Bridging Global Context Interactions for High-Fidelity Pluralistic Image Completion

Chuanxia Zheng, Guoxian Song, Tat-Jen Cham, Jianfei Cai, *Fellow, IEEE,* Linjie Luo, and Dinh Phung

Abstract—We introduce PICFormer, a novel framework for **P**luralistic **I**mage **C**ompletion using a trans**Former** based architecture, that achieves both high quality and diversity at a much faster inference speed. Our key contribution is to introduce a *code-shared* codebook learning using a restrictive CNN on small and non-overlapping receptive fields (RFs) for the *local* visible token representation. This results in a compact yet expressive discrete representation, facilitating efficient modeling of *global* visible context relations by the transformer. Unlike the prevailing autoregressive approaches, we proposed to sample all tokens simultaneously, leading to more than 100× faster inference speed. To enhance appearance consistency between visible and generated regions, we further propose a novel attention-aware layer (AAL), designed to better exploit distantly related high-frequency features. Through extensive experiments, we demonstrate that the PICFormer efficiently learns semantically-rich discrete codes, resulting in significantly improved image quality. Moreover, our diverse image completion framework surpasses state-of-the-art methods on multiple image completion datasets. The project page is available at https://chuanxiaz.com/picformer/.

Index Terms—Image Completion, Codebook Learning, Image Editing, Transformer.

1 Introduction

IMAGE completion, also named "inpainting" [4], refers to the task of filling masked regions with alternative reasonable content, as well as realistic appearance seamlessly. Applications include restoring damaged paintings [4], removing objects [5], generating new content for occluded regions [6], and freely editing an image [7].

To infer plausible content, many learning-based approaches [7]–[13] have been proposed. However, most of them provide *only a single solution* to a given masked image, despite the multi-modal nature of the problem. PIC [14], [15] is a pioneering effort that aimed to generate *multiple* and *diverse* plausible results. While it carefully sought a balance between $\mathcal{KL}$ loss and reconstruction loss in a variational encoder (VAE), its conventional architecture resulted in limited diversity and quality.

Inspired by iGPT [16], some recent efforts, *e.g.* ICT [12] and BAT-Fill [17], directly predict underlying discrete tokens' possibility through a transformer. However, these methods rely on a pre-clustered palette at the *pixel-level*, leading to diminished image quality. While the concurrent approach PUT [18], [19] employed the codebook learning and transformer at the *patch-level* for the pluralistic image completion, they predicted the underlying distribution of discrete tokens from continuous encoded features. It remains a challenge to correctly bridge and exploit globally visible pixels after it had been degraded by arbitrary masks. Furthermore, they require multiple samplings to predict the token, which takes a ruinously expensive inference time.

In this work, we propose PICFormer, a novel framework for **P**luralistic **I**mage **C**ompletion using a trans**Former** architecture. A crucial observation lies in the transformer's ability to directly exploit long-range dependencies at every encoder layer through the attention mechanism, which creates *an equal flowing opportunity for all visible pixels*, regardless of their relative spatial positions (Fig. 4). This mitigates the proximity-dominant influences that may otherwise result in semantically incoherent outcomes.

However, unlike in NLP, where each word is naturally treated as a vector for token embedding [20]–[22], it is unclear *what a good token representation should be for a visual task*. If every *pixel* is embedded as a token, the memory cost will make this infeasible except for very small down-sampled images, like 32×32 or 64×64 scale in iGPT [16], ICT [12] and Imagen [23]. To obtain a practical length for the transformer, we learn a token representation in the intermediate *feature* space, an approach also broadly taken by other vision transformers [24]–[28]. However, unlike these methods that use conventional CNN-based encoders to embed the tokens, *without considering the visible information flow in image completion*, we found it essential to decouple the local token representation and the global context interaction. In particular, we present a *restrictive CNN* to learn a *code-shared codebook*, with a simple, compact yet expressive discrete token representation. To do so, we ensure the tokens represent locally visible information, each within a *small* and *non-overlapping* patch, while enforcing the global context interactions between these local tokens to be explicitly and co-equally perceived in every transformer layer. As a result, each masked pixel will not be gradually affected by neighboring visible pixels.

