# Progressive Reconstruction of Visual Structure for Image Inpainting

Jingyuan Li[1], Fengxiang He[2], Lefei Zhang*[1], Bo Du[1], Dacheng Tao[2]

[1]School of Computer Science, Wuhan University, Wuhan, China
[2]UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering
The University of Sydney, Australia

{jingyuanli, zhanglefei, remoteking}@whu.edu.cn, {fengxiang.he, dacheng.tao}@sydney.edu.au

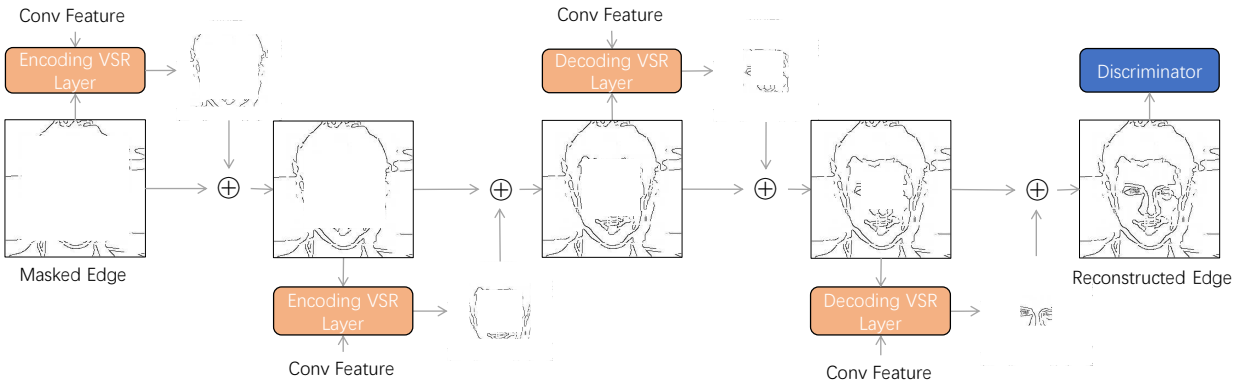


Figure 1: Progressive Reconstruction of Visual Structure. A small part of the new structure is produced in each VSR layer. At the beginning, the known information is limited and so the encoding layers only estimate the outer parts of the missing structure. As the information accumulates during the feeding forward procedure, the decoding layers can have the capability to restore the missing inner parts. The generated parts are collected and sent to discriminator simultaneously.

## Abstract

*Inpainting methods aim to restore missing parts of corrupted images and play a critical role in many computer vision applications, such as object removal and image restoration. Although existing methods perform well on images with small holes, restoring large holes remains elusive. To address this issue, this paper proposes a Progressive Reconstruction of Visual Structure (PRVS) network that progressively reconstructs the structures and the associated visual feature. Specifically, we design a novel Visual Structure Reconstruction (VSR) layer to entangle reconstructions of the visual structure and visual feature, which benefits each other by sharing parameters. We repeatedly stack four VSR layers in both encoding and decoding stages of a U-Net like architecture to form the generator of a generative adversarial network (GAN) for restoring images with either small or large holes. We prove the generalization error upper bound of the PRVS network is* $O\left(\frac{1}{\sqrt{N}}\right)$*, which theoretically guarantees its performance. Extensive empirical evaluations and comparisons on Places2, Paris Street View and CelebA datasets validate the strengths of the proposed approach and demonstrate that the model outperforms current state-of-the-art methods. The source code package is available at* https://github.com/jingyuanli001/PRVS-Image-Inpainting.

*Corresponding Author

## 1. Introduction

Image inpainting aims to restore missing parts in corrupted images. Recently, it has become an important task in computer vision and shows promising performance in many applications, such as object removal and image restoration [22, 24, 1].

Previous studies [28, 32, 29] based on texture searching produced reasonable results on images with small holes.

However, when filling large holes, these algorithms suffer from limited information (specifically, eligible structure information for recovering the lost parts) and usually hallucinate blurry textures or even meaningless content.

Recent studies try to solve this problem by introducing additional generators to estimate the visual structure of the missing part. They exploit the estimated visual structures as prior knowledge to improve recovery performance. For example, Nazeri *et al*. [16] and Xiong *et al*. [27] suggested explicitly encoding edge and saliency information to boost inpainting network performance, respectively. However, they failed to produce semantically meaningful and detailed structures. This is mainly because they utilized the adversarial loss to evaluate the generated structure, which treats each structure map as a whole and thus the network can hardly recover qualified local structures. Moreover, cascading two or more generators is suboptimal for parameter optimization.

In this paper, we design a Visual Structure Reconstruction (VSR) layer to restore visual structures by entangling the generation of structures and contents. Specifically, VSR adopts a partial convolution and a bottleneck block to restore a portion of edges in a missing region. The reconstructed edges are then combined with an input image with holes to progressively shrink the size of holes by filling semantically meaningful contents. We stack two VSR layers in the encoding stage and two VSR layers in the decoding stage. All four VSR layers together seamlessly assist a U-Net like architecture to progressively recovery the visual structure through the feeding forward procedure. We term the new end-to-end trainable GAN scheme for inpainting as the progressive reconstruction of visual structure (PRVS) network. This end-to-end network can be easily trained, and can appropriately restore the missing structure information for the subsequent recovery of the missing details.

