Perceptual similarity | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | -60° | -30° | 0° | 30° | 60° | Avg. | -60° | -30° | 0° | 30° | 60° | Avg. |
| DeepFillv2 | 0.5974 | 0.6988 | 0.7696 | 0.6839 | 0.5759 | 0.6651 | 0.5783 | 0.6943 | 0.7232 | 0.6688 | 0.5445 | 0.6418 |
| PICNet | 0.6332 | 0.7264 | 0.7711 | 0.7168 | 0.6230 | 0.6941 | 0.6054 | 0.7482 | 0.7544 | 0.7320 | 0.5724 | 0.6825 |
| RFRNet | 0.6237 | 0.7120 | 0.7747 | 0.7034 | 0.6129 | 0.6853 | 0.5738 | 0.7064 | 0.7245 | 0.6892 | 0.5546 | 0.6497 |
| Base-MPIE | 0.6521 | 0.7153 | 0.7450 | 0.7108 | 0.6420 | 0.6930 | 0.6787 | 0.7575 | 0.7799 | 0.7483 | 0.6485 | 0.7226 |
| Base-BFM | 0.7115 | 0.7561 | 0.7769 | 0.7561 | 0.7164 | 0.7434 | 0.7170 | 0.7872 | 0.7934 | 0.7788 | 0.7180 | 0.7589 |
| w/o img. DSC. & c. r. | 0.6731 | 0.6780 | 0.6677 | 0.6823 | 0.6780 | 0.6758 | 0.6373 | 0.6078 | 0.5954 | 0.6603 | 0.6445 | 0.6290 |
| w/o img. DSC. | 0.6820 | 0.6881 | 0.6884 | 0.6913 | 0.6941 | 0.6888 | 0.6250 | 0.6712 | 0.6663 | 0.6822 | 0.6870 | 0.6663 |
| w/o tex. DSC. | 0.7323 | 0.7732 | 0.7873 | 0.7635 | 0.7347 | 0.7582 | 0.7277 | 0.7922 | 0.7987 | 0.7805 | 0.7419 | 0.7682 |
| **DSD-GAN** | 0.7531 | 0.7755 | 0.7876 | 0.7736 | 0.7522 | 0.7684 | 0.7500 | 0.7914 | 0.7985 | 0.7839 | 0.7594 | 0.7767 |

Table 1: Comparison of SSIM (↑) and perceptual similarity (↑) under different input poses on the FaceScape dataset. (red and blue indicate the best and second-best models.)

| DeepFillv2 | PICNet | RFRNet | Base-MPIE | Base-BFM | w/o img. dsc. & c. r. | w/o img. dsc. | w/o tex. dsc. | **DSD-GAN** |
|---|---|---|---|---|---|---|---|---|
| 24.3861 | 40.3863 | 44.9248 | 30.3546 | 17.2121 | 56.4223 | 43.0560 | 9.9678 | 3.3124 |

Table 2: Comparison of FID scores (↓) on the FaceScape dataset. (red and blue indicate the best and second-best models.)

high-frequency details but they are not consistent with valid regions. For model (4) where the coordinate regression is additionally excluded, the overall facial structure is collapsed. On the other hand, the image-space discriminator (model (3)) can capture the overall structure well, but the textural detail is poor compared to model (4). We can observe weird stripe patterns in the inferred region and a more noticeable seam when zoomed in. Comparing (3) and (4) in Fig. 5, it is obvious that incorporating coordinate helps the texture-space discriminator learn the structural information indirectly. Our full model takes advantage of each module and can generate high-fidelity textures as shown in (b).

For quantitative evaluation, we use the FaceScape dataset [38] which contains complete face textures of 938 people. For each texture, we applied 5 different masks with input yaw rotations of $[-60°, -30°, 0°, 30°, 60°]$, then infer complete face texture on them. Each inferred texture is rendered to image space again for evaluation, with all these five poses but the input one. We calculate SSIM [43] and perceptual similarity between these generated face images and the ground truth, where perceptual similarity is computed as the cosine similarity of embedded features from VGGface [28]. We further compute the Fréchet inception distance (FID) to assess the quality of the generated faces. Table 1 and Table 2 again show the effectiveness of different modules. Removing any of them will lead to a significant performance drop.

