Reduce Information Loss in Transformers for Pluralistic Image Inpainting

Qiankun Liu1* Zhentao Tan1 Dongdong Chen2 Qi Chu† Xiyang Dai2 Yinpeng Chen2 Mengchen Liu2 Lu Yuan2 Nenghai Yu1
1CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China
2Microsoft Cloud + AI
{liuqk3, tzt}@mail.ustc.edu.cn, {gchu, ynh}@ustc.edu.cn
cddlyf@gmail.com, {xiyang.dai, yiche, mengcliu, luyuan}@microsoft.com

Abstract

Transformers have achieved great success in pluralistic image inpainting recently. However, we find existing transformer based solutions regard each pixel as a token, thus suffer from information loss issue from two aspects: 1) They downsample the input image into much lower resolutions for efficiency consideration, incurring information loss and extra misalignment for the boundaries of masked regions. 2) They quantize 2563 RGB pixels to a small number (such as 512) of quantized pixels. The indices of quantized pixels are used as tokens for the inputs and prediction targets of transformer. Although an extra CNN network is used to upsample and refine the low-resolution results, it is difficult to retrieve the lost information back. To keep input information as much as possible, we propose a new transformer based framework “PUT”. Specifically, to avoid input downsampling while maintaining the computation efficiency, we design a patch-based auto-encoder P-VQVAE, where the encoder converts the masked image into non-overlapped patch tokens and the decoder recovers the masked regions from the inpainted tokens while keeping the unmasked regions unchanged. To eliminate the information loss caused by quantization, an Un-Quantized Transformer (UQ-Transformer) is applied, which directly takes the features from P-VQVAE encoder as input without quantization and regards the quantized tokens only as prediction targets. Extensive experiments show that PUT greatly outperforms state-of-the-art methods on image fidelity, especially for large masked regions and complex large-scale datasets.

1. Introduction

Image inpainting, which focuses on filling meaningful and plausible contents in missing regions for the damaged images, has always been a hot topic in computer vision areas and widely used in various applications [1, 35, 39, 42, 47, 48, 56]. Traditional methods [1, 3, 9] based on texture matching can handle simple cases very well but struggle for complex natural images. In the last several years, benefiting from development of CNNs, tremendous success [28, 30, 34, 54] has been achieved by learning on large-scale datasets. However, due to the inherent properties of CNNs, i.e., local inductive bias and spatial-invariant kernels, such methods still do not perform well in understanding global structure and inpainting large masked/missing regions.

Recently, transformers have demonstrated their power in various vision tasks [4–8, 13, 16, 37, 50], thanks to their capability of modeling long-term relationship. Some recent works [49] also attempt to apply transformers for pluralistic image inpainting, and have achieved remarkable success in better diversity and large region inpainting quality. As shown in the top row of Figure 1, they follow the similar design: 1) Downsample the input image into lower resolutions and quantize the pixels; 2) Use the transformer to re-
cover the masked pixels by regarding each quantized pixel as the token; 3) Upsample and refine the low-resolution result by feeding it together with the original input image into an extra CNN network.

In this paper, we argue that using the above pixel-based token makes existing transformer based solutions suffer from the information loss issue from two aspects: 1) “Low resolution”. To avoid high computation complexity of transformer, the input image is downsampled into much lower resolution to reduce the input token number, which not only incurs information loss but also introduces misalignment for the boundaries of masked regions when upsampled back to the original resolution. 2) “Quantization”. To constrain the prediction within a small space, the huge amount \((256^3)\) of RGB pixels are quantized into much less \((\text{such as 512})\) quantized pixels through clustering. The indices of quantized pixels are used as discrete tokens both for the input and prediction target of transformer. Such practice would further result in the information loss.

