This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Improving Detail in Pluralistic Image Inpainting with Feature Dequantization

Kyungri Park[†] and Woohwan Jung
Department of Applied Artificial Intelligence, Hanyang University
{arten, whjung}@hanyang.ac.kr

Abstract

Pluralistic Image Inpainting (PII) offers multiple plausible solutions for restoring missing parts of images and has been successfully applied to various applications including image editing and object removal. Recently, VQGAN-based methods have been proposed and have shown that they significantly improve the structural integrity in the generated images. Nevertheless, the state-of-the-art VQGAN-based model PUT faces a critical challenge: degradation of detail quality in output images due to feature quantization. Feature quantization restricts the latent space and causes information loss, which negatively affects the detail quality essential for image inpainting. To tackle the problem, we propose the FDM (Feature Dequantization Module) specifically designed to restore the detail quality of images by compensating for the information loss. Furthermore, we develop an efficient training method for FDM which drastically reduces training costs. We empirically demonstrate that our method significantly enhances the detail quality of the generated images with negligible training and inference overheads. The code is available at https://github.com/hyudsl/FDM

1. Introduction

Pluralistic Image Inpainting (PII), which offers multiple plausible solutions for missing parts of images, has gained attention for enhancing user satisfaction in real-world applications. By providing a variety of candidates, PII not only increases user engagement but also ensures that the final results align closely with individual preferences. Consequently, PII has been applied to various real-world tasks, including image editing [12] and object removal [3, 28].

Recent studies [24, 30] have demonstrated that sampling values for masked regions in the codebook can generate more diverse and well-structured images. ICT [30] sampling the RGB value of the pixel-level codebook (a set of RGB values) for each masked pixel in the image. However, to reduce the sampling cost, ICT downsamples the input image, leading to information loss. To address this issue, PUT [24] encodes each patch into features instead of downsampling. PUT samples features from a patch-level codebook (a set of features) for each masked patch, and the decoder reconstructs the output image using a quantized feature map consisting of the selected features.

Masked Image | Generated Area | Inpainted Image

Figure 1. An example of the visible boundary between the generated area and the masked image caused by feature quantization. Although the generated area in the center image appears plausible and realistic, a slight color mismatch at the boundary makes it noticeable when combined with the masked image (on the right).

However, feature quantization with the codebook still leads to information loss by restricting the decoder's input space to a discrete feature space. This information loss can degrade the detail quality in output images. In image inpainting, degraded detail quality may result in mismatches between the texture and color of the generated area and the surrounding background. Such inconsistencies can create noticeable boundaries between the generated area and the background, causing the final image to appear unnatural. For instance, while the image in the center of Figure 1 appears realistic, the image on the right shows clear visibility of the masked area due to color mismatches when merging with the background.

To address this problem, we introduce FDM (Feature Dequantization Module), designed to estimate the gap (error) between the ideal features and the quantized features. FDM compensates for this gap by adding the estimated corrections to the quantized features, significantly improving the detail quality of the generated images. However, a straightforward end-to-end training of FDM is infeasible

[†]Major in Bio Artificial Intelligence

due to the iterative codebook sampling in VQGAN. Therefore, we propose an efficient method to train FDM that reduces training costs by over 99% based on our estimation. Moreover, the inference cost of FDM is negligible, as feature dequantization is performed only once per input image.

To evaluate the effectiveness of FDM, we conducted comparative experiments with state-of-the-art PII models. The results of our experiments indicated that applying FDM enhances the details and structural consistency while preserving diversity. Furthermore, we applied FDM to various image generation tasks beyond inpainting. These experiments consistently demonstrated improved performance, highlighting the effectiveness of FDM across a wide range of VQGAN-based image generation tasks.

The contributions of this paper are as follows:

- We identify the information loss in the state-of-the-art pluralistic image inpainting model, PUT.
- We propose the FDM which significantly improves the detail quality and consistency of the generated images by compensating for the information loss.
- We develop an efficient training method for FDM which drastically reduces training costs without sampling procedure.

2. Related Work

Pluralistic image inpainting methods. PII can generate multiple results for each input, unlike conventional inpainting methods that typically produce a single result. VAE [16]-based methods [23, 34] have been proposed to enable diverse image generation. PIC [34] encodes masked inputs into a Gaussian distribution, generating diverse images via latent vector sampling. PD-GAN [23] combines prior inpainted images and SPADE [25] to enhance diversity, with latent vector decoding conditioned on deterministic inpainting results. Recent Transformer-based methods [20, 24, 30] outperform VAE-based approaches. ICT [30] generates low-resolution prior image through the Transformer and then up-samples the prior image using a CNN to generate the final results. PUT [24] utilizes the VQGAN architecture [5] and effectively generates diverse results by reducing input information loss through a patch-based encoder. However, it has limited representational capacity due to the use of a discrete codebook, which can lead to distorted structures and color discrepancies. To address this issue, we propose FDM, which increases the representational capacity through feature dequantization.

