Diverse Image Inpainting with Normalizing Flow

Cairong Wang*[1], Yiming Zhu*[1], and Chun Yuan[1,2]✉

[1] Tsinghua Shenzhen International Graduate School, China
{wcr20,zym20}@mails.tsinghua.edu.cn
yuanc@sz.tsinghua.edu.cn
[2] Peng Cheng National Laboratory, China

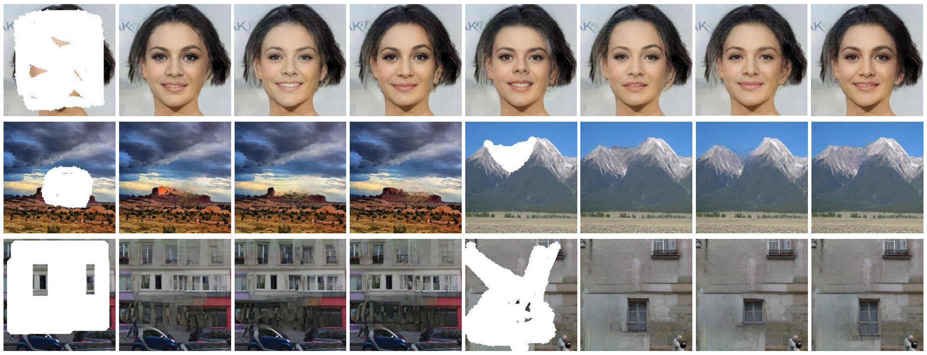

Fig. 1. Diverse free-form image completion results produced by our method.

Abstract. Image Inpainting is an ill-posed problem since there are diverse possible counterparts for the missing areas. The challenge of inpainting is to keep the "corrupted region" content consistent with the background and generate a variety of reasonable texture details. However, existing one-stage methods that directly output the inpainting results have to make a trade-off between diversity and consistency. The two-stage methods as the current trend can circumvent such shortcomings. These methods predict diverse structural priors in the first stage and focus on rich texture details generation in the second stage. However, all two-stage methods require autoregressive models to predict the probability distribution of the structural priors, which significantly limits the inference speed. In addition, their discretization assumption of prior distribution reduces the diversity of the inpainting results. We propose Flow-Fill, a novel two-stage image inpainting framework that utilizes a conditional normalizing flow model to generate diverse structural priors in the first stage. Flow-Fill can directly estimate the joint probability density of the missing regions as a flow-based model without reasoning pixel by pixel. Hence it achieves real-time inference speed and eliminates

* indicates equal contribution

discretization assumptions. In addition, as a reversible model, Flow-Fill can invert the latent variables for a specified region, which allows us to make the inference process as semantic image editing. Experiments on benchmark datasets validate that Flow-Fill achieves superior diversity and fidelity in image inpainting qualitatively and quantitatively.

Keywords: Diverse Image Inpainting, Normalizing Flow

1 Introduction

Image inpainting aims to generate meaningful content to fill in a corrupted image's missing areas. Unfortunately, it is a fundamentally ill-posed problem. For a given corrupted image (masked image), theoretically, there exist infinitely many natural (i.e., visually realistic and semantically reasonable) repaired images. This poses a significant challenge to synthesize diverse natural contents that maintain consistency with contextual information of the known image regions.

With the advancement of generative networks (such as VAEs, GANs), deep learning based inpainting methods [12,18,19,23,24,27,34,36–39] typically utilize an encoder-decoder framework to synthesize the high-level semantic information consistent with the context. They treat Inpainting as an image reconstruction problem, filling in the corrupted areas by learning a one-to-one mapping to the ground truth. Although these methods yield a realistic result to fill the empty region, they cannot generate multiple possibilities. In contrast to deterministic Inpainting, many studies have recently emerged to challenge the ill-posed problem by addressing diverse Inpainting. For example, Zheng *et al.* [44] and Zhao *et al.* [42] use VAE-based networks to learn the prior distributions of missing parts conditional on the given corrupted image. Modulated GAN-based methods [20,43] modulate the deep features of random input noise from coarse to fine. Diffusion-based methods [22,28] restore images from random noise by iteration. However, these one-stage methods have to make a trade-off between diversity and consistency. In addition to these one-stage methods, some two-stage methods [25, 32, 40] predict diverse structural priors in the first stage and focus on rich texture details generation in the second stage. Although two-stage methods moderate the trade-off of diversity and fidelity, all of them require autoregressive models to predict the probability distribution of the structural priors[1], which significantly limits the inference speed. In addition, their discretization assumption of prior distribution reduces the diversity of the inpainting results.

This paper follows the previous two-stage pipeline and develops Flow-Fill, a diverse image inpainting framework that utilizes a conditional normalizing flow network to naturally and accurately learn the distribution of structural priors in the first stage. As a result, Flow-Fill can achieve real-time inference speed and eliminate discretization assumptions compared to the existing two-stage

[1] Wan *et al.* [32] predicts the missing pixels one by one in an autoregressive form when inference. In addition, Wan *et al.* and Yu *et al.* [40] are based on Transformer, making the inference unbearably slow.

methods. Furthermore, compared to VAE and modulated GAN-based one-stage approaches mentioned above, Flow-Fill is reversible in the first stage and, therefore, can accurately learn the distribution of structural priors without suffering from mode-collapse and posterior collapse. In addition, as a flow-based model, Flow-Fill can map the input images to the corresponding latent variables and ensure exact reconstruction. Therefore, we can invert specific regions to the latent variable space for semantic Image Editing.

The main contributions in this work can be summarized as follows:

- We propose Flow-Fill, a flow-based two-stage diverse image inpainting framework. Flow-Fill is the first conditional normalizing flow network that completes large irregular corrupted areas with diverse results and achieves state-of-the-art image inpainting performance.
- As a flow-based model, Flow-Fill constructs a conditionally reversible bijective function, allowing inversion and inference about the content of a specific area. Therefore, Flow-Fill can be extended to region-specific semantic transfer tasks.
- Our Flow-Fill achieves a real-time inference speed that is approximately 87 times faster than autoregressive-based models and 142 times faster than diffusion-based models.
- Extensive experiments over multiple benchmark datasets demonstrate our proposed model's superiority in quality and diversity.