To mitigate these issues, we propose a new transformer based framework PUT, which can reduce the information loss as much as possible. As shown in the bottom row of Figure 1, the original high-resolution input image is directly fed into a patch based encoder without any downsampling and the transformer directly takes the features from the encoder as input without any quantization. Specifically, PUT contains two key designs: Patch-based Vector Quantized Variational Auto-Encoder (“P-VQVAE”, Section 3.1) and Un-Quantized Transformer (“UQ-Transformer”, Section 3.2). P-VQVAE is a specially designed patch autoencoder: 1) Its encoder converts each image patch into the latent feature in a non-overlapped way, where the non-overlapped design is to avoid the disturbance between masked regions and unmasked regions; 2) As the prediction space of UQ-Transformer, a dual-codebook is built for patch feature tokenization, where masked patches and unmasked patches are separately represented by different codebooks; 3) The decoder in P-VQVAE not only recovers the masked image regions from the inpainted tokens but also maintains unmasked regions unchanged. For UQ-Transformer, it utilizes the quantized tokens of unmasked patches as the prediction targets for masked patches, but takes the un-quantized feature vectors from the encoder as input. Compared to taking the quantized tokens as input, this design can avoid the information loss and help UQ-Transformer make more accurate predictions.

To demonstrate the superiority, we conduct extensive experiments on FFHQ [25], Places2 [61] and ImageNet [11]. The results show that our method outperforms CNN based pluralistic inpainting methods by a large margin on different evaluation metrics. Benefiting from less information loss, our method also achieves much higher fidelity than existing transformer based solutions, especially for large region inpainting and complex large-scale datasets.

2. Related Work

Auto-Encoders. Auto-encoders [21] is a kind of artificial neural network in semi-supervised and unsupervised learning. Among its subclasses, variational auto-encoders (VAE) [12, 27] is widely used for image synthesis tasks [41, 43] as a generative model. It can be trained in self-supervised strategy and generate diverse images through decoder with latent space sampling or autoregressive models [33, 45]. Later, vector quantized variational auto-encoder (VQ-VAE) [44] is proposed for discrete representation learning to circumvent issues of “posterior collapse”, and further developed by VQ-VAE-2 [38]. Recently, based on the similar quantization mechanism with VQ-VAE, VAGAN [16] and dVAE [37] are proposed for conditional image generation through transformers [46], while PeCo [14] trains a perceptual vision tokenizer for vision transformer BERT-pretraining [51]. Different from previous methods, the proposed “P-VQVAE”, which contains a non-overlapped patch encoder, a dual-codebook and a multi-scale guided decoder, is dedicated for image inpainting.

Visual Transformers. Thanks to the capability of long range relationship modeling, transformers have been widely used in different vision tasks, such as object detection [4, 6], image synthesis [5, 16, 37], object tracking [7, 50] and image inpainting [49]. Specifically, the autoregressive inference mechanism is naturally suitable for image synthesis related tasks, which can bring diverse results while guarantee the quality of the synthesized images [5, 16, 37, 49]. In this paper, we take full advantage of transformer and propose to replace discrete tokens with continuous feature vectors to avoid the information loss.

Image Inpainting. According to the diversity of inpainted images, there are two different types of definition for image inpainting task: deterministic image inpainting and pluralistic image inpainting. Most of the traditional methods, whether diffusion-based methods [2, 15] or path-based methods [1, 10, 19], can only generate single result for each input and may be failed while meeting large-area of miss-
ing pixels. Later, some CNN based methods [23, 28, 30, 32, 34, 54] are proposed to ensure consistency of the semantic content of the inpainted images, but still ignore the diversity of results. To generate several diverse results for each masked image, some CNN based [58, 60] and transformer based [49] methods have emerged recently. Among them, transformer based methods [49] show their merits in both quality and diversity than CNN based methods. However, their unreasonable design, like downsampling of input image and quantization of transformer inputs, results in the serious information loss issue. Thus, we propose a novel framework, PUT, which maximizes the input information to achieve better synthesis results.