Image generation methods with VQGAN. VQGAN, based on VQ-VAE [29], is a method designed for high-resolution image generation. While VQGAN and its variants [1, 10, 11] efficiently generate images using a codebook, the discrete latent space of the codebook imposes limitations on its representational capacity compared to conventional VAE-based image generation methods which typically operate in a continuous latent space. Several methods [18, 19, 21, 32] have been proposed to enhance the representational capacity of codebooks in image generation. The RQ-VAE [18] introduces the use of multiple codes to represent the latent vector through a residual quantizer. In contrast to employing multiple codebooks, the RQ-VAE efficiently enhances representational capacity by utilizing a single codebook while employing multiple codes. RQ-VAE needs additional patch sampler for multiple codes. In contrast, our proposed method can easily enhance performance by adding a very small module without altering the overall structure. FA-VAE [21] aims to recover missing details that occur during feature quantization in the frequency domain. It achieves this by learning to match the frequency of the feature map output during the decoding process with that of the feature map during the encoding process, effectively restoring missing details. In contrast to these previous methods, we propose FDM to enhance the representational capacity for pluralistic image inpainting.

3. Proposed Method

Our proposed method aims to enhance the details of inpainted images using the Feature Dequantization Module (FDM). We begin with an overview of our method, followed by a training strategy for FDM, and finally provide the entire training procedure.

3.1. Overview

Let $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ be an image and $\mathbf{m} \in \{0,1\}^{H\times W}$ be a binary mask, where H and W represent the height and width of the image, respectively. The masked image is represented as $\hat{\mathbf{x}} \in \mathbb{R}^{H\times W\times 3}$ where $\hat{\mathbf{x}} = \mathbf{x} \otimes \mathbf{m}$ and $\otimes$ denotes element-wise multiplication. Pixels with a value of 0 in $\hat{\mathbf{x}}$ will be inpainted. Our goal is to generate diverse images that contain content similar to the original image $\mathbf{x}$, starting from the masked image $\hat{\mathbf{x}}$. An overview of our proposed inpainting procedure consists of the following four steps as illustrated in Figure 2: Encoding, feature sampling, feature dequantization, and decoding. It should be noted that, except for Step 3, our approach aligns with PUT [24], and additionally removing Step 2 yields a variational auto-encoder (VAE). We introduce the dequantization (Step 3), to address the issue of information loss occurred by the patch-wise feature sampling (Step 2).

Step 1: Encoding. We first partition the masked image $\hat{\mathbf{x}}$ into $\frac{H}{r} \times \frac{W}{r}$ non-overlapping patches, each of size $r \times r$. For each patch, the encoder generates a C-dimensional feature vector. The collection of this feature vectors from a feature map, represented as $\hat{\mathbf{f}} \in \mathbb{R}^{\frac{H}{r}\times\frac{W}{r}\times C}$. Following [24], we

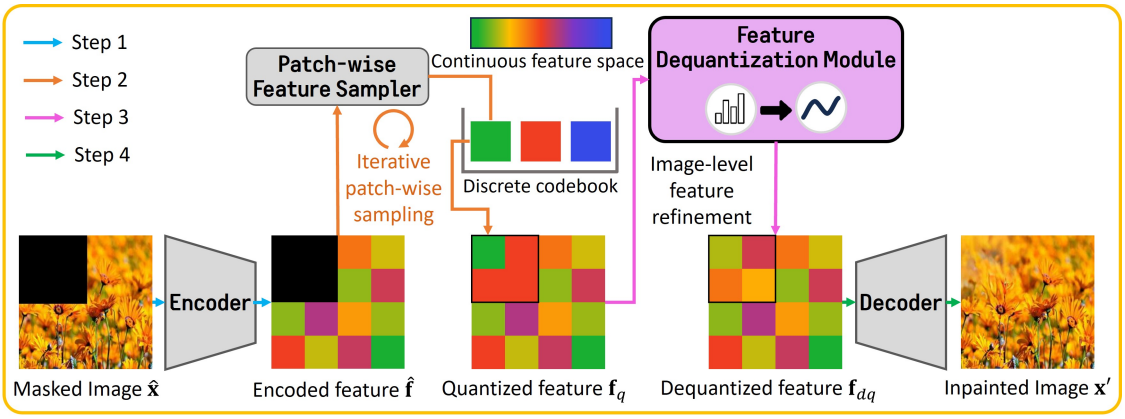


Figure 2. An overview of the proposed method. Quantized features are limited to distinct points, represented by green, red, and blue. However, through the proposed FDM module, dequantization expands the representation to a continuous space.

set the patch size r to 8 and the dimensionality C to 256. We provide detailed information about the structure of the encoder-decoder in the supplementary material.