2 Related Work

As an ill-posed problem, image inpainting has multiple realistic and high-fidelity results. Based on the number of inpainting solutions, most existing image inpainting methods can be broadly classified into deterministic image inpainting and diverse image inpainting.

Deterministic Image Inpainting. Traditional image inpainting is mainly divided into diffusion-based methods [2, 8] and patch-based methods [4, 10]. Diffusion-based methods gradually spread the contextual pixel information to the damaged regions. Patch-based methods find the best matching patch in the visible area or a specified data library and then transfer it to the hole. With the advancement of generative networks, deep-learning based inpainting [12, 18, 19, 23, 24, 27, 34, 36–39] often uses generative adversarial networks (GANs) to learn high-level semantic information consistent with the context. However, while these deep-learning-based methods generate realistic complementary results for the hole regions, they cannot generate diverse semantically meaningful results.

Diverse Image Inpainting. In order to obtain pluralistic image inpainting results, current methods can be broadly classified into four categories: 1) Some model the prior probability of the missing region based on a VAE paradigm

[42, 44]. Specifically, Zheng *et al.* [44] uses two coupled parallel VAEs to model the prior probability of the missing part. In contrast, Zhao *et al.* [42] uses two different encoders of VAEs to map the in-completed images and other additional reference images in the dataset to the same low-dimensional manifold. 2) Some gradually modulate a random noise to the repaired image [20, 43]. For example, Liu *et al.* [20] first uses a pre-trained deterministic inpainting network to obtain an initially restored image and then inject it into the generator to modulate the generation process. Furthermore, Zhao *et al.* [43] further introduced the input noise of the GAN into the modulation process as well, proposing the co-modulation scheme. 3) Some [25, 32, 40] two-stage methods model the probability of coarse low-resolution structural priors in the first stage and use GAN in the second stage to generate rich texture details based on the previously obtained structural prior. Specifically, Peng *et al.* [25] uses VQ-VAE to separate and obtain discrete structural priors in the first stage. Wan *et al.* [32] uses a Transformer to model the probability distribution of masked tokens (i.e., missing pixels) by borrowing ideas from MLM (masked language model [5]). Yu *et al.* [40] combined the autoregressive model with MLM to further consider the dependencies between the missing pixels (masked tokens). 4) Some more recent diffusion-based methods [22, 28] restore images from random noise by iteration. They have the same problem of slow inference as autoregressive-based methods.

Normalizing Flow Normalizing flows have continuously made achievements in image generation tasks as they possess diverse generative capacity and exact likelihood computation. Normalizing flows are invertible generative models that learn a bijection function between the complex data distribution and simple predefined distribution. NICE [6], Real-NVP [7], Glow [16] are proposed in succession to promote the fitting ability of the flow models to the original data distribution. These efforts were later applied to audio generation [14, 26, 29, 35], image modeling [3, 9, 21, 31], and video prediction [17]. However, there is still a blank in employing a flow model in diverse image inpainting. The flexible nature of distribution mapping makes the flow-based model suitable for a diverse generation. Although VAEs and GANs-based methods work in diverse image generation tasks, they rely on deep-feature extraction and elaborate auxiliary modules to model the potential distribution of data. In addition, VAEs and GANs-based methods often suffer from mode collapse, posterior collapse, vanishing gradients, and instability. Based on the above analysis, we adopt a flow-based model to naturally model the distribution of coarse structural prior. As a result, the training process is more stable. In addition, conditional normalizing flow learns a conditionally reversible bijective function between a specific distribution and Gaussian distribution. Therefore, we can extend it to semantic transfer tasks.

3 Our Approach: Flow-Fill

Suppose we have an image I_{gt} originally from a dataset, it degraded by a mask M to become a masked image I_m comprising the observed/visible pixels. Di-

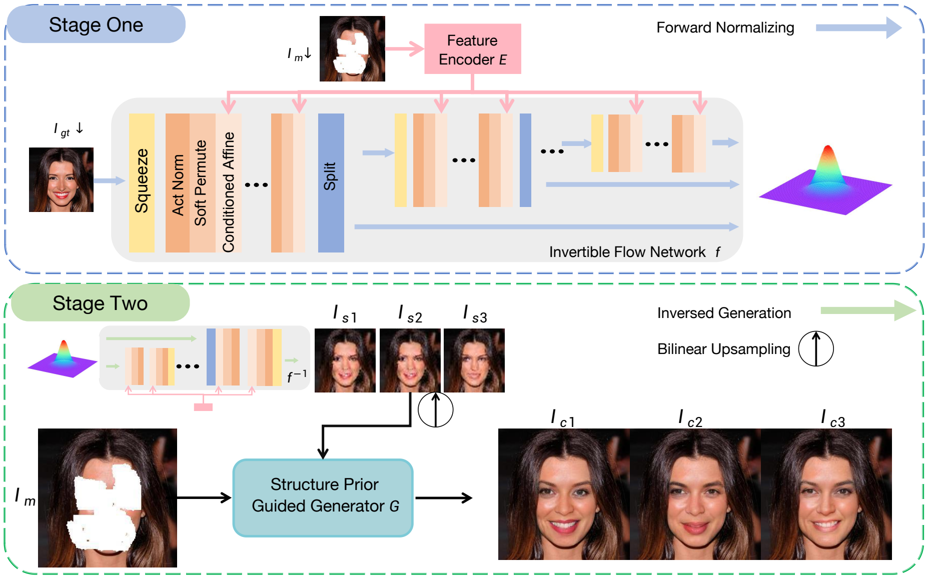

Fig. 2. Pipeline Overview. Our method is a two-stage diverse inpainting model. In the first stage, we adopt a conditional normalizing flow network to transform the conditional structural priors distribution $p(I_s|I_m)$ to a Gaussian density $p_Z(\mathbf{z})$. Therefore in the second stage, we can use the reverse mapping of the flow network to transform the random latent variables $\mathbf{z}$ to stochastic structural priors and then use another generator G to generate final texture-rich results guided by structural priors. I_{gt}: ground truth image, I_m: masked images, ↓: downscaled, I_s: structural prior, I_c: repaired image.