3. Method

The proposed method mainly consists of a Patch-based Vector Quantized Variational Auto-Encoder (P-VQVAE) and an Un-Quantized transformer (UQ-Transformer). The overview of our method is shown in Figure 2. Let \( x \in \mathbb{R}^{H \times W \times 3} \) be an image and \( m \in \{0, 1\}^{H \times W \times 1} \) be the mask denoting whether a region needs to be inpainted (with value 0) or not (with value 1). \( H \) and \( W \) are the spatial resolution. The image \( \hat{x} = x \otimes m \) is the masked image that contains missing pixels, where \( \otimes \) is the elementwise multiplication. The masked image \( \hat{x} \) is first fed into the encoder of P-VQVAE to get the patch based feature vectors. Then UQ-Transformer takes the feature vectors as input and predicts the tokens (i.e., indices) of latent vectors in a codebook for masked regions. Finally, the retrieved latent vectors are used as the quantized vectors for patches and fed into the decoder of P-VQVAE to reconstruct the inpainted image.

3.1. P-VQVAE

To avoid the information loss from input downsampling while maintaining the computation efficiency for the transformer, we utilize the merits of auto-encoder to replace the downsampled pixels with the features from encoder. Compared to the downsampled pixels, the features from encoder can have the same low-resolution for efficiency while contains more information for reconstruction. Considering the task of image inpainting, we specially design P-VQVAE, which contains a patch-based encoder, a dual-codebook and a multi-scale guided decoder.

**Patch-based Encoder.** Conventional CNN based encoders process the input image with several convolution kernels in a sliding window manner, which are unsuitable for image inpainting since they would introduce disturbance between masked and unmasked regions. Thus, the encoder of P-VQVAE (denoted as P-Enc) is designed to process input image by several linear layers in a non-overlapped patch manner. Specifically, the masked image \( \hat{x} \) is firstly partitioned into \( \frac{H}{r} \times \frac{W}{r} \) non-overlapped patches, where \( r \) is the spatial size of patches and is set to 8 by default. For a patch, we call it a masked patch if it contains any missing pixels, otherwise unmasked patch. Each patch is flattened and then mapped into a feature vector. Formally, all feature vectors are denoted as \( \hat{f} = \mathcal{E}(\hat{x}) \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C} \), where \( C \) (set to 256 by default) is the dimensionality of feature vectors and \( \mathcal{E}(\cdot) \) is the encoder function.

**Dual-Codebook for Vector Quantization.** Following the works in [16, 38, 44], the feature vectors from encoder are quantized into discrete tokens with the latent vectors in the learnable codebook. By contrast, we design a dual-codebook (denote as D-Codes) for vector quantization, which is more suitable for image inpainting. In D-Codes, the latent vectors are divided into two parts, denoted as \( e \in \mathbb{R}^{K \times C} \) and \( e' \in \mathbb{R}^{K' \times C} \), which are responsible for feature vectors that mapped from unmasked and masked patches respectively. \( K \) and \( K' \) are the number of latent vectors. Let \( m^\downarrow \in \{0, 1\}^{\frac{H}{r} \times \frac{W}{r} \times 1} \) be the indicator mask that indicates whether a patch is a masked (with value 0) or unmasked (with value 1) patch. The feature vector \( \hat{f}_{i,j} \) is quantized as:

\[
\begin{align*}
    \sigma_k & \text{ where } k = \arg\min_d \| \hat{f}_{i,j} \odot e_d \|_2, \text{ if } m^\downarrow_{i,j} = 1, \\
    \sigma'_{k'} & \text{ where } k' = \arg\min_t \| \hat{f}_{i,j} \odot e'_{t} \|_2, \text{ else},
\end{align*}
\]

(1)
where $\ominus$ denotes the operation of elementwise subtraction. Let $\hat{e} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}$ and $\hat{t} = \mathcal{I}(\hat{f}, e, e', m^t) \in \mathbb{N}^{\frac{H}{2} \times \frac{W}{2}}$ be the quantized vectors and tokens for $\hat{f}$, where $\mathcal{I}(\cdot, \cdot, \cdot)$ denotes the function that gets tokens for its first argument and it can be simply implemented by getting the indices of all quantized vectors in $\hat{e}$. The dual-codebook helps P-Enc learn more discriminative feature vectors for masked and unmasked patches since they are quantized and represented with different codebooks, which further disenchants transformer about the masked and unmasked patches to predict more reasonable results for masked patches.