**Invariance to input pose.** In Fig. 7, the image in the first column shows the ground-truth full texture, and the others are inferred images by DSD-GAN with three different visibility masks corresponding to different poses. It can be observed that DSD-GAN can deal with various poses from frontal ones to large pose angles, producing natural and perceptually stable results. The numerical results in Table 1 also shows the more stable performance of DSD-GAN compared to others.

## 4.2. Method Comparison

In this section, we evaluate the performance of our DSD-GAN and compare it with other methods. We consider three groups of method for comparison:

- **Fully-supervised baselines**. We train two baseline models with accessible complete face texture dataset: *Multi-PIE face textures* constructed by [6] as their (partial) training data, and *BFM textures* from [30]. We compare our model trained without any complete data with these two baselines.
- **State-of-the-art methods**. We compare DSD-GAN to prior art including two face texture completion methods of UV-GAN [6] and Yamaguchi *et al*. [37], and a state-of-the-art texture model GANFIT [9]. Note that *all the three method leveraged complete face texture* for learning. Unfortunately, *none of them provide their trained models or full training datasets*, making the comparison very difficult. Therefore, we run our method on relevant image samples from their papers and visually compare with their results.
- **Image inpainting algorithms.** To check the advantage of DSD-GAN over existing image inpainting approaches, we further compare with three deep inpainting models: *DeepFill v2* [40], *PICNet* [42], and *RFRNet* [21], all trained on face images from CelebA-HQ. Unlike ours, these methods are trained in image space using full supervision.

### 4.2.1 Comparison to fully-supervised baselines

To obtain strong baselines, we carefully augmented the face texture data and designed the training scheme. The Multi-PIE dataset contains 2,387 complete face textures of 337 subjects constructed from Multi-PIE database [11]. For BFM textures, we randomly sampled 500 textures from the

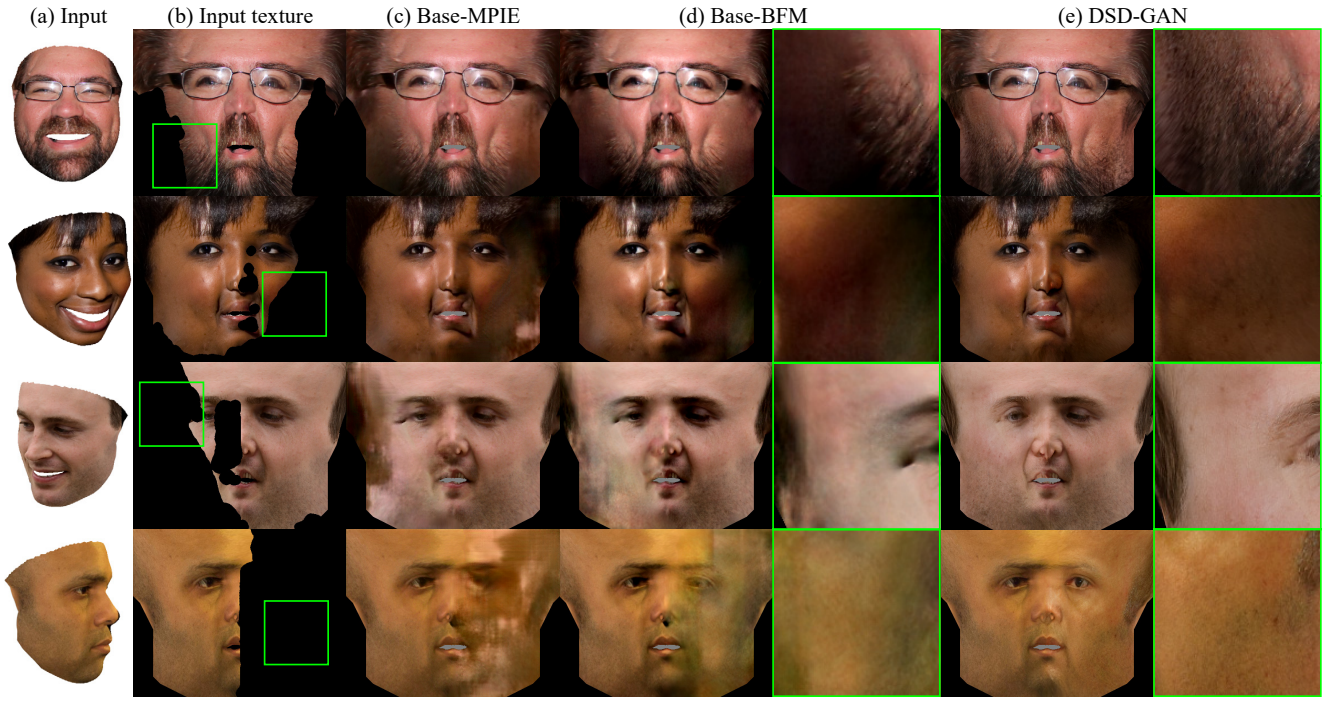



Figure 8: Comparison of texture completion results with two supervised baseline methods. (**Best viewed with zoom**)