On top of this practical sequencing approach (Fig. 2(a)), a transformer architecture with *a weighted bidirectional attention*

- *Chuanxia Zheng is with the Engineering Science department, University of Oxford, UK. E-mail: see https://www.chuanxiaz.com/*
- *Jianfei Cai and Dinh Phung are with the Department of Data Science & AI, Monash University, Australia.*
- *Guoxian Song and Linjie Luo are with ByteDance Inc.*
- *Tat-Jen Cham is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
- *This work was originally done when Chuanxia Zheng was a research fellow at Monash University.*

Manuscript received April 19, 2005; revised August 26, 2015.

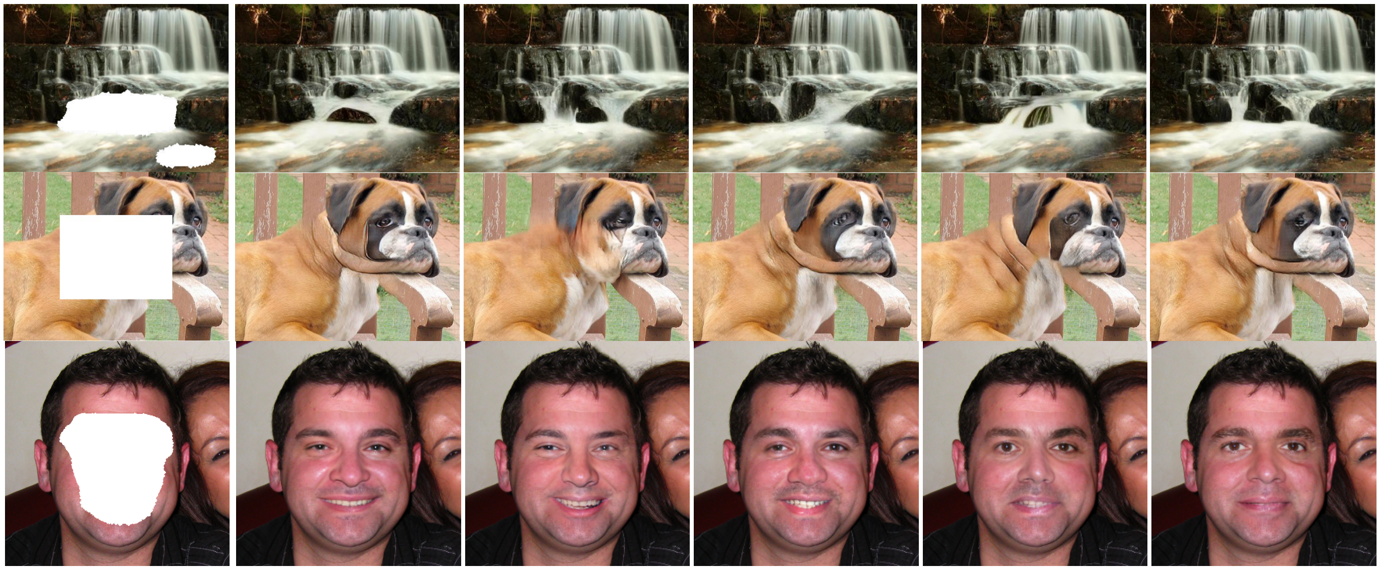


Fig. 1. **Multiple and diverse samples from our PICFormer.** Our method produces high-fidelity diverse results given masked images on various datasets (from top to bottom: Places [1], ImageNet [2], and FFHQ [3], respectively.) More results are represented as **animations** in the last column.

module is introduced for modeling the global context relationships. While this meets the multiple plausible solutions, the quantized operator is still lossy, and the PICFormer-Coarse only works for a *fixed* sequence length due to the position embedding (Fig. 2(b)). To further improve the quality of the completed image and allow our system to flexibly scale to images of arbitrary sizes, especially at higher resolution, a fully convolutional network (Fig. 2(c)) is subsequently applied to refine the visual appearance. A novel Attention-Aware Layer (AAL) is inserted between the encoder and decoder that adaptively balances the attention paid to visible and generated content, leading to semantically superior feature transfer (Fig. 7 and Fig. 8).