For the discriminator used for detail generation, we follow [12] to integrate the style loss and the perceptual loss taken from a VGG-16 pre-trained on ImageNet [21]. For the discriminator used for structure generation, we integrate a Patch-GAN discriminator with spectral normalization and adversarial loss. The combined training target is expected to help the model learn to produce well-structured results.

We theoretically analyze the generalization ability of the proposed method and gives an $O\left(\frac{1}{\sqrt{N}}\right)$ generalization bound which leads to two practical implementations based on some recent results [2, 15, 34]. First, the generalization bound demonstrates a negative correlation between the generalization ability and the complexity of the discriminator. From this result, we adopt a pre-trained VGG in the discriminator and fix the weight matrices. Since the corresponding capacity of hypothesis space is only one (the potentially smallest capacity), the weight-fixed VGG can significantly reduce the hypothesis complexity of the discriminator and thereby improve the generalization ability. Second, the theoretical results suggest a negative correlation between the spectral norms of the weight matrices, which leads to the spectral normalization in order to control the spectral norms (it is also suggested by [2, 14, 17]).

Extensive experiments on standard datasets Places2 [35], Paris Street View [4] and CelebA [13] datasets are conducted. The results demonstrate that our method significantly outperforms the state-of-the-art methods.

## 2. Related Work

### 2.1. Generative Model for Image Inpainting

Image inpainting aims to recover missing areas of a damaged image. There have been significant improvements in image inpainting through the use of deep learning [11]. Pathak *et al*. [18] introduced GANs [5] to inpainting, albeit producing relatively low-resolution hallucinations. Iizuka *et al*. [7] introduced local and global discriminators, assisted by dilated convolution [30] and Poisson blending [19] to preserve the richness of high-frequency information and to handle rectangular masks at any location.

Since convolution filters can only extract local information, it is difficult for traditional GANs to capture texture information from distant areas. As a result, Yang *et al*. [29], Yan *et al*. [28] and Yu *et al*. [32] investigated to collect appearing features, utilizing the idea of patch-match on deep feature maps, which enable a GAN to generate sharp and accurate results. However, these methods were designed for rectangular holes and could not handle larger, irregular masks due to the difficulty in searching for suitable patches.

Liu *et al*. [12] proposed a partial convolution layer to help inpaint irregular holes. Values of a new feature map are calculated from non-masked regions; meanwhile, the mask in each layer is updated. Perceptual loss and style loss taken from a pre-trained VGG-16 [23] on ImageNet [21] have also been introduced to replace the traditional adversarial loss. Yu *et al*. [31] further deployed the gated convolution layers in the model of [32] for the irregular inpainting task. While the methods mentioned above have made significant contributions to the field of inpainting, the absence of structural knowledge has constrained their potential in recovering continuously masked images.

### 2.2. Structure Information for Inpainting

Wang *et al*. [25] showed that the binary edge maps of an image could benefit an image synthesizing model and help to evaluate object boundaries during image generation. The learnt edge maps can characterize the image structure. Also, it is easier to estimate binary maps than RGB images. Thus, there are also natural initiatives to reconstruct the visual structure for inpainting, such as Nazeri *et al*. [16] and Xiong *et al*. [27].

Inspired by human artists, Nazeri *et al*. [16] used two GANs for the inpainting task, which utilizes the edge map from the first generator as the prior of the inpainting network. Similarly, Xiong *et al*. [27] divided the model into multiple sub-networks to restore the image step by step, enabling the model to be aware of the saliency information. Both methods simplify the task of inpainting by building precise medium targets, i.e. recovering the edge or the foreground-background of the corrupted image.

However, as the corruption is getting larger, they fail to appropriately reconstruct the visual structure. Detailed reason is given in the introduction.

## 3. Approach

We design a Progressively Reconstruction of Visual Structure (PRVS) network for image inpainting. The generator adopts the P-UNet as the backbone (see Fig. 3), which replaces each convolution layer in U-Net [20] with a partial convolution layer [12] in order to capture the local information of irregular boundaries. Besides, the generator stacks a series of visual structure reconstruction (VSR) layers in both the encoding and decoding stages of the P-UNet backbone, which entanglingly reconstruct the visual structure (edges) and visual features in a progressive manner. An upsampling module combines the benefits of the transposed convolution and partial convolution also advances inpainting result.

Below, we first introduce the partial convolution layer which helps us keep track of the mask shape in each layer. Then, we detailedly present the VSR layer and the loss functions. Afterwards, we present the PRVS network for inpainting. For the convenience, the values of the masked area and non-masked area in the masks are assigned to 0 and 1 respectively in our work.

### 3.1. Visual Structure Reconstruction Layer

VSR layer is composed of a structure generator and a feature generator. The structure generator first updates the input edge to shrink the size of missing regions. The updated edge map is then used to guide the generation of the new feature. Below we first introduce the partial convolution [12] which helps us keep track of the mask shape. Then we introduce the generation of edge and feature inside the VSR layer.