BFM model and augmented them by changing skin tones, resulting in 6,500 samples for training. For both, we applied visibility masks with random shapes and poses sampled online for training. Random lighting was applied to the BFM textures by using Spherical harmonics (SH) lighting [7]. Both baseline models consist of a generator and discriminator in the texture space with architectures identical to DSD-GAN. To train them, $L_1$ ground-truth reconstruction loss and adversarial loss were combined with their weights properly tuned. We refer to the two trained models as 'Base-MPIE' and 'Base-BFM', respectively.

Figure 8 compares the qualitative results of different methods on the testing data from FFHQ. From the top row to the bottom row, the yaw angle of the face poses gradually increases. Baseline models generate reasonable results for the near-front faces. However, they clearly lack high-frequency details, especially under large poses. Our DSD-GAN yields the best quality for both near-frontal and large poses with rich high-frequency details generated.

Table 1 and 2 compare the quantitative results on the FaceScape dataset. As can be seen, our results are significantly more accurate than the baselines under all metrics.

#### 4.2.2 Comparison to state-of-the-art models

**Comparison to UV-GAN [6].** UV-GAN is trained with 77K complete face textures in a supervised fashion and infers up to $256\times256$ resolution. In contrast, our method learns to infer $512\times512$ textures without any complete texture. Figure 9 (left) compares the two methods using images from UV-GAN paper. As can be seen, both methods produce decent results, and UV-GAN handles a larger texture area than ours. However, our completed textures contain richer details. The re-rendered face images appear more realistic and better preserve the appearance of the raw input images. For large-pose faces, it seems that UV-GAN discarded the raw texture details visible in the input face (eyebrows, gaze, *etc*.) and regenerated the whole content, as can be observed from the third sample in Fig. 9 (left).

**Comparison to Yamaguchi *et al*. [37].** The method of Yamaguchi *et al*. learns to infer complete UV *albedo* using 329 production-level full 3D face scans. Figure 9 (right) compares our result with theirs. Since [37] generates albedo texture, we ignore the shading difference in the comparison. As shown in the figure, our texture map covers a larger facial region. By a closer look, the result of [37] may contain some seam artifacts on the boundary between visible and self-occluded parts (marked with green arrows), whereas ours look more natural.

**Comparison to GANFIT [9].** GANFIT is a state-of-the-art statistical parametric model of facial albedo texture that can be used for high-fidelity 3D face reconstruction. It is trained with $\sim$10K complete and high-quality albedo UV maps from [4]. In Fig. 10, we compare our results with GANFIT. Again we should ignore the shading difference here for a fair comparison. It can be seen that the results of GANFIT are of high quality and appear natural in general.