Finally, we propose one-time sampling, rather than the prevailing autoregressive-dependent sampling as in ICT [12], BAT-Fill [17], and PUT [18]. This strategy achieves an impressive runtime of approximately 83ms per image on an NVIDIA 3090, which is over 100× faster inference compared to these recurrent sampling.

A preliminary version of this work was published in CVPR'22 [29] and the earlier pluralistic image completion was published in CVPR'19 [14] and IJCV'21 [15]. Compared to the conference version, *i.e.* TFill [29], we have introduced a significant amount of new materials on goals, methods, and results specified as follows: i) Here, we aimed to achieve not only high-fidelity but also *diverse* results, instead of the single "best" solution. ii) To achieve this goal, except using the original *restrictive CNN*, we introduce a novel *code-sharing* mechanism in the discrete latent space that facilitates a better discrete token representation. iii) Moreover, we simultaneously sample all tokens at one time, *resulting in a much faster inference time than the prevailing autoregressive approach for sequence generation* [12], [17], [18]. It is worth noting that the proposed pipeline has also been applied into a series of applications since then [30]–[32]. The code and models are publicly available at https://github.com/lyndonzheng/TFill.

2 Related Work

The image completion methods either utilize the visible context from within the image (**intra-image**) [4], or learn the statistical context from large datasets (**inter-image**) [33].

Intra-Image Completion. Classical image completion methods directly propagate, copy, or realign visible pixels to missing regions. One category of the intra-image completion methods is the diffusion-based image synthesis [4], [34]–[36], which propagates the locally surrounded visible pixels to the missing regions, achieving smooth results, yet work only for small and narrow holes. In contrast, patch-based approaches [5], [37]–[39] copy and realign the pixels from visible patches to missing regions for larger and more complex holes by analysing and parsing the low-level features in multiple patches. These approaches produce texture-consistent images. However, they only utilize information within a single image, and thus they are *not* able to generate semantically new content.

Inter-Image Completion. To generate new content, inter-image completion approaches borrow statistical information from a large dataset. Hays and Efros [33] first filled new content from a huge dataset by cutting the corresponding regions from the retrieval image and pasting them into the missing regions. However, it requires the dataset to be large enough to contain an image similar to the arbitrary masked image, which is hard to be met.

Recently, learning-based approaches have become prevailing for image compilation. Köhler, Schuler, Schölkopf, *et al.* [40] and Ren, Xu, Yan, *et al.* [41] first introduced Convolutional Neural Networks (CNNs) [42] for the image completion task. However, they work only on small and thin holes. In contrast, Pathak, Krahenbuhl, Donahue, *et al.* [8] performed CNN on larger 64 × 64 holes by utilizing the Generative Adversarial Networks (GANs) [43]. This is followed by [9], [13], [30], [44]–[46]. More recently, Lama [47] utilized the Fourier convolutions to handle the large mask