Step 2: Patch-wise feature sampling. The feature-sampler takes the encoded feature map $\hat{\mathbf{f}}$ as input and outputs a quantized feature $\mathbf{f}_q$. In this step, the feature vector of each masked patch is replaced by a vector sampled from a codebook which is a set of $\mathcal{N}$ possible feature vectors. To produce diverse results, each patch is sampled autoregressively using Gibbs sampling from a probability distribution predicted by a patch-wise feature-sampler. Based on the learned distribution, the feature-sampler can sample from among the most suitable $\mathcal{K}$ vectors, where $\mathcal{K}$ represents a hyper-parameter that controls the diversity of sampling. Since the codebook contains a limited number of possible latent vectors, the feature-sampler can easily learn distribution of latent vector in each patch. The masked features are replaced with features from the codebook, which are sampled by the feature-sampler.

However, the sampling with codebook induces the problem of feature quantization. The feature map generated by the feature-sampler is quantized to the codebook's feature vectors. Feature quantization restricts the decoder to access only limited points, causing information loss. Consequently, this information loss reduces the representational capability and degrades the detail quality of the output image. In image inpainting, where consistency with the background is essential, detail degradation can lead to mismatched colors and texture with the background, resulting in lower image quality. While increasing size of the codebook improves approximation accuracy of the continuous feature space, it does not completely resolve the issue of information loss and increases sampling difficulty.

Step 3: Feature dequantization. To address information loss caused by feature quantization, we introduce a simple solution called FDM (Feature Dequantization Module) which involves dequantizing the sampled features.

Quantization error is defined as the difference between the original continuous feature and its quantized representation. FDM compensates for the error by adding the estimated quantization error to the quantized features. Let $\mathbf{f} \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C}$ be the ideal continuous feature related to the quantized feature $\mathbf{f}_q$. The quantization error can be represented as $\mathbf{f}_{qe} = \mathbf{f} - \mathbf{f}_q$. FDM predicts $\mathbf{f}_{qe}$ and adds it to $\mathbf{f}_q$ to achieve dequantization.

FDM takes $\mathbf{f}_q$ and the downsampled mask $\mathbf{m}_d \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C}$ as inputs to predict the quantization error:

$$\tilde{\mathbf{f}}_{qe} = \mathcal{F}_\theta([\mathbf{f}_q, \mathbf{m}_d])$$

where $\tilde{\mathbf{f}}_{qe}$ is predicted quantization error by the estimation function $\mathcal{F}_\theta$. Note that the feature of unmasked patch in $\mathbf{f}_q$ is the same as $\hat{\mathbf{f}}$ because the feature-sampler only replaces the features of masked patches. The estimation function $\mathcal{F}_\theta$ is a simple network composed of 8 residual blocks [8], each consisting of a 3x3 convolutional layer followed by a ReLU activation function and a 1x1 convolutional layer. After the 1x1 convolutional layer, the output is added to the input of the block and passed through a ReLU activation function. Residual blocks were chosen for their ability to learn while preserving input information, ensuring structural similarity with quantized features even after dequantization.

The predicted quantization error $\tilde{\mathbf{f}}_{qe}$ is added to $\mathbf{f}_q$ for dequantization:

$$\mathbf{f}_{dq} = \mathbf{f}_q + \tilde{\mathbf{f}}_{qe} \otimes (1 - \mathbf{m}_d)$$

where $\otimes$ represents the element-wise multiplication operation, and $\mathbf{f}_{dq}$ is the predicted dequantized feature. It is worth noting that the feature dequantization is performed done only once for an input image. By doing this, we can minimize the computational overhead. Additionally, FDM

is applied after all patches have been sampled. It complements the features of each patch across the entire image, thereby enhancing the consistency of image.

Step 4: Decoding. The decoder takes the dequantized feature $\mathbf{f}_{dq}$ as input and generates an inpainted image $\mathbf{x}'$:

$$\mathbf{x}' = Decoder(\mathbf{f}_{dq})$$

where the decoder is a convolutional neural network.

3.2. Training the Feature Dequantization Module

The full model training with the reconstruction loss is a straightforward method to train FDM. However, it incurs substantial costs particularly due to the iterative sampling. Thus, we propose an efficient method to train FDM without the feature-sampler and the decoder to minimize the training cost. Additionally, this method prevents the potential catastrophic forgetting that may occur when the decoder is jointly trained with a randomly initialized FDM, offering a cost-effective and reliable alternative to the conventional, more expensive training process.