verse image inpainting refers to generating multiple and diverse visually realistic and semantically reasonable completed/repaired images $\{I_c\}$. We formulate this task to learn the conditional probability distribution $p(I_c|I_m)$ over completed images, sampling from which could produce diverse inpainting results corresponding to a given I_m. As depicted in Fig. 2, we adopt the two-stage pipeline that generates diverse structural priors I_s in the first stage and texture details in the second stage. Therefore $p(I_c|I_m)$ is decomposed into two parts $p(I_c|I_m) = p(I_c, I_s|I_m) = p(I_s|I_m) \cdot p(I_c|I_s, I_m)$, in which $p(I_s|I_m)$ is stochastic and $p(I_c|I_s, I_m)$ is deterministic. We adopt a conditional normalizing flow network f to model $p(I_s|I_m)$, and a deterministic inpainting network G to generate final results with rich textures: $I_c = G(I_s, I_m)$.

3.1 Normalizing the Conditonal Distribution of Structural Priors

Normalizing flows are invertible density estimation models that learn a bijection function f_θ between a complex data distribution p_X and a simple pre-defined prior p_Z. Given a data sample $\mathbf{x} \in X$, the core idea of normalizing flow is that, according to the *change-of-variable formula*, the probability density $p(\mathbf{x})$ can be explicitly computed as:

$$p(\mathbf{x}, \theta) = p_Z(f_\theta(\mathbf{x})) \left| \det \frac{\partial f_\theta}{\partial \mathbf{x}} \right| \tag{1}$$

Here the second factor is the volume-scaling determinant of Jacobian $\frac{\partial f_\theta}{\partial \mathbf{x}}$. This allows the *exact* maximum likelihood estimation (MLE) for $p(\mathbf{x})$. Given the conditions $\mathbf{c}$, to learn the conditional distribution $p(\mathbf{x}|\mathbf{c})$ using normalizing flow, Eq. 1 is extended to a conditional scheme:

$$p(\mathbf{x}|\mathbf{c}, \theta) = p_Z(f_\theta(\mathbf{x}; \mathbf{c})) \left| \det \frac{\partial f_\theta}{\partial \mathbf{x}}(\mathbf{x}; \mathbf{c}) \right| \tag{2}$$

In our work, f_θ is implemented by an invertible neural network stacked by T bijective layers $f_\theta = f_\theta^0 \circ f_\theta^1 \circ f_\theta^2 \circ ... \circ f_\theta^{T-1}$. The complex $\mathbf{x}$ is *normalized* to $\mathbf{z}$ as if it were a flow through a series of transformations, so such a model is called normalizing flow. Thanks to the natural bijective distribution mapping properties of normalizing flow models, we design the conditional normalizing flow network f_θ to directly map/normalize $p(I_s|I_m)$ to a simple distribution $p_Z(\mathbf{z})$ (e.g., Gaussian distribution). Therefore the conditional distribution $p(I_s|I_m, \theta)$ is implicitly defined by the reverse mapping: $p_Z(\mathbf{z}) \overset{f_\theta^{-1}}{\to} p(I_s|I_m, \theta)$. We can sample I_s by sampling $\mathbf{z} \sim p_Z(\mathbf{z})$ and then use the reverse mapping to get $I_s = f_\theta^{-1}(\mathbf{z}, I_m)$.

According to Eq. 2, the probability density $p(I_s|I_m)$ is computed as:

$$p(I_s|I_m, \theta) = p_Z(f_\theta(I_s; I_m)) \left| \det \frac{\partial f_\theta}{\partial I_s}(I_s; I_m) \right| \tag{3}$$

In practice, we first use another CNN network E_θ as an encoder to extract the given masked image's features $ft = E_\theta(I_m)$, and then inject ft to flow network f_θ. We simply downsample the ground-truth images to 64×64 to get $I_s = I_{gt} \downarrow$. Thus the first stage input masked image is also down-sampled and is denoted by $I_m \downarrow$. For network training, we calculate the negative log-likelihood (NLL) loss to apply maximum likelihood estimation for $p(I_s|I_m \downarrow)$:

$$\begin{aligned} \mathcal{L}(\theta; I_s, I_m) &= -\log p(I_s|E_\theta(I_m \downarrow), \theta) \\ &= -\log P_Z(f_\theta(I_s; ft)) - \log \left| \det \frac{\partial f_\theta}{\partial I_s}(I_s; ft) \right| \end{aligned} \tag{4}$$

3.2 Flow Network Design

To calculate the NLL loss (Eq. 4) and to generate inpainting results using the reverse mapping, each layer of our flow network f_θ needs to be carefully designed to calculate both the Jacobian determinant and the inverse cheaply. Our work is based on the widely used un-conditional Glow [16] architecture and its conditional extension [1, 21]. Here we briefly view the flow layers we borrow in our network and then describe the overall stage one network architecture.

Actnorm. Since the performance of Batch Normalizing is known to degrade for small per-GPU minibatch size, Glow [16] proposed the Actnorm as a substitute for Batch Normalizing. The scaling and bias of Actnorm are data-independent (only the initialization is data-dependent) and learnable, removing the impact of a small minibatch size. Due to memory constraints, We choose Actnorm to enable small minibatch size training.