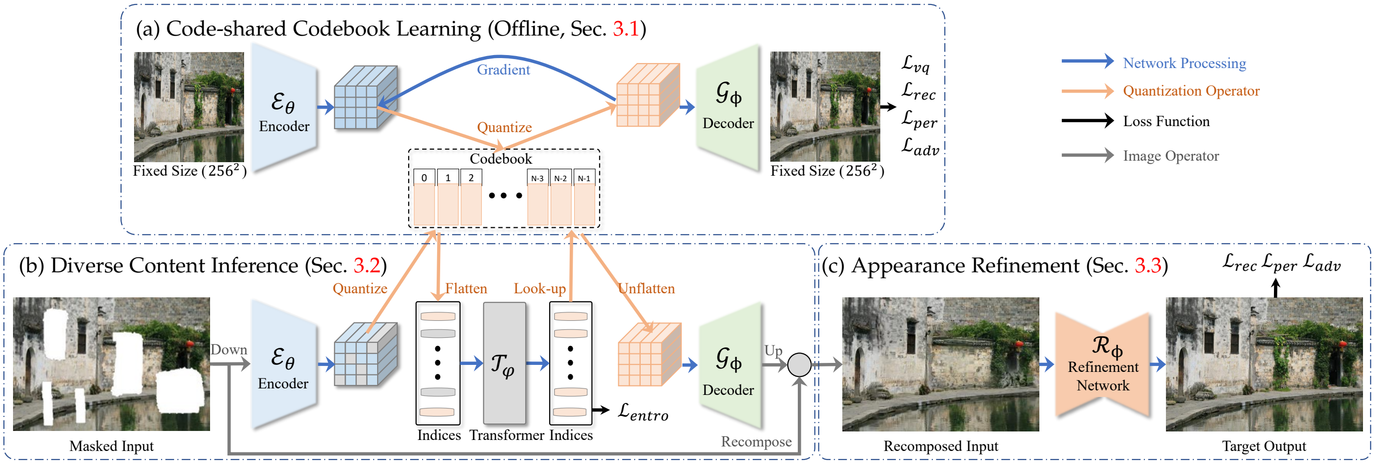


Fig. 2. **The overall pipeline of PICFormer**. (a) It first learns a quantizer using a code-shared strategy, along with a restrictive CNN. (b) A transformer is then applied to infer the composition of the original embedded indices. (c) Finally, we sample the top $\mathcal{K}$ results, merge them with the original high-resolution image, and pass them to a refinement network with an **A**ttention-**A**ware **L**ayer (**AAL**) to transfer high-quality information from both visible and generated regions. Note that only the bottom pipeline is used during inference, while the top pipeline is for learning the quantizer offline.

completion. Besides, various auxiliary information has been explored for semantic image completion [7], [48]–[51]. While these approaches learn the statistical information from a large dataset and then infer reasonable content for a masked input image, their completed appearances may *not be consistent with the original visible pixels*.

Intra- and Inter-Image Completion. To generate reasonable content, as well as visually consistent appearance, a natural idea is to combine intra- and inter-image completion approaches. For instance, Yang, Lu, Lin, *et al.* [52] applied neural patch synthesis for high-resolution image completion, where high-frequency details are copied and aligned using the mid-layer features. Inspired by PatchMatch [39], Yu, Lin, Yang, *et al.* [10] introduced a Contextual Attention (CA) module to copy similar features from visible regions to missing regions. This is followed by a series of works [29], [49], [53]–[60]. However, all these approaches provide only one "single" result for one masked image, which is not true in many scenarios, especially for the large missing regions.

Pluralistic Image Completion. We first introduced the "pluralistic image completion (PIC)" task in [14], which generates multiple and diverse results for a given masked image. This task has witnessed a series of following research since then [11], [17], [19], [30], [61], [62]. Although these approaches provide some diversified results, their image quality cannot always be guaranteed due to the variational training in GANs network, which suffers from unstable training and "mode collapse". Repaint [45] and PanoDiffion [46] introduced the diffusion models to generate diverse results. Inspired by iGPT [16], Wan *et al.* [12] and Yu *et al.* [17] directly applied this architecture for pluralistic image completion. However, their discrete space is a pre-clustered palette on pixel-level, in which images are downsampled to a small resolution, *e.g.* 32 × 32. This may not impact the image classification task [16], [63], but the generated results are of low-quality and unimpressive visual appearance. Inspired by the vector quantization (VQ) approaches [26], [64]–[68], Peng *et al.* [69] and Liu *et al.* [18] applied the VQ-based learning pipeline for the pluralistic image completion.

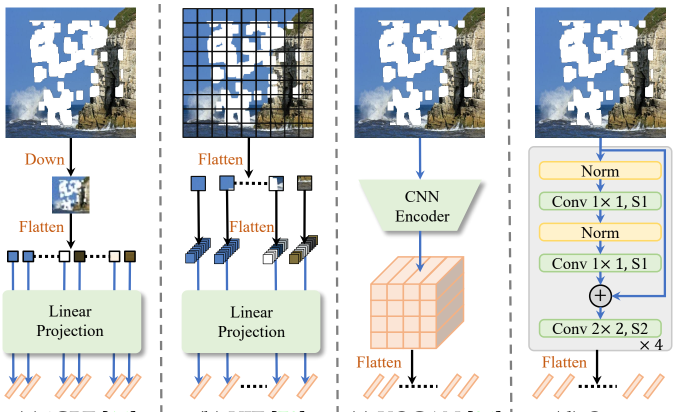
3 Approach

Given a single masked image x_m, our goal is to learn a model Φ to generate high-fidelity completed images. We opt to go beyond a single best result to deal with *multiple* and *diverse* solutions.