#### 3.1.1 Partial Convolution

Partial convolution layer is helpful for recovering masked area [12]. In each step, partial convolution layer updates the mask; meanwhile, the values of the updated feature map only rely on the values in unmasked area. The new value of the mask is 1 if the sum of values in previous mask covered by convolution window is not 0. Let $\mathbf{X}'$ denotes the feature map generated by partial convolution layer. $x'_{ijk}$ means the new feature value at location $i, j$ in the $k^{th}$ channel. $\mathbf{W}_k$ is the $k^{th}$ convolution kernel in the layer. $\mathbf{x}_{ij}$ and $\mathbf{m}_{ij}$ is the input feature tensor patch and input mask tensor patch (whose size is the same as the convolution kernel) centered at location $i, j$ respectively.

$$x'_{ijk} = \begin{cases} \mathbf{W}_k^T(\mathbf{x}_{ij} \odot \mathbf{m}_{ij})\frac{sum(\mathbf{1})}{sum(\mathbf{m}_{ij})} + b, & \text{if sum}(\mathbf{m}_{ij}) \text{ != } 0 \\ 0, & \text{else} \end{cases}$$

Similarly, the value of new mask value at location $i, j$ can be expressed as:

$$m'_{ij} = \begin{cases} 1, & \text{if sum}(\mathbf{m}_{ij}) \text{ != } 0 \\ 0, & \text{else} \end{cases}$$

The partial convolution layers help us keep track of the mask shape during the feed-forward procedure and enable us to progressively reconstruct the visual structures.

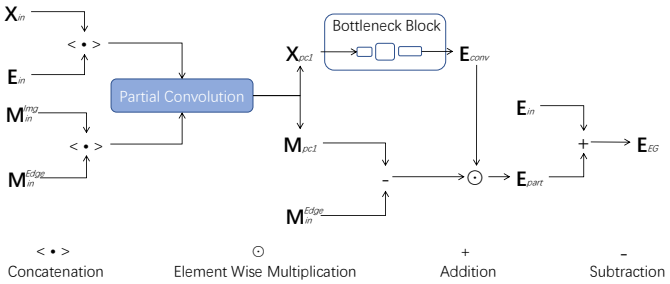


Figure 2: The generation of visual structure. Structure part is generated by a partial convolution followed by a residual block, then combined with input structure.

#### 3.1.2 Visual Structure Generator

In this section, we denote the partial convolutions as $Pconv(,)$, where the first parameter is the input feature and the second is the input mask. We use $\langle , \rangle$ to express concatenation in the channel dimension. There are four input factors in the structure generator, which are the image feature map $\mathbf{X}_{in} \in R^{H \times W \times C}$, structure map $\mathbf{E}_{in} \in R^{H \times W \times 1}$, previous mask for image $\mathbf{M}_{in}^{Img} \in \{0,1\}^{H \times W \times C}$, and previous mask for edge $\mathbf{M}_{in}^{Edge} \in \{0,1\}^{H \times W \times 1}$, respectively (See Fig. 2). These two masks have the same shape but different channel numbers. We first adopt a partial convolution layer to update the feature map and mask as follows,

$$\mathbf{X}_{pc1}, \mathbf{M}_{pc1} = Pconv(\langle \mathbf{X}_{in}, \mathbf{E}_{in} \rangle, \langle \mathbf{M}_{in}^{Img}, \mathbf{M}_{in}^{Edge} \rangle) \quad (3.1)$$

The feature map generated by the first partial convolution is then fed to a residual block [6] and a one-channel output convolution kernel to produce a structure map $\mathbf{E}_{conv}$. In this paper, we use the bottleneck residual block with kernel sizes