**Multi-Scale Guided Decoder.** For image inpainting task, an indisputable fact is that the unmasked regions should be kept unchanged. To this end, we design a multi-scale guided decoder, denoted as MSG-Dec, to construct the inpainted image $\hat{x}^l$ by referencing the input masked image $\hat{x}$. Let $\hat{t}^l$ be the inpainted tokens by transformer (Ref. Figure 2 and Section 3.3) and $\hat{e}^l$ be the retrieved quantized vectors from codebook based on $\hat{t}^l$. The construction procedure is formulated as:

$$\hat{x}^l = \mathcal{D}(\hat{e}^l, m, \hat{x}),$$  

(2)

where $\mathcal{D}(\cdot, \cdot, \cdot)$ is the decoder function. The decoder consists of two branches: a main branch which starts with the quantized vectors $\hat{e}^l$ and uses several deconvolution layers to generate inpainted images and a reference branch which extracts multi-scale feature maps (with spatial sizes $\frac{H}{2^l} \times \frac{W}{2^l}$, $0 \leq l \leq \log_2 r$) from the masked image $\hat{x}$. The features from the reference branch are fused to the features with the same scale from the main branch through a Mask Guided Addition (MGA) module as:

$$\hat{e}^{l,l-1} = \text{Deconv}((1 - m^{l,l}) \otimes \hat{e}^{l,l} + m^{l,l} \otimes \hat{t}^{R,l}),$$  

(3)

where $\hat{e}^{l,l}$ and $\hat{t}^{R,l}$ are the features with spatial size $\frac{H}{2^l} \times \frac{W}{2^l}$ from the main branch and the reference branch respectively. $m^{l,l}$ is the indicator mask obtained from $m$ for corresponding spatial size.

**Training of P-VQVAE.** To avoid the decoder learning to reconstruct input image $\hat{x}$ only from the reference image, we get the reference image by randomly erasing some pixels in $\hat{x}$ with another mask $m'$ (see in Figure 3). Let $\hat{x}^R = \mathcal{D}(\hat{e}, m \otimes m', \hat{x} \otimes m')$ be the reconstructed image. In our design, the unmasked pixels in the reference image will be utilized to recover the corresponding pixels in $\hat{x}^R$, while the latent vectors in codebook $e'$ and $e$ will be used to recover the pixels in $\hat{x}^R$ masked by $m$ and the remaining pixels respectively. The loss for training P-VQVAE is:

$$L_{vae} = \mathcal{L}_{rec}(\hat{x}, \hat{x}^R) + \| \mathbf{sg}[\hat{f}] \otimes \hat{e} \|_2^2 + \beta \| \mathbf{sg}[\hat{e}] \otimes \hat{f} \|_2^2,$$  

(4)

where $\mathbf{sg}[\cdot]$ refers to a stop-gradient operation that blocks gradients from flowing into its argument. $\beta$ is the weight for balance and is set to 0.25. $\mathcal{L}_{rec}(\cdot, \cdot)$ is the function to measure the difference between inputted and reconstructed images, including the L1 loss between the pixel values in two images and the gradients of two images, the adversarial loss [18] obtained by a discriminator network, as well as the perceptual loss [24] and the style loss [17] between the two images. Following [38, 44], the second term in Eq. (4) is replaced by Exponential Moving Average (EMA) to optimize the vectors in D-Codes. More details about the training of P-VQVAE could be found in the supplementary material.

**3.2. UQ-Transformer.**

In existing transformers for image inpainting [49] and synthesis [16, 37], the quantized discrete tokens are used as both the inputs and prediction targets. Given such discrete tokens, transformers suffer from the severe information loss issue, which is harmful to their prediction. In contrast, to take full advantage of feature vectors $\hat{f}$ from the encoder of P-VQVAE, our UQ-Transformer directly takes them as the inputs and predicts the discrete tokens for masked patches.