To achieve that, we learn to estimate the underlying prior distribution in a discrete space, rather than mapping the whole dataset to a predefined distribution, *e.g.* $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in previous works [11], [14], [30], [61]. Our proposed PICFormer, illustrated in Fig. 2, consists of three major stages during training: i) An encoder-decoder network (Fig. 2(a)), along with a learnable codebook is offline trained to embed images into discrete vectors using the proposed *code-shared* learning and the *restrictive* CNN. ii) With such an effective quantizer, both masked input image x_m and ground truth target image x_{gt} are respectively embedded and flattened into index sequences $\mathbf{s}_m$ and $\mathbf{s}_{gt}$. Then, a weighted bidirectional transformer is employed to infer the possibility $p(\mathbf{s}|\mathbf{s}_m)$ of degraded indices by exploiting the global interactions within a sequence of tractable length (Fig. 2(b)). iii) Finally, a refinement network (Fig. 2(c)) is designed to refine appearance by utilizing high-resolution features globally, and also frees the limitation to fixed sizes.

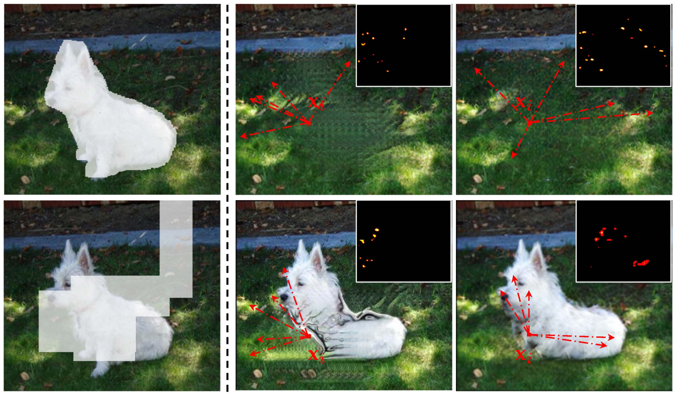
3.1 Code-shared Codebook Learning

To infer the multiple and diverse plausible contents, our PICFormer-*Coarse* (Fig. 2(b)) utilizes a transformer to *equally* perceiving global visible context. However, considering the dramatically increased computational cost in the transformer, different embedding methods have been explored to extract a practicable length of visual tokens [12], [16], [24]–[26], [28], [70]–[73]. These visual tokens' RF is either as small as a pixel (*e.g.* iGPT [16], Fig. 3(a)) that loses important context details, or is as large as the full image size (*e.g.* VQGAN [26], Fig. 3(c)) that has been gradually influenced by neighboring pixels in deep CNN layers. While patch embedding [70] (Fig. 3(b)) achieves impressive performance in many tasks, one-layer linear projection is still not good enough [27]. This motivated us to develop the following



(a) iGPT [16] (b) VIT [70] (c) VQGAN [26] (d) Ours

Fig. 3. **Token representation.** (a) Pixel to token. (b) Patch to token. (c) Feature to token. (d) Restrictive **R**eceptive **F**ield (RF) feature to token.



(a) Masked Input (b) HiFill [55] (c) Ours

Fig. 4. **An example of information flow.** The position $\mathbf{x}_i$'s response (flow) is calculated by inferring the *Jacobian* matrix between it to all other pixels. PICFormer correctly captures long-range visible context flow in different settings (top: object removal, bottom: object completion), even with a large mask splitting two semantically important zones.

restrictive CNN (Fig. 3 (d)) and *code-shared* strategy (Fig. 5) that plays the key role in token representations learning.

Quantization. Our discrete token embedding approach is built upon [64]. Given an image $x \in \mathbb{R}^{H\times W\times 3}$, it will be represented as a spatial composition of codebook entries $\mathbf{z}_q \in \mathbb{R}^{h\times w\times n_q}$. In particular, an image is reconstructed by:

$$\hat{x} = \mathcal{G}_\phi(\mathbf{z}_q) = \mathcal{G}_\phi(\mathbf{q}(\mathcal{E}_\theta(x))), \tag{1}$$