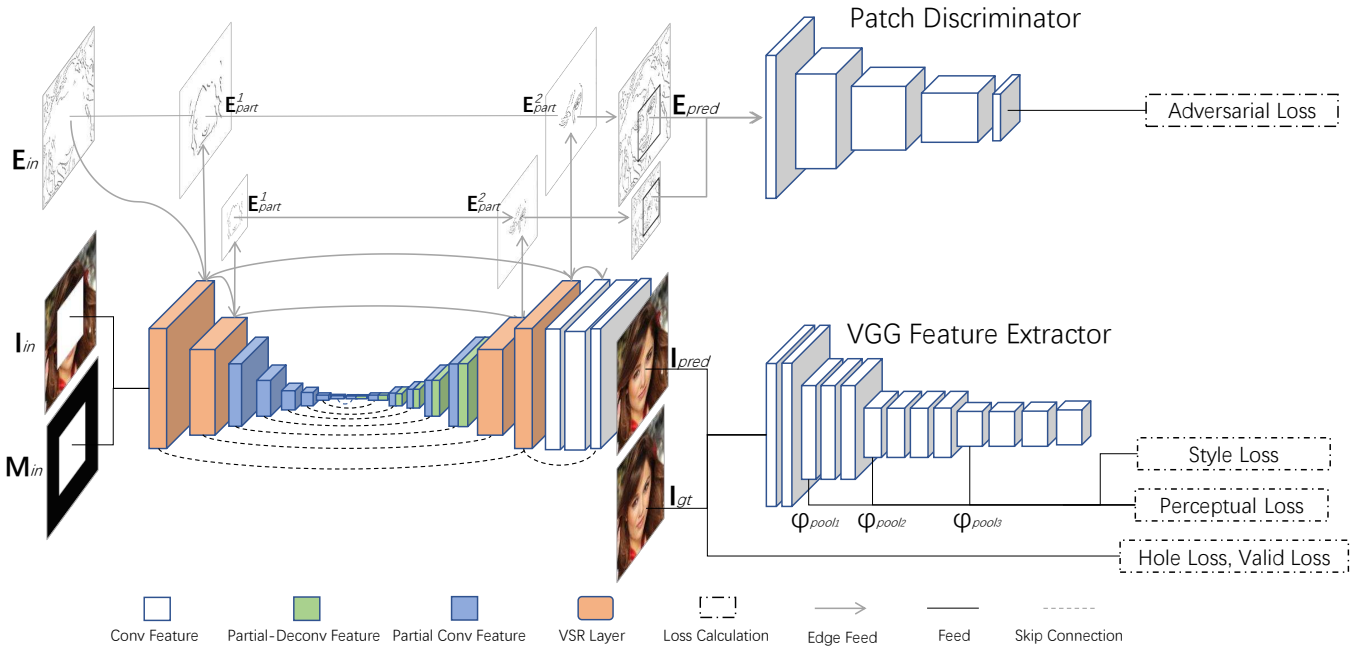


Figure 3: Overall architecture of our proposed model. The VSR Layer is put in the first two layers and last two layers in our network. The generated structure and feature maps are sent to next and decoding layers. Finally, two structure (edge) maps of different scales are generated to learn structure information.

and channel numbers of 1, 3, 1 and 64, 16, 64, respectively. We then use the mask $\mathbf{M}_{pc1}$ from the partial convolution to correct the shape of structure map. The input structure $\mathbf{E}_{in}$ is used to replace the previously known area in the new structure map $\mathbf{E}_{conv}$ and so only the newly generated parts $\mathbf{E}_{conv} \odot (\mathbf{M}_{pc1} - \mathbf{M}_{in})$ are preserved. This can be described as Eq. (3.2). This helps the partial convolution and residual block in the edge generator focus on the newly generated part.

$$\mathbf{E}_{EG} = \mathbf{E}_{conv} \odot (\mathbf{M}_{pc1} - \mathbf{M}_{in}^{Edge}) + \mathbf{M}_{in}^{Edge} \tag{3.2}$$

The final outputs of structure generator are $\mathbf{M}_{pc1}$ and $\mathbf{E}_{EG}$. In our design, the generators only need to estimate structure parts that are closest to the known area, which is easier to generate based on the feature map.

#### 3.1.3 Architecture of VSR Layer

The main purpose of the VSR layer is to incorporate the structural information in the reconstructed feature map. We concatenate $\mathbf{E}_{EG}$ with the input original feature map $\mathbf{X}_{in}$, using the structure map to guide the generation of next feature map. The concatenated feature maps and the corresponding masks are then sent into another partial convolution layer to update the image feature map $\mathbf{X}_{out}$ (see Eq. (3.3)). We use the mask $\mathbf{M}_{pc1}$ from the structure generator (which is only updated once) as the output mask and use it to correct the shape of image feature map (the element wise multiplication in Eq. (3.3)). If any down-sampling operation makes the new feature map become smaller, max pooling is applied to $\mathbf{M}_{in}$ to produce a mask of expected shape.

$$\mathbf{X}_{out}, \mathbf{M}_{pc2} = \mathbf{M}_{pc1} \odot (Pconv(\langle \mathbf{X}_{in}, \mathbf{E}_{EG} \rangle, \langle \mathbf{M}_{in}, \mathbf{M}_{pc1} \rangle)) \tag{3.3}$$

The generated feature from VSR layer carries more structural information in this way, which helps recover the image. The final outputs of the VSR layer are $\mathbf{E}_{EG}$, $\mathbf{X}_{out}$ and $\mathbf{M}_{pc1}$.

### 3.2. Structure Learning and Loss Functions

Many structure parts are generated by VSR layers and it's time-consuming to learn these structure parts separately. Note that the newly restored structures from different layers do not share any overlapping region, so it is natural to cumulate the restored structures. We therefore filter out the parts of structure that are not used to assist image generation and keep only the newly generated parts like Eq. (3.4), where the $\mathbf{E}_{part}$ is the newly generated structure in each VSR layer:

$$\mathbf{E}_{part} = \mathbf{E}_{EG} \odot (\mathbf{M}_{pc1} - \mathbf{M}_{in}) \tag{3.4}$$

Those filtered parts from generators are collected and sent to discriminator. We use $E^i$ to denote the combined structure map from $i^{th}$ VSR layer at each level, where "level"

means the group of layers having the same input size. $E^0$ means the original input. The combination process can be expressed as following:

$$\mathbf{E}^{i+1} = \mathbf{E}^i \odot \mathbf{M}^i + \mathbf{E}^{i+1}_{part} \tag{3.5}$$

In this way, two structure maps of different scales (256 and 128) are generated.

For the discriminator, we use a Patch-GAN [9] discriminator and a pre-trained and fixed VGG-16 network [23] for structure generation learning and image generation learning, respectively, as follows. For structure generation learning, we use a Patch-GAN discriminator to evaluate whether each structure patch belongs to the real or fake distribution. The Patch-GAN discriminator calculates the adversarial loss for structure from the generator. The adversarial loss for structure map from $i^{th}$ level is denoted as $L^i_{adv}$. Besides, spectral normalization [14], which divides weight matrix by the corresponding Lipschitz constant, is applied in our discriminator. Theoretical analysis demonstrates that spectral normalization can control the generalization error (see also [34]).