Conditional Affine. The coupling layer was first proposed by [6]. It divides the input into two parts and keeps one part unchanged to make the inverse and Jacobian cheaply calculated. This also captures the dependency between the two parts by using the information from the remaining part to transform the other part. We use the conditional form [1, 21] of the affine coupling layer to inject masked images' features as conditions into the flow network:

$$\mathbf{h}_1^{t+1} = \mathbf{h}_1^t, \quad \mathbf{h}_2^{t+1} = \exp(s_\theta^t(\mathbf{h}_1^t; ft)) \cdot \mathbf{h}_2^t + t_\theta^t(\mathbf{h}_1^t; ft) \tag{5}$$

Where $(\mathbf{h}_1^t, \mathbf{h}_2^t) = \mathbf{h}^t$ is a partition of t-th layer's input activations. $s_\theta^t(\cdot)$ and $t_\theta^t(\cdot)$ are two arbitrary CNN networks that calculate the scaling and bias of $\mathbf{h}_2^t$. To further inject stronger conditional information, we use the affine injector layer proposed in [21] to apply the affine transformation on the full activations:

$$\mathbf{h}^{t+1} = \exp(s_\theta'^t(ft)) \cdot \mathbf{h}^t + t_\theta'^t(ft) \tag{6}$$

Here $s_\theta'^t(\cdot)$ and $t_\theta'^t(\cdot)$ are two other arbitrary networks. We implement Eq. 5 and Eq. 6 together to form our Conditional Affine layer.

Squeeze. We adopt the squeeze layer [16] as the downsampling operation. The squeeze layer reshapes every 2×2 adjacent pixel into the channel dimension. The flow network captures long-distance dependence by reducing the spatial resolution of activations.

Soft Permutation. Glow [16] proposed the invertible 1×1 convolution as the permutation operation on the channel dimension. It can be viewed as a linear transformation $\mathbf{h}^{t+1} = \mathbf{W}\mathbf{h}^t$, performed on each spatial position. Like [1], we set the weight matrix $\mathbf{W}$ as a fixed orthogonal matrix. This makes it easy to calculate both the Jacobian and the inverse of $\mathbf{W}$. Hence the training process is faster and more stable.

Overall Network Architecture of The First Stage. As depicted in Fig. 2, in the first stage, our Flow-Fill architecture consists of feature extraction network E_θ and flow network f_θ. The flow network f_θ is composed of L flow-blocks. Each flow block contains a Squeeze layer to reduce the spatial resolution of activations, followed by K conditional flow steps. Each conditional flow step consists of an Actnorm, a Soft Permutation, and a Conditional Affine. Except for the last flow-block, each flow-block contains a Split layer [16] at the end, dividing the output into two parts, one as the final output $\{\mathbf{z}_i\}_{i=1}^L$ and one as input for the next flow-block. Our work uses the reimplemented generator proposed in [39] as the masked images' feature extraction network E_θ. It is a coarse-to-fine inpainting generator with Gated Convolution. We select some intermediate feature maps of E_θ to concatenate together as the features ft of the input masked images to inject into the flow network. See the Appendix for more details about E_θ and feature selection.

3.3 Guided Texture Generation

The structural priors I_s obtained in the first stage are low-resolution and have no texture details. In the second stage, we upsample I_s and concatenate it with I_m as the input of another generator G_ϕ (parameterized by ϕ). G_ϕ is a deterministic inpainting network that generates the final repaired image with rich textures under the guidance of I_s. Thus the overall two-stage inpainting process is:

$$\begin{aligned} &\mathbf{z} \sim p_Z(\mathbf{z}) \\ &I_s = f_\theta^{-1}(\mathbf{z}; E_\theta(I_m \downarrow)) \\ &I_c = G_\phi(I_s \uparrow, I_m) \end{aligned} \tag{7}$$

Here ↓ indicates downsampling, and ↑ indicates upsampling. Follow [32], G_ϕ is composed of an encoder, decoder, and several residual blocks. The difference is that we replace all vanilla convolutions with gated convolution [39]. More details about the network architecture of G_ϕ are shown in the supplementary material.

For stage two training, we get I_s by downsampling I_{gt} and doing some degradation to maintain consistency with the results generated in the first stage. Specifically, we first calculate the latent variables $\mathbf{z} = f_\theta(I_{gt} \downarrow, I_m \downarrow)$ and replace some dimensions of $\mathbf{z}$ with random noise to get the slightly disturbed $\mathbf{z}'$. Then do the reconstruction by using the disturbed $\mathbf{z}'$: $I_s = f_\theta^{-1}(\mathbf{z}', I_m \downarrow)$. G_ϕ is optimized by adversarial training. Specifically, the adversarial loss is,

$$\mathcal{L}_{adv} = \mathbb{E}[\log 1 - D_\psi(I_c)] + \mathbb{E}[\log D_\psi(I_{gt})] \tag{8}$$

Here D_ψ is the discriminator parameterized by ψ. Alone with the L_1 reconstruction loss,

$$\mathcal{L}_{rec} = \mathbb{E}(\|I_c - I_{gt}\|_1) \tag{9}$$

G_ϕ and D_ψ are trained by solving the min-max optimization:

$$\min_G \max_D \mathcal{L}_{total} = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{rec} \tag{10}$$

In our experiments, λ_1 and λ_2 are empirically set at 0.1 and 1.0.