Quantization error prediction. Recall that FDM predicts the quantization error $\mathbf{f}_{qe}$ by the following equation: $\tilde{\mathbf{f}}_{qe} = \mathcal{F}_\theta([\mathbf{f}_q, \mathbf{m}_d])$. Thus, we can directly train FDM to predict the quantization error $\mathbf{f}_{qe}$ without the decoder with the following quantization error prediction loss L_{qe}:

$$\mathcal{L}_{qe} = |\mathbf{f}_{qe} - \mathcal{F}_\theta([\mathbf{f}_q, \mathbf{m}_d])|. \quad (1)$$

However, the training with the above loss involve the iterative patch-wise feature sampling process to generate $\mathbf{f}_q$. It causes a significant overhead in the training process since the number of iterations is equal to the number of patches $\frac{H}{r} \cdot \frac{W}{r}$(1024 in our experiments). To avoid the excessive training cost, we propose a method to train without the feature-sampler.

Training without the feature-sampler. The main idea to eliminate the feature-sampler from the training process is to utilize unmasked images. Firstly, we encode the unmasked image $\mathbf{x}$ using a frozen encoder to obtain the encoded unmasked feature $\mathbf{f}' \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C}$. Then, the feature-sampler quantizes the feature $\mathbf{f}'$ to obtain $\mathbf{f}'_q \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C}$. Finally, the target value $\mathbf{f}'_{qe}$ is calculated as $\mathbf{f}' \otimes \mathbf{m} - \mathbf{f}'_q \otimes (1 - \mathbf{m}_d)$. In other words, we approximate $\mathcal{L}_{qe}$ in Equation (1) using this formulation:

$$\mathcal{L}_{qe} \approx |\mathbf{f}'_{qe} - \mathcal{F}_\theta([\mathbf{f}'_q, \mathbf{m}_d])|$$

where $\mathbf{f}'_{qe} = \mathbf{f}' \otimes \mathbf{m} - \mathbf{f}'_q \otimes (1 - \mathbf{m}_d)$.

Given the identical sampling setting used in evaluation, one training iteration with the patch sampling consumes around 387 seconds with 16 batches, making training cumbersome. Our modification reduces the training time to approximately 2.1 second per iteration, making FDM training 184 times faster compared to the training with patch-sampling method.

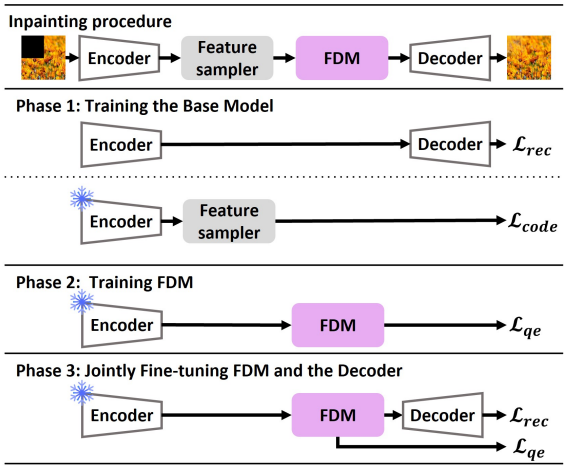


Figure 3. An overview of the training procedure.

3.3. Training Procedure

Figure 3 shows the entire training procedure of our proposed method. First, we train the encoder-decoder and the feature-sampler as in [24]. Next, we train FDM as presented in 3.2. Finally, we jointly fine-tune FDM and the decoder.

Phase 1: Training the base model. We train the base model PUT which consists of encoder-decoder and a patch-wise feature-sampler. To train the encoder-decoder, we employ the image reconstruction loss $\mathcal{L}_{rec}$ between the target image and the reconstructed image. It is a weighted sum of L1 loss $\mathcal{L}_{l1}$, gradient loss $\mathcal{L}_G$, adversarial loss [7] $\mathcal{L}_A$, perceptual loss [13] $\mathcal{L}_P$, and style loss [6] $\mathcal{L}_S$:

$$\mathcal{L}_{rec} = \mathcal{L}_{l1} + \lambda_G \mathcal{L}_G + \lambda_A \mathcal{L}_A + \lambda_P \mathcal{L}_P + \lambda_S \mathcal{L}_S. \quad (2)$$

We set $\lambda_G = 5$, $\lambda_A = 0.1$, $\lambda_P = 0.1$ and $\lambda_S = 250$ following [24]. For training the feature-sampler, we utilize the code classification loss $\mathcal{L}_{code}$. This loss measures the cross-entropy between the predicted code class probability of each patch and the target code class, where the code class indicates the latent vector of the codebook.

In phase 1, the encoder-decoder is initially trained using $\mathcal{L}_{rec}$ for image reconstruction. Then, with the encoder frozen, the feature-sampler is trained using $\mathcal{L}_{code}$ to learn the distribution of latent vectors per patch. While the trained encoder-decoder and feature-sampler alone can construct an inpainting model, we further enhance performance by integrating the FDM, into the model.