For image generation learning, the perceptual loss and style loss from a pre-trained and fixed VGG-16 are used. The perceptual loss and style loss compare the difference between the deep feature map of the generated image and the ground truth. These loss functions are formalized in the following. $\phi_{pool_i}$ means feature maps from $i^{th}$ pooling layer in the fixed VGG-16. In following equations, $H_i$, $W_i$ and $C_i$ are used to express the height, weight and channel size of the $i^{th}$ feature map. The perceptual loss can be then written as following:

$$L_{preceptual} = \sum_{i=1}^{N} \frac{1}{H_i W_i C_i} |\phi^{gt}_{pool_i} - \phi^{pred}_{pool_i}|_1 \tag{3.6}$$

Similarly, the computation of style loss is as follow:

$$\phi^{style}_{pool_i} = \phi_{pool_i} \phi^T_{pool_i} \tag{3.7}$$

$$L_{style} = \sum_{i=1}^{N} \frac{1}{C_i * C_i} \Big| \frac{1}{H_i W_i C_i} (\phi^{style_{gt}}_{pool_i} - \phi^{style_{pred}}_{pool_i}) \Big|_1 \tag{3.8}$$

Besides, $L_{valid}$ and $L_{hole}$ which calculate L1 differences in the unmasked area and masked area respectively are also used in our model. In summary, our total loss function is as follow:

$$\begin{aligned} L_{total} = {} & \lambda_{hole} L_{hole} + \lambda_{valid} L_{valid} + \lambda_{tv} L_{tv} + \lambda_{style} L_{style} \\ & + \lambda_{perceptual} L_{perceptual} + \lambda_{adv} (L^1_{adv} + L^2_{adv}) \end{aligned} \tag{3.9}$$

Although the perceptual loss and style loss are designed for learning RGB image generation, the shared parameters make structure generation benefit from the target functions. Similarly, the image generation also benefits from the adversarial loss for structure learning.

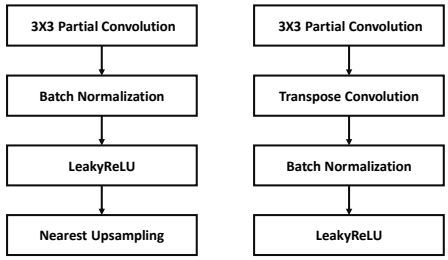


Figure 4: Partial-deconvolution up-sampling. On the left is the previous up-sampling module in P-UNet, on the right is ours. We add a deconvolution layer following the partial convolution. Nearest up-sampling is used to enlarge the mask in the decoder stage of PRVS network.

### 3.3. Overall Architecture

In the generator, two VSR layers are deployed in the encoder stage and two are deployed in the decoder stage of the P-UNet (16 layers), as shown in Fig. 3. At the beginning of the generator, there is little information in the corrupted region and a single VSR layer is not expected to recover the visual structure for the whole corrupted area. However, as information accumulates during down-sampling and up-sampling, the VSR layers in the decoding stage are capable to estimate the visual structure of the central area. As a result, the outer and inner visual structures are generated by VSR layers in the encoding and decoding stages, respectively. These VSR layers together form the visual structure for inpainting. Besides the VSR layer, a partial-deconvolution layer (Fig. 4) which combines partial convolution with transpose convolution is also used in the up-sampling layers. In the original P-UNet, the skip connections make it hard to directly apply the transposed convolution without harming the benefits of partial convolutions. To address this issue, we use a partial convolution layer to make the mask shape the same in different channels followed by transposed convolution [33] to up-sample feature map. A bottleneck residual block is added to the end of our model (the white blocks in Fig. 3) to merge the last structure map. Contextual attention modified from [32] is also used to help obtain better textures before the third last layer. For more details, please refer to Appendix A.

## 4. Theoretical Analysis

Generalization ability is of vital importance to machine learning algorithms, which refers to the ability to generalize the good performance on training data to unseen data. Our proposed method is built based on a GAN which is used to generate a group of new sample points that follow the distribution of the existing data. The learning procedure is to narrow the gap between the latent distribution of the existing data and the generated data. A recent theoretical result

demonstrates that the discriminator is the bottleneck of the generalization abilities of GANs. The generalization ability of GANs is guaranteed as long as the hypothesis complexity of the discriminator is small enough, regardless of the size of the generator hypothesis set (see Lemma 3 in Appendix B.2; cf. [34], Theorem 3.1). Denote the latent distributions of the existing data and the generated data as $\mu$ and $\nu$, respectively. Suppose the empirical distribution of the training sample set is $\hat{\mu}_N$ and the empirical distribution of the generated data is $\nu_N$, where $N$ is the size of the training sample set. Denote the generator as $g \in \mathcal{G}$ and the discriminator as $f \in \mathcal{F}$, where $\mathcal{G}$ is the distribution class of the generated data and $\mathcal{F}$ is the hypothesis class of the discriminator. Mathematically, GANs minimize the integral probability metric (IPM) $d_{\mathcal{F}}(\hat{\mu}_N, \nu)$ between the distributions $\hat{\mu}_N$ and $\nu$ [15], which is defined as:

$$d_{\mathcal{F}}(\hat{\mu}_N, \nu) \triangleq \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{x \in \hat{\mu}_N}[f(x)] - \mathbb{E}_{x \in \nu}[f(x)] \right\}. \quad (4.1)$$