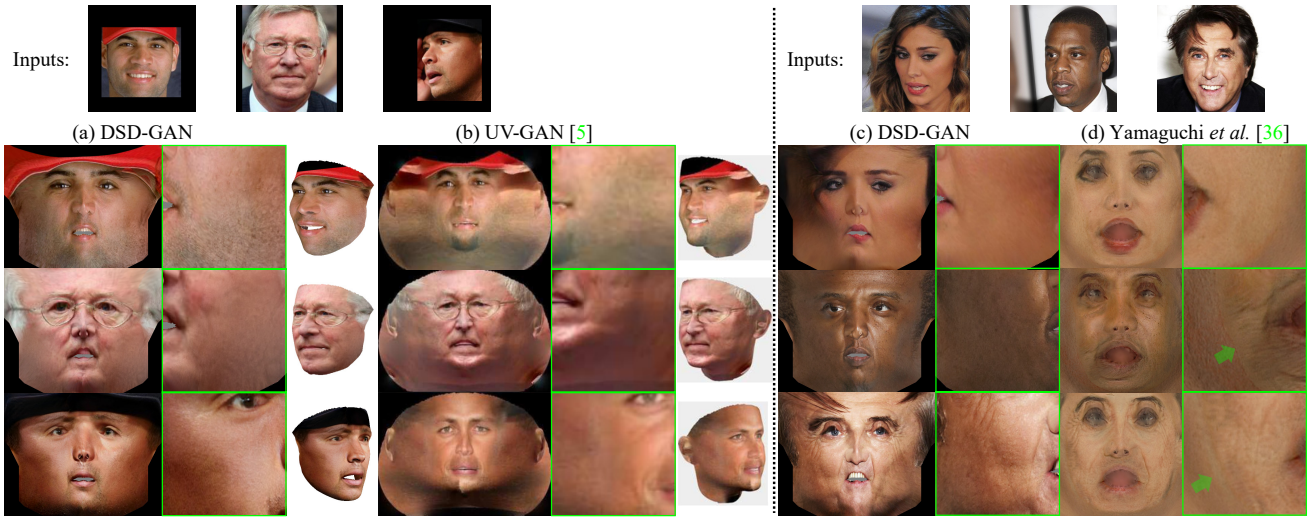



Figure 9: Comparison with UV-GAN [6] and Yamaguchi *et al*. [37]. The results of [6] and [37] are from their papers, and a full comparison can be found in the *suppl. material*. Green arrows indicate seam artifacts. (**Best viewed with zoom**)

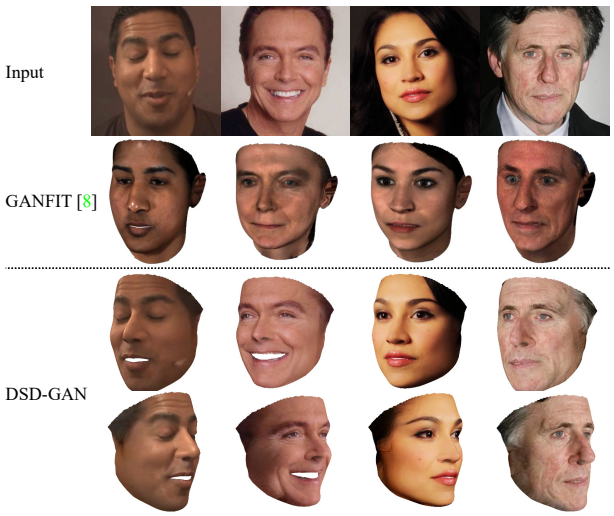



Figure 10: Comparison with GANFIT [9]. The results of [9] are from their paper, and a full comparison can be found in the *suppl. material*.

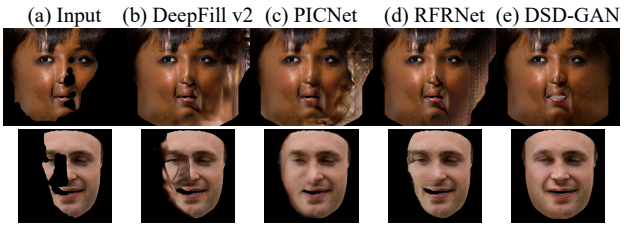



Figure 11: Inpainting results in texture space (top) and image (bottom) space. Our result in image space is obtained by unwarpping the image to texture space for completion then re-rendering back to image space.

However, it may not be the best way to preserve texture details of the inputs. Our method works consistently well for all these image samples.

Due to space constraint, more visual results including a comparison to another approach from Lin *et al*. [24] are presented in the *suppl. material*.

#### 4.2.3 Comparison to inpainting models

In Fig 11, we compare with three image inpainting algorithms: *DeepFill v2* [40], *PICNet* [42], and *RFRNet* [21]. For these methods, we use the models released by the authors which were all trained using face images from CelebA-HQ. We test them in both UV texture space and image space. Since they are not trained with texture-space data, it is expected that they may generate significant artifacts on UV texture. Their inpainting results in image space also suffer from obvious artifacts given large holes. We also quantitatively evaluate their inpainting performance in image space using the FaceScape dataset, and Table 1 and 2 show that their results less accurate than ours.

## 5. Conclusion

We presented a face texture completion framework that does not require any complete face textures, where dual-space discriminators work in a complementary manner. Our learned network could generate high-fidelity complete textures as shown in thorough experiments. DSD-GAN can be applied to various vision tasks. As shown by [7], generating synthetic data with largely different poses is a critical application to support the training of face-vision tasks. Improved texture quality will lead to a reduced domain gap between real and synthetic data. Furthermore, DSD-GAN can be used for free-view relighting with the combination of a simple albedo inference model (*e.g*. img2img translation). In addition, high-fidelity UV facial texture is applicable for realistic facial avatar generation for virtual reality.