Phase 2: Training FDM. In Phase 2, the feature dequantization module is trained using $\mathcal{L}_{qe}$ as described in Section 3.2, with the encoder frozen during this process. We freeze the encoder to prevent them from forgetting previously learned features.

Phase 3: Jointly fine-tuning FDM and the decoder. After the initial training FDM, we jointly fine-tune FDM and the

decoder which are separately trained before this phase. The loss function for the fine-tuning $\mathcal{L}_{rec}$ is as follows:

$$\mathcal{L}_{tuning} = \lambda_{qe}\mathcal{L}_{qe} + \lambda_{rec}\mathcal{L}_{rec} \tag{3}$$

where $\lambda_{qe} = 1$, $\lambda_{rec} = 1$, and $\mathcal{L}_{rec}$ remains the same as in Equation (2). Since the input of the decoder is modified by FDM, we fine-tune the decoder. To ensure that FDM does not forget its ability to predict quantization error, we continue to use $\mathcal{L}_{qe}$ during training.

4. Experiments

4.1. Datasets and Settings

Datasets. The evaluation is conducted at 256 × 256 resolution on two datasets: Places [35] and Paris Street View Dataset [4]. Irregular masks provided by PConv [22] are used for both training and testing. In the experimental results, mask ratios between 0.2 and 0.4 are referred to as small masks, while those between 0.4 and 0.6 are labeled as large masks. Following [30], we only utilized a subset of Places for our experiments, while keeping the training and test splits consistent with the original dataset. We keep 237,777 images for training and reserve 800 images for testing. In the Paris Street View Dataset, for consistent evaluation, we reorganized the existing split to use 14,000 images for training and 1,000 images for testing.

Metric. Evaluation is conducted using Fréchet Inception Distance (FID) [9] and Learned Perceptual Image Patch Similarity (LPIPS) [33]. We selected FID and LPIPS as evaluation metrics because they closely resemble human perceptual capabilities. In contrast, PSNR, SSIM [31], and MAE are not well-aligned with human perceptual capabilities [17, 27] due to their pixel-wise calculations. Therefore, they are not suitable for evaluating pluralistic inpainting. However, FID and LPIPS evaluate images in the feature space, allowing for judgments that are much closer to human perception, making them more suitable for pluralistic inpainting. Thus, we provided result of PSNR, SSIM, and MAE in the supplementary materials. For evaluation, we use only one generated result per input.

We evaluate diversity score using LPIPS similar to [30]. Unlike evaluating inpainting performance, when measuring diversity, LPIPS is computed using only the generated images. First, we generate N paired pluralistic inpainted images using the same mask. Then, the mean LPIPS score for each pair is used as the diversity score. For evaluation, we generated $N = 5$ pairs for each image.

Compared methods. We compare the proposed method with the following state-of-the-art pluralistic inpainting approaches: ICT [30], MAT [20], LDM [26] and PUT [24]. We evaluate using the provided pre-trained models. In cases where pre-trained models are not available, we train the models using the code and settings provided by the authors.

Experimental settings. The detailed structures of the encoder-decoder and feature-sampler follow the PUT architecture and use the same model size for both datasets. We set the patch size r to 8, resulting in 1024 patches for a 256×256 resolution image. The number of latent vectors in the codebook $\mathcal{N}$ is set to 512. We set $\mathcal{K} = 50$ for pluralistic results in ICT, PUT, and FDM. Training of FDM utilizes the same settings as when training the encoder-decoder. The learning rate is warmed up from 0 to 2e-4 in the first epoch and then decayed with a cosine scheduler. FDM and decoder are optimized with Adam [15] ($\beta_1 = 0, \beta_2 = 0.9$). The FDM and encoder-decoder are trained for 100 epochs, while the feature sampler is trained for 300 epochs. Training stops if there is no improvement in the baseline metric—validation loss for encoder-decoder and FDM, and classification accuracy for the feature sampler—over 10 consecutive epochs. All models are trained on a machine with six GeForce RTX 2080 Ti GPUs.

4.2. Main Results

Qualitative comparisons. Figure 4 provides a detailed comparison between PUT and FDM. Results from PUT contain color discrepancies and distorted structures. For example, in the top-left sample, the region generated by PUT appears brighter than the background, making the masked area clearly visible. However, after applying feature dequantization through FDM, the generated color matches the background more accurately. Another example is in the top-right sample, where the parking lines generated by PUT should be straight but appear distorted. Additionally, there are areas where asphalt is generated in grass colors. In our results, parking lines are straight, and asphalt is represented in the same color as the background.

The pluralistic results of various methods are depicted in Figure 5. In this comparison, ICT produces significant artifacts, resulting in unnatural image. While MAT is more natural than ICT, it lacks diversity and exhibits similar structures across images. In contrast, our proposed method reliably generates diverse samples based on the probabilistic patch sampling of PUT.