Meanwhile, $d_{\mathcal{F}}(\mu, \nu_N)$ expresses the distance between the latent distribution of existing data and the empirical distribution of the generated data, which is usually called the empirical risk. Additionally, $\inf_{\nu \in \mathcal{G}} d_{\mathcal{F}}(\mu, \nu)$ expresses the distance between the best hypothesis and the observed data, which is usually called the expected risk. Finally, the generalization error of GANs is defined as:

$$d_{\mathcal{F}}(\mu, \nu_N) - \inf_{\nu \in \mathcal{G}} d_{\mathcal{F}}(\mu, \nu). \quad (4.2)$$

For more details about the definition of the generalization error, please refers to [34].

As Fig. 3 shows, the discriminator is constituted by two parts, a pre-trained and weight-fixed VGG-16 classifier and a five-layer CNN (the patch discriminator). For the brevity, we denote these two parts respectively as VGG Feature Extractor (VFE) and patch discriminator (PD). Specifically, the PD is constituted by a series of convolutional layers and nonlinear operations (nonlinearities) which are expressed as $(A_1, \sigma_1, A_2, \sigma_2, A_3, \sigma_3, A_4, \sigma_4, A_5, \sigma_5)$, where $A_i$ is a convolutional layer, and $\sigma_i$ is a nonlinearity (leaky ReLU). Then we can obtain the following lemma on the hypothesis complexity of the discriminator.

**Theorem 1** (Covering bound for the discriminator)**.** *Suppose the spectral norm of each weight matrix is bounded: $\|A_i\|_\sigma \leq s_i$. Also, suppose each weight matrix $A_i$ has a reference matrix $M_i$, which is satisfied that $\|A_i - M_i\|_\sigma \leq b_i$, $i = 1, \ldots, 5$. The Lipschitz constant of $\sigma_5$ is supposed as $\rho$. Then, the $\varepsilon$-covering number satisfies that*

$$\log \mathcal{N}\left(\mathcal{F}|_S, \varepsilon, \|\cdot\|_2\right) \leq \frac{\log\left(2W^2\right)\|X\|_2^2}{\varepsilon^2}\left(\rho \prod_{i=1}^{5} s_i\right)^2 \left(\sum_{i=1}^{5} \frac{b_i^{2/3}}{s_i^{2/3}}\right)^3, \quad (4.3)$$

*where $W$ is the largest dimension of the feature maps throughout the algorithm.*

A detailed proof is omitted here but provided in the appendix. Finally, we obtain the following theorem. For brevity, we denote the right-hand side (RHS) of Eq. (4.3) as $\frac{R^2}{\varepsilon^2}$.

**Theorem 2.** *Assume that the discriminator set $F$ is even, i.e., $f \in \mathcal{F}$ implies $-f \in \mathcal{F}$, and that all discriminators are bounded by $\Delta$, i.e., $\|f\|_\infty \leq \Delta$ for any $f \in \mathcal{F}$. Assume $\hat{\mu}_N$ and $\nu_N$ satisfy*

$$d_{\mathcal{F}}(\hat{\mu}_N, \nu_N) \leq \inf_{\nu \in \mathcal{G}} d_{\mathcal{F}}(\hat{\mu}_N, \nu) + \phi. \quad (4.4)$$

*Then with probability at least $1 - \delta$, we have*

$$d_{\mathcal{F}}(\mu, \nu_N) - \inf_{\nu \in \mathcal{G}} d_{\mathcal{F}}(\mu, \nu) \leq \frac{24R}{N}\left(1 + \log\frac{N}{3R}\right) + 2\Delta\sqrt{\frac{2\log(\frac{1}{\delta})}{N}} + \phi. \quad (4.5)$$

A detailed proof is omitted here but provided in the appendix. Eq. (4.5) gives an $O\left(\frac{1}{\sqrt{N}}\right)$ generalization bound for our proposed and provides two practical implementations: (1) use a pre-trained and fixed VGG-16 as a part of the discriminator. Thereby, we significantly reduce the hypothesis complexity and enhance the generalization ability; and (2) utilize the regularization technique of spectral normalization to scale the spectral norms of all weight matrices to 1 (to make $s_i = 1$), which is much lower than the ones without spectral normalization. Meanwhile, there is a positive correlation between our generalization bound (Eq. (4.5)) and the product of the spectral norms of all weight matrices. Therefore, spectral normalization can also significantly help to achieve a significantly lower upper bound for the generalization error, and thus improve generalization ability.