Specifically, $\hat{f}$ is firstly mapped by a linear layer and then added with extra learnable position embeddings for the encoding of spatial information. Finally, following [36], the feature vectors are flattened along spatial dimension to get the final input for the subsequent several transformer blocks. The output of the last transformer block is further projected to the distribution over $K$ latent vectors in codebook $e$ with a linear layer and a softmax function. We formulate the above procedure as $\hat{p} = T(\hat{f}) \in [0, 1]^{\frac{H}{2} \times \frac{W}{2} \times K}$, where $T(\cdot)$ refers the UQ-Transformer function.

**Training of UQ-Transformer.** Given a masked image $\hat{x}$, the distribution of its corresponding inpainted tokens over $K$ latent vectors can be obtained with the pre-trained P-VQVAE and UQ-Transformer $\hat{p} = T(\mathcal{E}(\hat{x}))$. The ground-truth tokens for $x$ is $t = \mathcal{I}(\mathcal{E}(x), e, e', O(m^x))$ (Ref. Section 3.1), where $O(\cdot)$ sets all values in the given argument to 1. UQ-Transformer is trained with cross-entropy loss by fixing P-VQVAE:

$$L_{trans} = -\frac{1}{\sum_{i,j} (1 - m^t_{i,j}) \mathbf{log} \hat{p}_{i,j, t_{i,j}}}.$$  

(5)

In order to make the training stage consistent with inference stage, where only the quantized vectors can be obtained for masked regions, we randomly quantize the feature vectors in $\mathcal{E}(\hat{x})$ to the latent vectors in codebook with probability 0.3 before feeding it to UQ-Transformer.

**3.3. Sampling Strategy for Image Inpaining**

For the production of diverse results, the tokens for masked patches ($m^t_{i,j} = 0$) are iteratively sampled with
Gibbs sampling. Specifically, in each iteration, we first select the patch with the maximum predicted probability among the remaining masked patches. Then the token for the selected patch is sampled from the top-$K$ predicted elements. Finally, the corresponding latent vector for the sampled token is retrieved to replace the feature vector of the selected patch before feeding UQ-Transformer for the next iteration. After sampling the tokens for all masked patches, we can get all quantized vectors $\hat{e}^I$ with the inpainted tokens $\hat{t}^I$, and the inpainted image can be constructed using Eq. (2). For the production of deterministic results, the tokens for masked patches are sampled with the largest probabilities at once.

4. Experiments

The evaluation is conducted at $256 \times 256$ (i.e., $H = 256$ and $W = 256$) resolution on three different datasets, including FFHQ [25], Places2 [61] and ImageNet [11]. We use the original training and testing splits for Places2 and ImageNet. For FFHQ, we maintain the last 1K images for evaluation and others for training. Following ICT [49], only 1K images are randomly chosen from the test split of ImageNet for evaluation and the irregular masks provided by PConv [28] are used both for training and testing.

4.1. Implementation Details

We use P-VQVAE with the same model size and UQ-Transformer with different model sizes for different datasets. The number of latent vectors in dual-codebook (i.e., $K$ and $K'$) both are set to 512. The detailed architecture of P-VQVAE and UQ-Transformer could be found in the supplementary material. We train P-VQVAE with batch size 128 and train UQ-Transformer with batch size 48 for FFHQ and 96 for Places2 and ImageNet. The learning rate is warmed up from 0 to 2e-4 and 3e-4 in the first 5K iterations for P-VQVAE and UQ-Transformer, and then de-
Specifically, DFv2 and EC generally produce blurry images, while the texture of the results generated by MED and PIC contain lots of unnatural artifacts. Compared with ICT, PUT is more powerful in the understanding of global context and maintaining the meaningful textures, as shown in the last row in Figure 4. We argue that the superiority of PUT is mainly due to: 1) high-resolution and 2) unquantized transformer. Both of them are vital for preserving the information contained in the input images, which are helpful to produce photo-realism images.

4.2. Main Results

We compare the proposed PUT with the following state-of-the-art inpainting approaches: DeepFillv2 (DFv2) [55], Edge-Connect (EC) [32], MED [29], PIC [60] and ICT [49]. Among them, the last two ones can generate pluralistic results for each input while the others can only produce one deterministic result for the given input. For a fair comparison, we directly use the pre-trained models provided by the authors when available, otherwise train the models by ourselves using the codes and settings provided by the authors.