Quantitative comparisons. Table 1 presents a comparison of methods across each evaluation metric. In most cases, the proposed method achieved the best score across both datasets. Particularly, our proposed method achieved the best FID score, especially with large masks.

The proposed method consistently outperforms PUT, demonstrating the effectiveness of FDM. Moreover, performance generally ranks on the order of FDM, PUT, and ICT, underscoring the importance of minimizing information loss to achieve superior results. The significant performance improvement observed with large masks compared to small masks further highlights this point. This is because a larger mask reduces the background information available

Figure 4. Detail comparison between proposed method and PUT. More results are presented in the supplementary material.

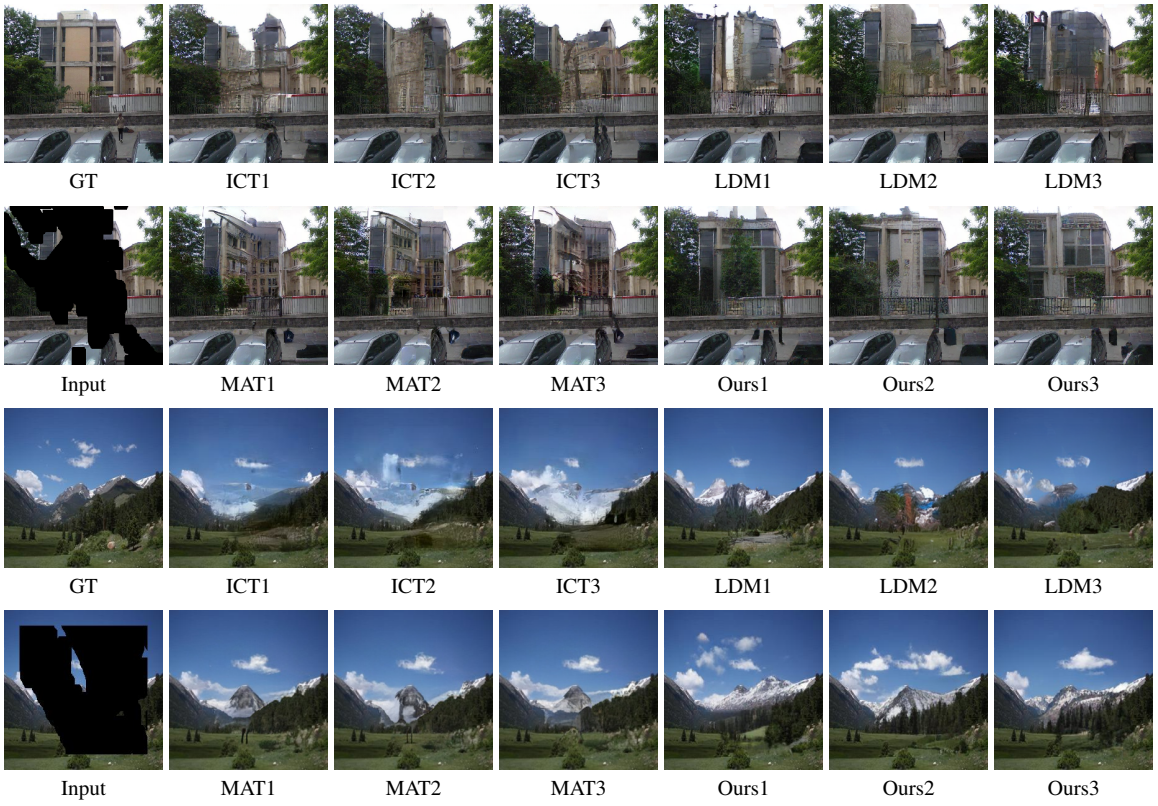


Figure 5. Visual comparison of diverse inpainting results in Places and Paris Street View.

for reference, thereby increasing the model's dependence on the provided information for accurate representation.

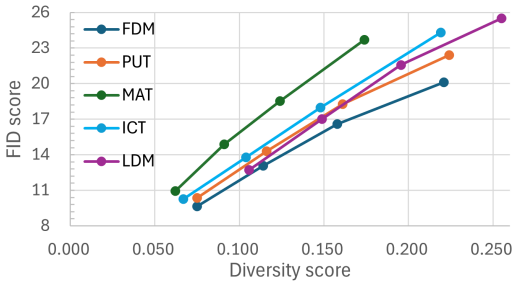
Figure 6 shows the diversity score of each method. In the graph, the y-axis represents the FID score, and the x-axis represents the diversity score. Each point corresponds to different mask ratios, ranging from 0.2 to 0.6, with intervals of 0.1 units. An ideal position on the graph is in the bottom-right corner, indicating low FID scores and high diversity scores. This indicates the ability to generate diverse and natural-looking images. MAT, while demonstrating FID performance comparable to our proposed method on the Places dataset, shows a very low diversity score. LDM has a high diversity score, but at higher mask ratios, it shows a lower FID score compared to our method. Our proposed method achieved the best FID scores in most cases while still maintaining diversity of PUT.