## 5. Experiments & Results

### 5.1. Setup

Our model was trained with the batch size of 5 on an NVIDIA RTX 2080TI 11G GPU. We used the Adam Optimizer [10] to optimize our generator and discriminator. We first used $2 \times 10^{-4}$ as our initial learning rate to train our model. Then we finetuned our model with a learning rate of $1 \times 10^{-5}$. During finetuning, the batch normalization layers [8] in the encoding stage of generator were frozen to stabilize training. It took three days including one day's finetuning to train the model on CelebA and Paris Street View datasets. For Places2, two weeks' training and two days' finetuning was needed. For the hyper-parameters, we chose 50, 50, 0.01, 180, 0.1, 0.1 for $\lambda_{hole}$, $\lambda_{valid}$, $\lambda_{tv}$, $\lambda_{style}$, $\lambda_{perceptual}$, $\lambda_{adv}$ respectively.

Figure 5: Comparisons of inpainting methods. From left to right: Masked Images. Edge-Connect [16]. PConv [12]. Ours. Our model is able to generate high quality result even if the mask is large. The results are from Places2 and Paris Street View datasets. All images are not post processed.

| Places-SSIM | P-UNet | Edge-Connect | Ours |
|---|---|---|---|
| 10%-20% | 0.944 | 0.942 | **0.956** |
| 20%-30% | 0.892 | 0.891 | **0.914** |
| 30%-40% | 0.833 | 0.831 | **0.861** |
| 40%-50% | 0.762 | 0.759 | **0.797** |
| 50%-60% | 0.631 | 0.629 | **0.672** |
| Places-PSNR | P-UNet | Edge-Connect | Ours |
| 10%-20% | 27.67 | 27.48 | **28.87** |
| 20%-30% | 24.60 | 24.54 | **25.66** |
| 30%-40% | 22.52 | 22.53 | **23.46** |
| 40%-50% | 20.88 | 20.92 | **21.74** |
| 50%-60% | 18.80 | 18.83 | **19.51** |
| Places-MAE | P-UNet | Edge-Connect | Ours |
| 10%-20% | 0.0147 | 0.0151 | **0.0125** |
| 20%-30% | 0.0262 | 0.0265 | **0.0225** |
| 30%-40% | 0.0388 | 0.0389 | **0.0337** |
| 40%-50% | 0.0530 | 0.0531 | **0.0466** |
| 50%-60% | 0.0768 | 0.0768 | **0.0689** |

Table 1: Results from Places2 Dataset. The methods compared are designed for irregular hole inpainting tasks. Comparisons with Yu *et al*. [31] are in the Appendix C.

### 5.2. Training & Testing

We evaluated our model and compared baselines on the following datasets:

-**Places2 Challenge Dataset**: A dataset released by MIT containing over 8,000,000 images from over 365 scenes. Although the dataset is designed for classification, it is suitable for building inpainting models as it enables the model to learn the distribution from many natural scenes.

-**CelebA Dataset**: A dataset focuses on human face images, containing over 180,000 training images. The models trained on this dataset can be easily transferred to face editing/completion tasks.

-**Paris Street View Dataset**: A dataset commonly used for inpainting methods. It contains 14,900 training images and 100 testing images.

For images from CelebA dataset, we cropped the center 178×178 pixels from the images. For Paris Street View dataset, we divided the training image into left, middle and right and therefore obtained 44,700 images in total. All images for our experiment were resized to 256×256. For ground truth structure, the Canny edge [3] algorithm was used. The masks for model training and testing is from [12]. For testing, we chose 10,000 images from the dataset, iteratively using the testing mask grouped by mask ratio.

### 5.3. Quantitative Result

For quantitative analysis, we compared our model with current state-of-the-art methods on Paris Street View, Places2 and CelebA datasets. The results in Table 1 were averaged on 10,000 images from Places2 validation set. We tested the models on different mask ratios (the percentages in the first column). The results on other datasets can be found in Appendix C. The compared models are 1) Edge-Connect [16] 2) P-UNet[12] and 3) GatedConv[31]. Our model shows the superiority in quantitative results. We evaluated the generated results from the aspect of peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and mean absolute error (MAE) [26].

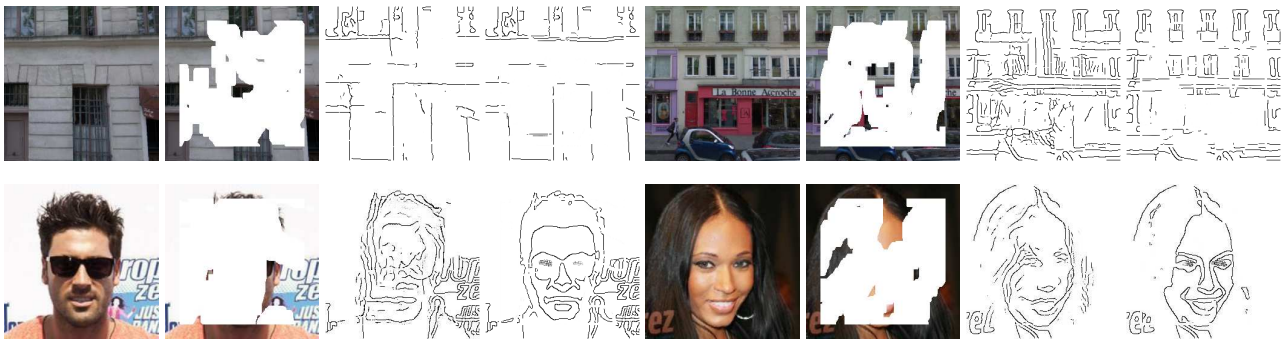

Figure 6: Comparisons between different ways to generate structure (edge). From left to right: Ground truth, masked image, edge from a single generator, edge from our model. Edges from our model described necessary structure details better.