4 Experiments

4.1 Experimental Settings

Implementation Details. Our proposed method is implemented in PyTorch. We set the flow network architecture hyperparameters $L = 4$ and $K = 10$. The flow network f_θ and the feature extraction network E_θ were trained together for a total of 300k iterations with NLL loss (Eq. 4). For optimizer, we use Adam [15]

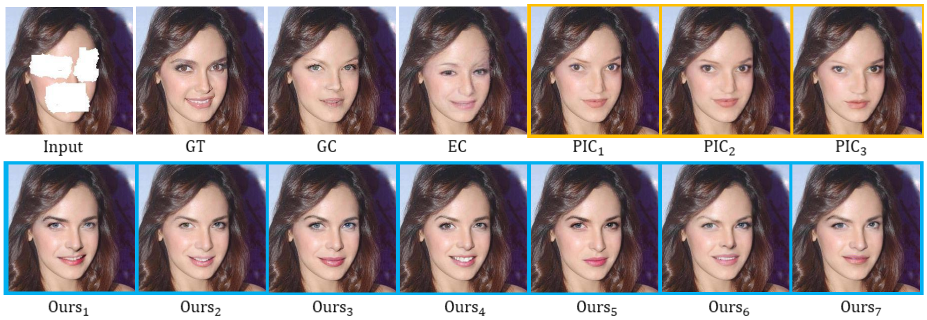

Fig. 3. Qualitative comparison with state-of-the-art methods on CelebA-HQ. The completion results of our method are with better quality and diversity.

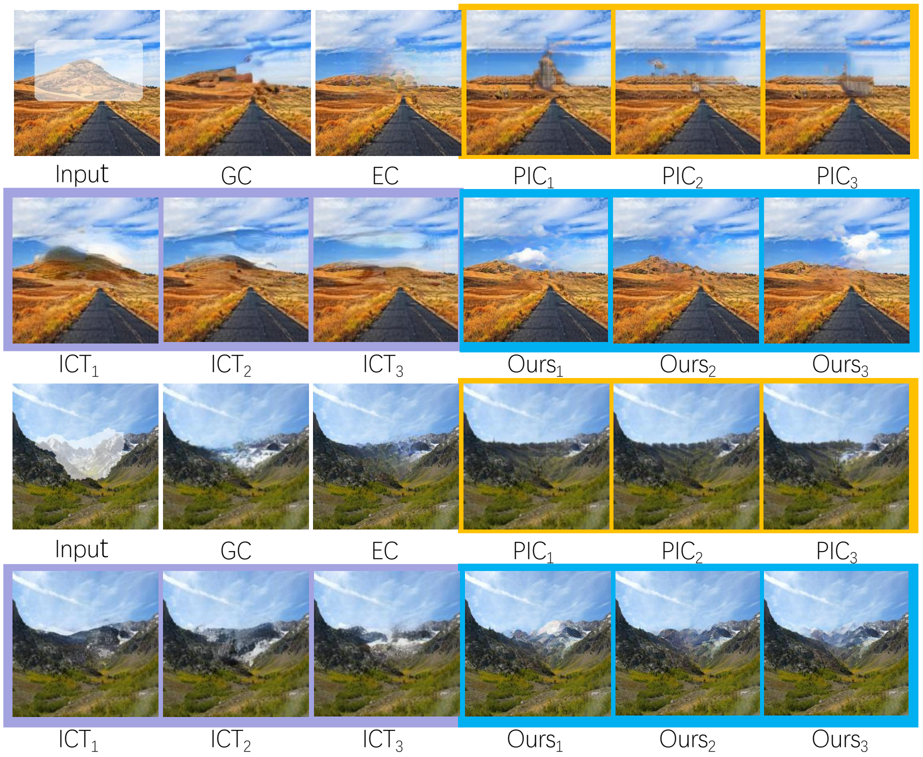

Fig. 4. Qualitative comparison with state-of-the-art methods on Places2. The completion results of our method are with better quality and diversity.

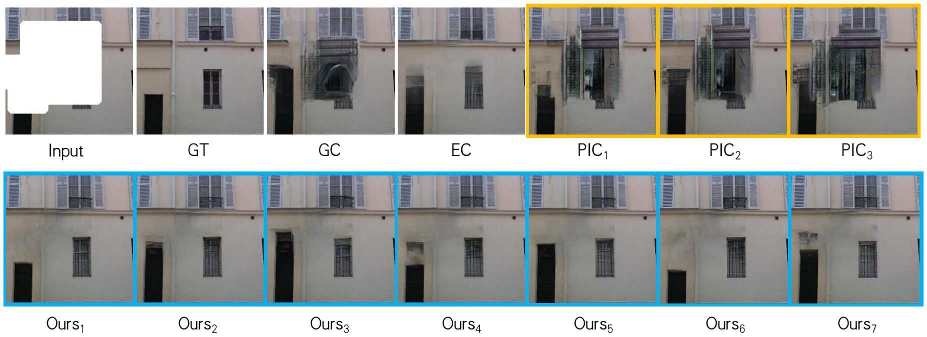

Fig. 5. Qualitative comparison with state-of-the-art methods on Paris StreetView. The completion results of our method are with better quality and diversity.

with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initialized as $5 \cdot 10^{-4}$ and halved at 50%, 75%, 90%, and 95% of the training iterations. To train the guided inpainting generator G_ϕ we use Adam [15] optimizer with fixed learning rate 1e-4, $\beta_1 = 0.0$ and $\beta_2 = 0.9$. The first stage of training takes about one day on a single NVIDIA(R) Tesla(R) V100 GPU with a minibatch size of 16. The second stage of training takes about four days on a single NVIDIA(R) Tesla(R) V100 GPU with a minibatch size of 8.

Datasets and Evaluation Metrics. We conduct our experiments on three datasets, including CelebA-HQ [13], Places2 [45], and Paris StreetView [24]. We follow the selection in [20] to produce the training, and validation sets for Places2. For CelebA-HQ and Paries StreetView, we keep the original training, validation, and testing split. All images are scaled to the resolution of 256×256 before inputting into the network. For non-square images in Pairs StreetView, random cropping is used. We train and evaluate our model with the irregular mask [18]. We adopt reconstruction-based metrics, including peak signal-to-noise ratio (PSNR), structural similarity (SSIM [33]), and mean ℓ_1 error, to measure the low-level similarity between the inpainting result and ground truth. Our goal is to generate diverse, visually realistic, and semantically reasonable inpainting results that are unnecessarily similar to ground truth. Therefore, we further use Fréchet Inception Distance (FID [11]) as perceptual quality metrics, which are consistent with human judgment.