4.3. Analysis

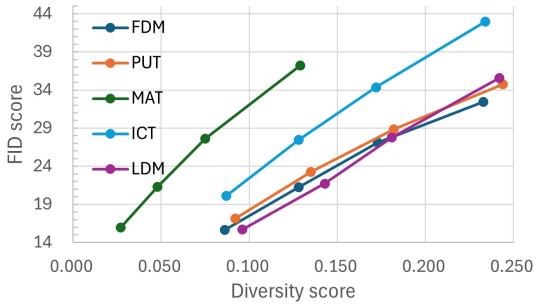
Computational overhead of FDM. Table 2 shows the computational overhead of FDM in training and inference times. During training, it occupies only 12% of the total training time. Training FDM with sampling, estimated based on the

Table 1. Quantitative results of different methods. Red indicates the best score, while blue indicates the second-best score. Note that results for MAT and LDM on Places were obtained by training with 8.4 M and 1.8 M data, respectively. In contrast, the results of ICT, PUT, and our proposed method were obtained from training with the 0.24 M data same as [30].

Methods	Paris Street View				Places			
	FID		LPIPS		FID		LPIPS	
	small	large	small	large	small	large	small	large
ICT	12.02	21.14	0.143	0.252	23.74	38.65	0.155	0.260
MAT	13.22	21.45	0.151	0.266	18.39	32.09	0.128	0.222
LDM	14.87	23.53	0.150	0.255	18.71	31.66	0.136	0.265
PUT	12.58	20.52	0.135	0.240	20.25	32.01	0.137	0.240
Ours (PUT +FDM)	11.63	18.66	0.131	0.234	18.46	29.67	0.127	0.230



(a) Paris Street View



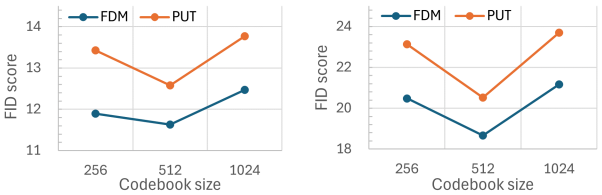
(b) Places

Figure 6. Diversity score on different methods.

Table 2. Computational overhead of FDM on the Paris Street View dataset with a mask ratio of 20-30%.

Methods	♯ Param.	Traning Time	Inference	
			Time	FLOPs
PUT	119M	05d 09h	25.236s	59.331T
FDM	3M	00d 17h	0.004s	0.004T
Ours (PUT +FDM)	122M	06d 02h	25.240s	59.335T

required time for one iteration, would take approximately 131 days. To tackle this problem, we suggested training FDM using ground truth without sampling. With our training strategy, FDM can be trained in just 17 hours, which is 184 times faster than the naive training method. Therefore, we propose a method to practically apply FDM to inpainting through ground-truth training. The number of parameters for FDM is approximately 2.5% of the total number of parameters. The FLOPs for FDM is approximately 0.01% of the total FLOPs. Moreover, the inference time accounts for only about 0.02% of the overall time. The result demonstrates that FDM improves generation quality with a minimal time overhead.



(a) Small mask (b) Large mask

Figure 7. FID scores based on codebook size in the Paris Street View dataset.

Table 3. L2 distance between generated feature and ideal feature.

Mask ratio	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6
PUT	3.685	3.675	3.678	3.669
Ours (PUT +FDM)	2.873	2.876	2.893	2.901
Difference	0.812	0.799	0.785	0.768

Relationship between codebook size and FDM's effectiveness. Figure 7 is the FID score graph based on codebook size. Inpainting performance does not always scale proportionally with codebook size, unlike reconstruction performance. Therefore, increasing codebook size is less effective than FDM in addressing information loss. Moreover, FDM consistently improves performance regardless of codebook size. Given its consistent performance improvements and simplicity, FDM provides an efficient solution for addressing information loss.

Feature dequantization. Table 3 presents L2 distance between the generated features and the corresponding ideal features in the Places dataset. The ideal features are ob-

Figure 8. Detail comparison between ours and VQGAN.

Table 4. FID score comparison for various image generation tasks.

Methods	Uncond. (FFHQ)	Semantic (ADE20K)	Class (ImageNet)
VQGAN	17.03	33.26	14.65
Ours (VQGAN +FDM)	**15.84**	**31.32**	**13.46**

tained by encoding the original image using the encoder. The results indicate that the features restored by FDM closely align with the original feature space across all mask ratios. Our proposed method generates features that are closer to the ideal features compared to those generated by PUT, resulting in more plausible results.