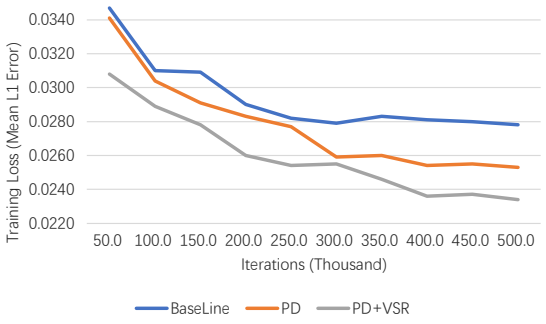


Figure 7: Training loss of each module. PD means baseline and partial-deconvolution layers. PD+VSR means baseline, partial-deconvolution layers and the VSR layers.

### 5.4. Inpainting Quality Comparison

From the perspective of applications, qualitative results are more important than quantitative results. We compared our model with respect to visual outcome in Fig. 5. According to this figure, it is observed that as the hole size gets larger, previous models become unstable while our model can still produce well-structured content.

We expect the model to benefit from joint training of structure generation and image generation by parameter sharing. To validate our idea, comparisons on edge quality between various model were also conducted in Fig. 6. We compared edges from our model with that from a single generator.

### 5.5. Effectiveness of modules

We tested the effectiveness of each new module in Table 2, which consisted of three combinations of different modules. The first model tested was P-UNet using the same hyper-parameters as that mentioned in Section 5.1. The second model was equipped with the partial-deconvolution layer to replace the nearest up-sampling. The third model includes our VSR layer. We used the same hyper-parameters for each model to ensure fairness. The pixel attention module [32] was removed except for "Full" to address its possible impact.

| | Baseline | PD | PD+VSR | Full |
|---|---|---|---|---|
| SSIM | 0.697 | 0.707 | 0.716 | 0.724 |
| PSNR | 22.04 | 22.19 | 22.31 | 22.48 |
| MAE | 0.0572 | 0.0556 | 0.0545 | 0.0534 |

Table 2: Effectiveness of modules. We tested each module on Paris Street View dataset with the mask ratio of 50%-60%.

## 6. Conclusion

In this paper, we propose a novel image inpainting method that progressively incorporates structure information into the feature to output more structured image based on generated adversarial networks (GANs). Specifically, the generator adopts four novel visual structure reconstruction (VSR) layers to progressively reconstruct the structure. Besides, partial-deconvolution is utilized in the generator in order to address the limitation of partial convolution with existing modules. In the discriminator, we adapt a patch discriminator to evaluate the generated structures with the adversarial loss and a pre-trained and weight-fixed VGG-16 to evaluate the images with style loss and perceptual loss. Theoretical analysis evaluates our method and gives a theoretical guarantee. Extensive experiments on many standard datasets validate the feasibility of our method.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China under grants 61771349 and 61822113, and Australian Research Council Projects FL-170100117 and DP180103424.

## References


[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009. 1

[2] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Proc. NIPS*, pages 6240–6249, 2017. 2, 11, 12, 13, 14

[3] J Canny. A computational approach to edge detection. *IEEE TPAMI*, 8(6):679–698, 1986. 7

[4] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM TOG*, 31(4):101, 2012. 2

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 3

[7] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM TOG*, 36(4):107, 2017. 2

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, pages 448–456, 2015. 6

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017. 5

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[11] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proc. ISCAS*, pages 253–256, 2010. 2

[12] 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 *Proc. ECCV*, pages 85–100, 2018. 2, 3, 7, 14

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, pages 3730–3738, 2015. 2

[14] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2, 5

[15] Alfred Muller. Integral probability metrics and their generating classes of functions. *AAP*, 29(2):429–443, 1997. 2, 6

[16] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 2, 3, 7, 14

[17] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017. 2

[18] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016. 2

[19] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM TOG*, 22(3):313–318, 2003. 2

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pages 234–241, 2015. 3

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2

[22] Rakshith Shetty, Mario Fritz, and Bernt Schiele. Adversarial scene editing: Automatic object removal from weak supervision. *CoRR*, abs/1806.01911, 2018. 1

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 5

[24] Linsen Song, Jie Cao, Linxiao Song, Yibo Hu, and Ran He. Geometry-aware face completion and editing. *CoRR*, abs/1809.02967, 2018. 1

[25] 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 *Proc. CVPR*, pages 8798–8807, 2018. 2

[26] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 7

[27] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. *arXiv preprint arXiv:1901.05945*, 2019. 2, 3

[28] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proc. ECCV*, pages 3–19, 2018. 1, 2

[29] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proc. CVPR*, pages 6721–6729, 2017. 1, 2

[30] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[31] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2, 7, 14

[32] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proc. CVPR*, pages 5505–5514, 2018. 1, 2, 5, 8

[33] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Robert Fergus. Deconvolutional networks. In *Proc. CVPR*, pages 2528–2535, 2010. 5

[34] Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in gans. *arXiv preprint arXiv:1711.02771*, 2017. 2, 5, 6, 13

[35] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2018. 2