4.2 Performance Evaluation

We compare our method with the following state-of-the-art inpainting algorithms: GC [39], EC [23], PIC [44], ICT [32], and BAT [40]. GC and EC are single-solution methods. PIC, ICT, and BAT are multiple-solution methods. The performance of the compared methods was acquired by using the publicly available pre-trained models or implementation codes.

Table 1. Quantitative comparison over CelebA-HQ and Places2 datasets. For each metric, the best score is highlighted in **bold**, and the second-best score is highlighted in underline.

Methods	**Dataset**	**FID↓**			**ℓ_1(%) ↓**			**PSNR↑**			**SSIM↑**		
		20-40%	40-60%	Random	20-40%	40-60%	Random	20-40%	40-60%	Random	20-40%	40-60%	Random
EC [23]	CelebA-HQ [13]	9.06	16.45	12.46	2.19	4.71	3.40	26.60	22.14	24.45	0.923	0.823	0.877
GC [39]		14.12	22.80	18.10	2.70	5.19	3.88	25.17	21.21	23.32	0.907	0.805	0.858
PIC [44]		10.21	18.92	14.12	2.50	5.65	4.00	25.92	20.82	23.46	0.919	0.780	0.852
BAT [40]		**6.32**	**12.50**	**9.33**	1.91	4.57	3.18	27.82	22.40	25.21	**0.944**	0.834	0.890
Ours		7.75	14.91	11.29	**1.42**	**3.31**	**2.34**	**28.06**	**23.10**	**25.60**	**0.944**	**0.856**	**0.895**
EC [23]	Places2 [45]	25.64	39.27	30.13	2.20	4.38	2.93	26.52	22.23	25.51	0.880	0.731	0.831
GC [39]		24.76	39.02	29.98	2.15	4.40	2.80	26.53	21.19	25.69	0.881	0.729	0.834
PIC [44]		26.39	49.09	33.47	2.36	5.07	3.15	26.10	21.50	25.04	0.865	0.680	0.806
ICT [32]		21.60	33.85	25.42	2.44	4.31	2.67	26.50	22.22	25.79	0.880	0.724	0.832
BAT [40]		**17.78**	**32.55**	**22.16**	2.15	4.64	2.84	26.47	21.74	25.69	0.879	0.704	0.826
Ours		19.03	33.26	25.40	**1.87**	**3.92**	**2.47**	**26.76**	**22.38**	**25.84**	**0.892**	**0.799**	**0.847**

Qualitative Comparisons. We conduct qualitative comparisons over CelebA-HQ [13], Places2 [45] and Paris StreetView [24] datasets. For CelebA-HQ and Paris StreetView, our mthdod is compared with GC [39], EC [23], and PIC [44]. For Places2, our method is compared with GC, EC, PIC, and ICT [32]. All results are the direct output of the model without any post-processing.

Fig. 3 shows the results on CelebA-HQ [13]. GC [39], and EC [23] generate generally reasonable content, but with some artifacts, they can only generate a single result. The results of PIC [44] have better fidelity than GC and EC but limited diversity. Compared to these methods, ours is superior in both photorealism and diversity. Fig. 5 shows the case of large missing areas on Paris StreetView [24]. This time GC and PIC generate incongruent content with the visible region. EC is much better but lacks sharp details. Again, ours have the best fidelity and diversity. Results on nature scenery images are shown in Fig. 4. EC's results have pronounced artifacts, while GC's are much better. The PIC's results look more realistic than ICT's [32], but the diversity is not as good as ICT's. Only ours look both natural and varied.

Quantitative Comparisons. We quantitatively compare our method with other deterministic and non-deterministic inpainting methods on ClebabA-HQ [13] and Places2 [45]. All tests use irregular masks [18], categorized according to the mask ratios. Here 'Random' indicates that the mask from this category has a mask ratio from 20% to 60%. Unlike PIC [44], which unitizes its discriminator to sort the results, our method uses all random samples without any selection to better evaluate our model's average performance. As shown in Tab. 1, Ours achieve the best reconstruction scores and have comparable perceptual quality.

4.3 Region-specific Semantic Transfer

Given a masked region as a condition, Flow-Fill can build a bijection between semantic contents in this area and latent variable space. With this property, we can directly calculate the latent variables **z** that contain the target image's semantic information and fill it in different source images by reverse inference

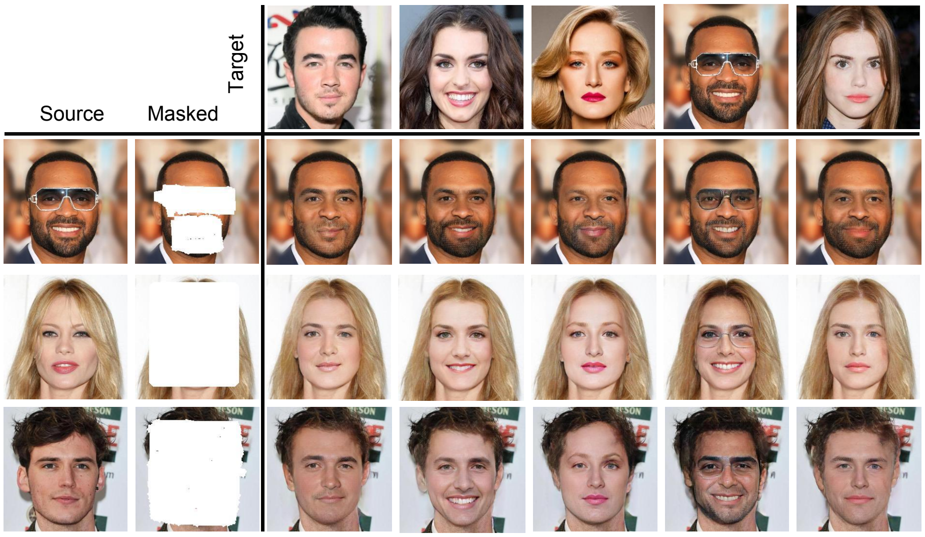

Fig. 6. Specific-region semantic transfer results. We use the target image to compute the latent variable in the first stage and thus obtain an inpainting result similar to the target style (eyes, eyebrows, mouth, glasses, etc.).