## References


[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, and Zhifeng Chen et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2015. 5

[2] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In *ACM Siggraph*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 1, 2

[3] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3D Face Morphable Models "In-The-Wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 48–57, 2017. 2

[4] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3D Morphable Model Learnt From 10,000 Faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5543–5552, 2016. 2, 7

[5] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse Generative Neural Networks for Self-Supervised Scene Completion of RGB-D Scans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3

[6] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. UV-GAN: Adversarial Facial UV Map Completion for Pose-Invariant Face Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7102, 2018. 1, 2, 4, 6, 7, 8

[7] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, Mar. 2019. 2, 3, 7, 8

[8] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. In *European Conference on Computer Vision*, Mar. 2018. 2

[9] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: Generative Adversarial Network Fitting for High Fidelity 3D Face Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, Feb. 2019. 2, 6, 7, 8

[10] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised Training for 3D Morphable Model Regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, June 2018. 2, 4

[11] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and vision computing*, 28(5):807–813, 2010. 6

[12] Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng. CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1294–1307, 2018. 2

[13] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective Face Frontalization in Unconstrained Images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015. 1

[14] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3D Avatar Creation from Hand-Held Video Input. *ACM Transactions on Graphics*, 34(4):1–14, 2015. 1

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics*, 36(4):107, July 2017. 2

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 4

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*, 2018. 3

[18] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 3

[19] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. 5

[20] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. AvatarMe: Realistically renderable 3D facial reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–769, 2020. 1, 2

[21] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent Feature Reasoning for Image Inpainting. In *IEEE International Conference on Computer Vision*, pages 7760–7768, 2020. 6, 8

[22] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. COCO-GAN: Generation by Parts via Conditional Coordinating. In *IEEE International Conference on Computer Vision*, Mar. 2019. 4

[23] Jinpeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. Face Parsing with RoI Tanh-Warping. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 3

[24] Jiangke Lin, Yi Yuan, Tianjia Shao, and Kun Zhou. Towards High-Fidelity 3D Face Reconstruction from In-the-Wild Images Using Graph Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, Mar. 2020. 2, 8

[25] 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 *European Conference on Computer Vision*, pages 85–100, 2018. 2

[26] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *IEEE International Conference on Computer Vision*, Apr. 2017. 4

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier

GANs. In *International Conference on Machine Learning*, July 2017. 4

[28] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep Face Recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *British Machine Vision Conference*, pages 41.1–41.12. BMVA Press, Sept. 2015. 6

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

[30] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D Face Model for Pose and Illumination Invariant Face Recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Sept. 2009. 2, 6

[31] Stylianos Ploumpis, Haoyang Wang, Nick Pears, William A. P. Smith, and Stefanos Zafeiriou. Combining 3D Morphable Models: A Large scale Face-and-Head Model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[32] E. Richardson, M. Sela, and R. Kimmel. 3D Face Reconstruction by Learning from Synthetic Data. In *International Conference on 3D Vision*, pages 460–469, Oct. 2016. 2

[33] Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. Photorealistic Facial Texture Inference Using Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, Dec. 2016. 2

[34] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David Jacobs. SfSNet: Learning Shape, Reflectance and Illuminance of Faces in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6296–6305, 2018. 2

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3

[36] Sicheng Xu, Jiaolong Yang, Dong Chen, Fang Wen, Yu Deng, Yunde Jia, and Xin Tong. Deep 3D Portrait from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7710–7720, 2020. 1

[37] Shuco Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. High-Fidelity Facial Reflectance and Geometry Inference from an Unconstrained Image. *ACM Transactions on Graphics*, 37(4):162, 2018. 2, 4, 6, 7, 8

[38] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. FaceScape: A Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In *IEEE International Conference on Computer Vision*, Apr. 2020. 6

[39] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *International Conference on Learning Representations*, Apr. 2016. 3

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-Form Image Inpainting with Gated Convolution. In *IEEE International Conference on Computer Vision*, June 2018. 6, 8

[41] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative Image Inpainting with Contextual Attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 2

[42] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic Image Completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, Apr. 2019. 6, 8

[43] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. 6