**Qualitative Comparisons.** We first qualitatively compare the inpainted results of different methods in Figure 4. Specifically, DFv2 and EC generally produce blurry images, while the texture of the results generated by MED and PIC contain lots of unnatural artifacts. Compared with ICT, PUT is more powerful in the understanding of global context and maintaining the meaningful textures, as shown in the last row in Figure 4. We argue that the superiority of PUT is mainly due to: 1) high-resolution and 2) unquantized transformer. Both of them are vital for preserving the information contained in the input images, which are helpful to produce photo-realism images.

**User Study.** For the evaluation of subjective quality, the user study is further conducted. Only pluralistic methods are evaluated, including PIC [60], ICT [49] and PUT. Specifically, we randomly sample 20 pairs of image and mask from the test set of each dataset. For each pair, we generate two inpainted results using each method, and ask the participants to rank these six images according to their photorealism from high to low. We calculate the ratio of each method among the rank 1 images. Results are shown in Figure 5. Our method takes up at least 60% of the rank 1 images, demonstrating its superiority.

**Quantitative Comparisons.** We further demonstrate the superiority of PUT in diversity and fidelity with other pluralistic methods. Specifically, the mean LPIPS distance [57] between pairs of randomly generated results for the same input image is calculated. Following ICT [49], five pairs of methods are evaluated, including PIC [60], ICT [49] and PUT. Specifically, we randomly sample 20 pairs of image and mask from the test set of each dataset. For each pair, we generate two inpainted results using each method, and ask the participants to rank these six images according to their photorealism from high to low. We calculate the ratio of each method among the rank 1 images. Results are shown in Figure 5. Our method takes up at least 60% of the rank 1 images, demonstrating its superiority.

Table 1. Quantitative results of different methods. The methods are divided into deterministic and pluralistic groups. The subscript “50” of ICT and PUT is the value of $K$, while the subscript “all” of ICT and PUT means all tokens are sampled at one iteration.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FID ↓</th>
<th>PSNR ↑</th>
<th>SSIM↑</th>
<th>MAE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT (Ours)</td>
<td>0.936</td>
<td>0.845</td>
<td>0.906</td>
<td>0.861</td>
</tr>
<tr>
<td>PUT (Ours)</td>
<td>0.936</td>
<td>0.845</td>
<td>0.906</td>
<td>0.861</td>
</tr>
</tbody>
</table>

Table 1. Quantitative results of different methods. The methods are divided into deterministic and pluralistic groups. The subscript “50” of ICT and PUT is the value of $K$, while the subscript “all” of ICT and PUT means all tokens are sampled at one iteration.
per input image are generated. Meanwhile, the Fréchet Inception Distance (FID) [20] is also computed between the inpainted images and ground-truth images to reveal the fidelity of generated results. The curves of LPIPS and FID are shown in Figure 6. It can be seen that among the three pluralistic methods, PUT achieves the best fidelity (lowest FID) on all datasets, especially for large area of masked regions and complex scenes (ImageNet [11]). Although PUT and ICT both are implemented with transformers, PUT presents a higher diversity, we owe this to the feeding of original continuous feature vectors without quantization to transformer, which has no information loss issue.

In Table 1, we compare all methods in terms of several metrics, including the peak signal-to-noise (PSNR), structural similarity index (SSIM), relative L1 (MAE) and FID. Only one recovered output is produced for each input. Overall, for pluralistic methods, PUT50 performs the best in almost all metrics on all datasets. Specifically, the FID score of PUT50 on ImageNet with mask ratio 40%-60% is 10.44 lower than that of ICT50. For deterministic methods, PUTall also achieves almost the best performance. However, the metrics are suitable for the comparison between pluralistic and deterministic methods since diverse and meaningful results can be generated by pluralistic methods.

In Table 2, we further compare PUTall with LaMa [40] and CoModGAN [59], both of which are recently proposed methods for image inpainting with resolution $512 \times 512$. For PUTall, an upsample network used in ICT [49] is trained to upsample the results from $256 \times 256$ to $512 \times 512$. As we can see, PUTall achieves a much better PSNR than CoModGAN, and it is also comparable with LaMa.