(no need to retrain the inpainting model). Note that in the semantic transfer task, we use $\mathbf{z}$ inverted from target images instead of randomly sampled.

Specifically, given a target image I_t and a masked source image I_m (masked with mask M), we first normalizing it to the latent variables $\mathbf{z} = f_\theta(I_t \downarrow; E_\theta((I_t \cdot M) \downarrow))$. Then we generate the structural prior I_s by inversed generation: $I_s = f_\theta^{-1}(\mathbf{z}; E_\theta(I_m \downarrow))$. Finally, the result with rich texture is generated: $I_c = G_\phi(I_s \uparrow, I_m)$. Experimental results are shown in Fig. 6. By selecting the target image, we can control the hairstyle, lip color, whether to open the mouth, whether to wear glasses, etc., of the inpainting result.

Table 2. Quantitative comparison on Paris StreetView dataset. The best score is highlighted in **blod**.

Method	Mask Ratio	PSNR ↑	SSIM ↑	ℓ_1(%) ↓	FID ↓	LPIPS ↑
PIC [44]	20% – 40%	24.80	0.817	3.43	56.83	0.046
BAT [40]		26.52	0.864	3.43	36.19	0.076
Ours		**26.87**	**0.897**	**1.95**	**33.98**	**0.078**
PIC [44]	40% – 60%	20.12	0.570	7.47	90.91	0.127
BAT [40]		21.89	0.678	5.83	**64.20**	0.147
Ours		**22.35**	**0.798**	**3.99**	65.86	**0.151**
PIC [44]	Random	22.97	0.718	4.94	72.16	0.082
BAT [40]		24.50	0.786	3.96	**48.19**	0.106
Ours		**24.58**	**0.849**	**2.90**	50.18	**0.109**

4.4 Analysis

Diversity. Following [44, 46], we utilize the LPIPS distance [41] to measure the diversity score. LPIPS is computed based on the in-depth features of the VGG [30] model pre-trained on ImageNet. Specifically, we randomly sampled five output pairs for each masked input image to calculate the average score. Because results with high variability are likely to be unreasonable, we measured PSNR, SSIM [33], mean ℓ_1 error, and FID [11] simultaneously. Tab 2 shows the results. Our model achieves the best diversity while maintaining high fidelity in all cases.

Computational time. We randomly selected 200 images on the test set of Places2 [45] and calculated the average computational time per image. As shown in Tab. 3, our method achieves a real-time inference speed approximately 87 times faster than autoregressive-based models and 142 times faster than diffusion-based models. All tests were performed on an NVIDIA(R) GeForce RTX 3090 GPU.

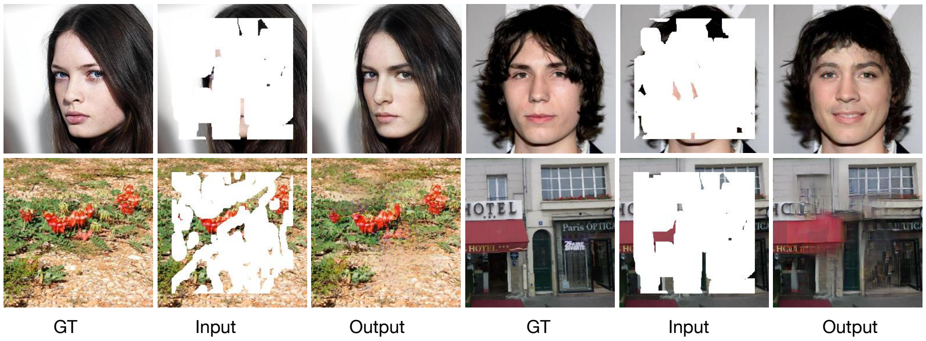

Fig. 7. Inpainting results of our method. We achieve the first flow-based large missing region complementation.

Table 3. Comparison of inference speed. Rows and columns correspond to different masked areas and methods respectively.

	BAT [40]	ICT [32]	Palette [28]	Ours
20-40%	11.33s	9.40s	27.01s	**0.18**s
40-60%	22.21s	15.53s	27.23s	**0.19**s
random	16.60s	13.03s	27.13s	**0.19**s

Table 4. Ablation studies on CelebA-HQ. The mask ratio is 20-60%. The best score is highlighted in **blod**.

First stage resolution	FID↓	ℓ_1(%) ↓	PSNR↑	SSIM↑
32 × 32	12.44	2.54	25.04	0.888
48 × 48	11.86	2.40	25.46	0.892
64 × 64	**11.29**	**2.34**	**25.60**	**0.895**
96 × 96	13.19	2.61	24.95	0.883
256 × 256	41.94	3.86	22.31	0.844

Ablation study on first stage resolution. We ablate the resolution for the first stage. As shown in Tab. 4, normalizing flow is difficult to generate high-quality images with high resolution. We use normalizing flow to complement a low-resolution (64×64) coarse result in the first stage and use GAN to generate a high-resolution visual pleasing result in the second stage. Thus we circumvent the difficulties of flow models in generation and achieve the first flow-based large missing region complementation. Some inpainting examples for large regions of missing images are shown in Fig. 7.

Searching for flow network structure hyperparameters. Our flow network consists of L flow-blocks, and each flow-blocks consists of K flow-steps. In general, the larger the L, K, the better the model performance. To reduce the model size while maintaining a good inpainting performance, we form this problem to a constrained optimization problem:

$$\begin{aligned} L^*, K^* &= \arg\max_{L,K} \; \mathcal{Q}(L,K) + \lambda \mathcal{T}(L,K) \\ s.t. \quad & 12 \leq L + K \leq 15 \end{aligned} \tag{11}$$

Where $\mathcal{Q}$, $\mathcal{T}$ denotes the inpainting performance and network size function with respect to L, K. After a rough grid search, the best L and K are 4 and 10.