4.4. Applying FDM to Image Generation Tasks

We conducted additional experiments to demonstrate that FDM can improve VQGAN in various image generation tasks: unconditional image generation, semantic-conditional image generation, and class-conditional image synthesis. For the three tasks, we use the FFHQ [14], ADE20K [36], and ImageNet [2] datasets, respectively.

Experimental settings. The experiments are conducted by adding FDM to VQGAN [5], a prominent vector-quantization based image generation model. FDM is applied after all patches are predicted, as described in Section 3.1. For unconditional and class-conditional, FDM takes only the quantized feature $\mathbf{f}_q$ as input without a mask. The training procedure follows Section 3.3, and Phase 1 is skipped since we use the pre-trained model provided by [5]. We follow the training settings provided by [5].

Qualitative comparisons. Figure 8 provides a detailed comparison between VQGAN and Ours. Images generated by VQGAN often lack detailed representation. For instance, in the ADE20K sample at the center, VQGAN-produced images suffer from blurring and gradient effects, which obscure the distinction between building windows and walls. However, in the images generated using our method, the window frames are clear and distinguishable from the walls. Another example from the rightmost sample shows that window frames in VQGAN-generated images are split into two sections, whereas our method produces window frames as single, continuous pieces.

Quantitative comparisons. Table 4 shows the FID score comparison for image synthesis. After applying FDM, there was an improvement in FID scores across all tasks. This demonstrates that our method can effectively and simply enhance vector-quantization based models across various image generation tasks, not just inpainting.

5. Conclusion and Limitations

In this paper, we studied the pluralistic image inpainting (PII) problem which offers multiple plausible solutions for missing image parts. We introduced FDM (Feature Dequantization Module), which enhances representational capacity through feature dequantization, thereby improving the details of generated images. FDM can be seamlessly applied during the inference phases with minimal overhead. In addition, we proposed an efficient training method to train FDM which dramatically reduces the training cost by removing the sampling in the training phase. Furthermore, through experiments, our proposed method has demonstrated effectiveness across various image generation tasks, not just limited to image inpainting.

In VQGAN-based PII, the sequence in which patches are inpainted plays an important role in defining both the quality and variety of the output images. However, since our method FDM is applied after the feature-sampler, it cannot affect the order of inpainting. To address this, we can consider configuring FDM to perform dequantization on the unmasked parts of the input containing masked patches. This enables FDM to be applied during the sampling process, allowing it to affect the inpainting order.

6. Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program(IITP-2024-RS-2024-00423071) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation). This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.RS-2023-00261068, Development of Lightweight Multimodal Anti-Phishing Models and Split-Learning Techniques for Privacy-Preserving Anti-Phishing)

References

[1] Shiyue Cao, Yueqin Yin, Lianghua Huang, Yu Liu, Xin Zhao, Deli Zhao, and Kaigi Huang. Efficient-vqgan: Towards high-resolution image generation with efficient vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7368–7377, 2023. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[3] Nizam Ud Din, Kamran Javed, Seho Bae, and Juneho Yi. A novel gan-based network for unmasking of masked face. *IEEE Access*, 8:44276–44287, 2020. 1

[4] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012. 5

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2, 8

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[10] Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22596–22605, 2023. 2

[11] Mengqi Huang, Zhendong Mao, Quan Wang, and Yongdong Zhang. Not all image regions matter: Masked vector quantization for autoregressive image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2023. 2

[12] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 4

[14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 8

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014. 2

[17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 5

[18] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 2

[19] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and WOOK SHIN HAN. Draft-and-revise: Effective image generation with contextual rq-transformer. *Advances in Neural Information Processing Systems*, 35:30127–30138, 2022. 2

[20] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 5

[21] Xinmiao Lin, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Yu Kong. Catch missing details: Image reconstruction with frequency augmented variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1736–1745, 2023. 2

[22] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 5

[23] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9371–9381, 2021. 2

[24] Qiankun Liu, Zhentao Tan, Dongdong Chen, Qi Chu, Xiyang Dai, Yinpeng Chen, Mengchen Liu, Lu Yuan, and Nenghai Yu. Reduce information loss in transformers for pluralistic image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11347–11357, 2022. 1, 2, 4, 5

[25] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 2

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image

synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[27] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 4491–4500, 2017. 5

[28] Rakshith R Shetty, Mario Fritz, and Bernt Schiele. Adversarial scene editing: Automatic object removal from weak supervision. *Advances in Neural Information Processing Systems*, 31, 2018. 1

[29] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[30] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701, 2021. 1, 2, 5, 7

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[32] Tackgeun You, Saehoon Kim, Chiheon Kim, Doyup Lee, and Bohyung Han. Locally hierarchical auto-regressive modeling for image generation. *Advances in Neural Information Processing Systems*, 35:16360–16372, 2022. 2

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

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

[35] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 5

[36] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 8