### 4.3. Discussions

#### Effectiveness of different components.

To show the effectiveness of the patch-based encoder, dual-codebook, multi-scale guided decoder and un-quantized transformer, several methods are designed: 1) PUTconv means that the patch-based encoder is replaced with a normal CNN encoder, which is implemented with convolution layers; 2) PUTone refers that there is only one codebook for vector quantization; 3) PUTno-ref means that there is no reference branch in the decoder; and 4) PUTtok denotes that the transformer takes the quantized vectors as input rather than the original feature vectors from encoder. In order to show the effectiveness of random quantization of feature vectors while training UQ-Transformer, we further design the fifth model, PUTqua0, that trains UQ-Transformer without random quantization. For all models, except the modifications mentioned, others remain the same with our default model.
PUT. Results are shown in Table 3 and Figure 7.

Among all models, PUT\textsuperscript{conv} performs the worst in all metrics, demonstrating the effectiveness of the non-overlapping patch partition design. Within CNN based encoder, the input images are processed in a sliding window manner, introducing the interaction between masked and unmasked regions, which is fatal to transformer for the prediction of masked regions.

Comparing PUT\textsuperscript{one} to PUT, the only difference is the training of P-VQVAE with one or two codebooks since the codebook $e'$ is not used in the inference stage. P-VQVAE can learn more discriminative features for masked and unmasked patches with the help of dual-codebook. Interestingly, PUT indeed performs better than PUT\textsuperscript{one} except the FID score on FFHQ. We speculate that face generation is much easier because all faces share a similar structure. Besides, the facial structure is related with the position in the image since most of the images in FFHQ contain faces in their near-center locations. As we can see in Figure 7, PUT\textsuperscript{one} sometimes predicts black patches, which is similar to those patches containing missing pixels. Nonetheless, PUT achieves overall better performance than PUT\textsuperscript{one}.

Compared with PUT, PUT\textsuperscript{no,ref} constructs the inpainted image without referencing to the input masked image, which leads to a inferior performance in terms of all metrics. For PUT, some useful textures can be recovered with the help of the guidance from the unmasked regions in reference image. As we can see in Figure 7, the result of PUT\textsuperscript{no,ref} is over smoothed, which is unnatural.

Compared with PUT\textsuperscript{no}, PUT performs much better in all metrics. Without quantizing feature vectors to discrete representations, no information contained in the feature will be lost. Such practice helps transformer to understand complex content and maintain the inherit meaningful textures in the input image. However, the training of UQ-Transformer should be carefully designed by randomly quantizing the input feature vectors since only quantized vectors can be obtained for masked regions at the inference stage. The effectiveness of such random quantization during training is obvious while comparing PUT with PUT\textsuperscript{no}.  

5. Conclusions and Limitations

In this paper, we present a novel method, PUT, for pluralistic image inpainting. PUT consists of two main components: 1) patch-based auto-encoder (P-VQVAE) and 2) unquantized transformer (UQ-Transformer). With the help of P-VQVAE and UQ-Transformer, PUT processes the original high-resolution image without quantization. Such practice preserves the information contained in the input image as much as possible. The main limitation of PUT is the inference speed for the production of diverse results. However, it is a common issue of existing transformer based auto-regressive methods \cite{65, 66, 67, 68}. Two solutions could be adopted to alleviate this limitation: 1) replacing the used transformer block with more efficient ones \cite{69, 70} and 2) sampling tokens for several patches at each iteration. In addition, PUT may be used for editing the contents of images to achieve illicit goals, which can be mitigated using existing synthesized image detectors \cite{71}. Experimental results demonstrate the superiority of PUT, including the fidelity and diversity, especially for large masked regions and complex scenes (such as ImageNet \cite{72}).

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62002336, No. U20B2047) and the Exploration Fund Project of University of Science and Technology of China (YD3480002001).
References


[43] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingmeng He, Lu Yuan, Gang Hua, and Nenghai Yu. Efficient semantic image synthesis via class-adaptive normalization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. 2