5 Conclusion

We propose a novelty two-stage image inpainting framework named Flow-Fill, which can directly estimate the joint probability density of the missing regions without reasoning pixel by pixel. Hence it achieves real-time inference speed and eliminates discretization assumptions. In addition, as a flow-based model, Flow-Fill can directly calculate the latent variables containing the specified semantic information, which allows us to control the reverse inpainting process to a certain extent. Experiments on benchmark datasets qualitatively and quantitatively verify that Flow-Fill achieves superior diversity and fidelity in image inpainting.

Acknowledgement. This work was supported by SZSTC Grant No. JCYJ20190809172201639 and WDZC20200820200655001, Shenzhen Key Laboratory. ZDSYS20210623092001004. We thank Yunpeng Bai for helpful discussion.

References

1. Ardizzone, L., Lüth, C., Kruse, J., Rother, C., Köthe, U.: Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392 (2019) 3.2, 3.2
2. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 417–424 (2000) 2
3. Chen, H.J., Hui, K.M., Wang, S.Y., Tsao, L.W., Shuai, H.H., Cheng, W.H.: Beautyglow: On-demand makeup transfer framework with reversible generative network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10042–10050 (2019) 2
4. Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P.: Image melding: Combining inconsistent images using patch-based synthesis. ACM Transactions on graphics (TOG) **31**(4), 1–10 (2012) 2
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 2
6. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014) 2, 3.2
7. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016) 2
8. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 341–346 (2001) 2
9. Grover, A., Chute, C., Shu, R., Cao, Z., Ermon, S.: Alignflow: Cycle consistent learning from multiple domains via normalizing flows. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4028–4035 (2020) 2
10. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Transactions on Graphics (ToG) **26**(3), 4–es (2007) 2
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 4.1, 4.4
12. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017) 1, 2
13. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017) 4.1, 1, 4.2, 4.2
14. Kim, S., Lee, S.g., Song, J., Kim, J., Yoon, S.: Flowavenet: A generative flow for raw audio. arXiv preprint arXiv:1811.02155 (2018) 2
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 4.1
16. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018) 2, 3.2, 3.2
17. Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., Dinh, L., Kingma, D.: Videoflow: A flow-based generative model for video. arXiv preprint arXiv:1903.01434 **2**(5) (2019) 2
18. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European conference on computer vision (ECCV). pp. 85–100 (2018) 1, 2, 4.1, 4.2

19. Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: European Conference on Computer Vision. pp. 725–741. Springer (2020) 1, 2
20. Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J.: Pd-gan: Probabilistic diverse gan for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9371–9381 (2021) 1, 2, 4.1
21. Lugmayr, A., Danelljan, M., Gool, L.V., Timofte, R.: Srflow: Learning the super-resolution space with normalizing flow. In: European conference on computer vision. pp. 715–732. Springer (2020) 2, 3.2, 3.2
22. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022) 1, 2
23. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019) 1, 2, 4.2, 1, 4.2
24. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016) 1, 2, 4.1, 4.2
25. Peng, J., Liu, D., Xu, S., Li, H.: Generating diverse structure for image inpainting with hierarchical vq-vae. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10775–10784 (2021) 1, 2
26. Prenger, R., Valle, R., Catanzaro, B.: Waveglow: A flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3617–3621. IEEE (2019) 2
27. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 181–190 (2019) 1, 2
28. Saharia, C., Chan, W., Chang, H., Lee, C.A., Ho, J., Salimans, T., Fleet, D.J., Norouzi, M.: Palette: Image-to-image diffusion models. arXiv preprint arXiv:2111.05826 (2021) 1, 2, 3
29. Serrà, J., Pascual, S., Segura Perales, C.: Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. Advances in Neural Information Processing Systems **32** (2019) 2
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 4.4
31. Sun, H., Mehta, R., Zhou, H.H., Huang, Z., Johnson, S.C., Prabhakaran, V., Singh, V.: Dual-glow: Conditional flow-based generative model for modality transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10611–10620 (2019) 2
32. Wan, Z., Zhang, J., Chen, D., Liao, J.: High-fidelity pluralistic image completion with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4692–4701 (2021) 1, 1, 2, 3.3, 4.2, 1, 4.2, 3
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004) 4.1, 4.4
34. Xu, S., Liu, D., Xiong, Z.: E2i: Generative inpainting from edge to image. IEEE Transactions on Circuits and Systems for Video Technology **31**(4), 1308–1322 (2020) 1, 2

35. Yamaguchi, M., Koizumi, Y., Harada, N.: Adaflow: Domain-adaptive density estimator with application to anomaly detection and unpaired cross-domain translation. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3647–3651. IEEE (2019) 2
36. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: Proceedings of the European conference on computer vision (ECCV). pp. 1–17 (2018) 1, 2
37. Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7508–7517 (2020) 1, 2
38. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018) 1, 2
39. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4471–4480 (2019) 1, 2, 3.2, 3.3, 4.2, 1, 4.2
40. Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., Miao, C.: Diverse image inpainting with bidirectional and autoregressive transformers. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 69–78 (2021) 1, 1, 2, 4.2, 1, 2, 3
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 4.4
42. Zhao, L., Mo, Q., Lin, S., Wang, Z., Zuo, Z., Chen, H., Xing, W., Lu, D.: Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5741–5750 (2020) 1, 2
43. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. arXiv preprint arXiv:2103.10428 (2021) 1, 2
44. Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1438–1447 (2019) 1, 2, 4.2, 1, 4.2, 4.2, 2, 4.4
45. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017) 4.1, 1, 4.2, 4.2, 4.4
46. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. Advances in neural information processing systems **30** (2017) 4.4