where the encoder $\mathcal{E}_\theta$ embeds an image x as a feature $\hat{\mathbf{z}}$, with the same dimensionality as the codebook entries $\mathbf{z}_q$, and the decoder $\mathcal{G}_\phi$ reversely transfers the quantized feature $\mathbf{z}_q = \mathbf{q}(\hat{\mathbf{z}})$ back to the image domain. The quantized operator $\mathbf{q}(\cdot)$ is performed by looking up the closest entry z_k in the codebook for each spatial grid feature $\hat{z}_{ij}$ in $\hat{\mathbf{z}}$:

$$\mathbf{z}_q = \mathbf{q}(\hat{\mathbf{z}}) = \arg\min_{z_k\in\mathbb{Z}} \|\hat{z}_{ij} - z_k\|. \tag{2}$$

Restrictive CNN. Existing approaches [26], [66] employed the conventional CNN to embed the quantized tokens. These tokens already hold global RF, which has a large negative effect on image completion, because the masked holes have been gradually determined by the neighboring visible pixels. To mitigate this issue, we propose the *restrictive* CNN blocks (Fig. 3(d)) that ensure each token represents only the visible information in a local patch, *leaving the long-range dependencies to be explicitly modeled by a transformer*. Our proposed change is simple. In each block, the 1 × 1 filter and `layernorm` is applied for non-linear projection, followed by a partial convolution layer [44] that uses a 2×2 filter with stride 2 to extract visible information.

In fact, some concurrent works also begin to explore the influence of different token embeddings, like Swin [74] used shift windows to get multi-scales features and ViT$_c$ [27] demonstrated an early CNN token embedding is important for visual transformer. However, they do not consider context flowing from visible to masked regions. In Fig. 4, we empirically show this is precisely the case for prior CNN-based models. Because masked regions originally hold uniform values, *i.e.* 0, they will take the neighboring visible pixels as a filled and reasonable value for the next layer. In contrast, as the small patch is directly embedded using local visible context with important weight, the proposed restrictive CNN is better suited for image completion tasks.

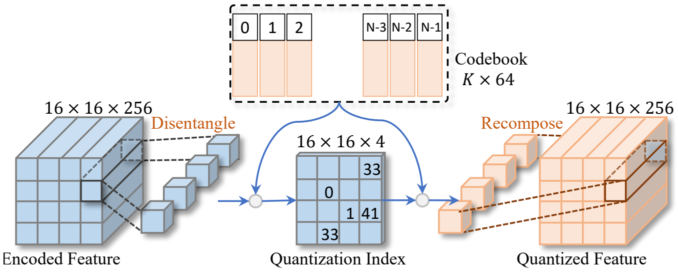


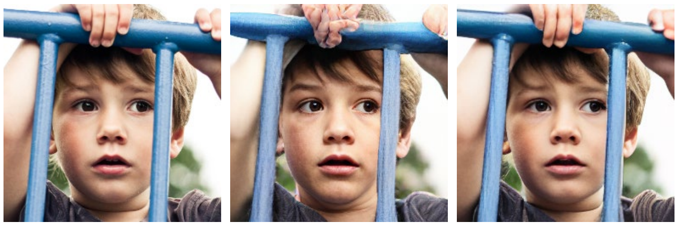
Fig. 5. **Code-shared codebook learning.** Unlike existing methods [26], [64] that quantize the whole feature, we first separate each feature into multiple groups (*e.g.* 4 or 8) and then quantize these sub-features.

Code-shared Codebook. Our broad conjecture here is that the feature in a single spatial location may consist of *multiple attributes* that should be disentangled. Motivated by this, we propose to chunk each grid feature into multiple tablets and then embed each of them into a vector in the codebook. Note that this is very different from existing VQ-based quantizers, which embed each spatial grid feature entirely to its closest codebook entry. Our key novel insight is that doing so will allow the network to learn to decouple the attributes along the channel dimension, while at the same time supporting a flexible recomposition of these entries, yet without increasing the number of learnable weights in the encoder and decoder.

Our proposed pipeline is illustrated in Fig. 5. Given an encoded feature $\hat{\mathbf{z}} \in \mathbb{R}^{h\times w\times n_z}$, we directly subdivide it along the channel dimension into multiple chunks,

$$\hat{z}_{ij} = \{\hat{z}_{ij}^{(1)}, \cdots, \hat{z}_{ij}^{(c)}\}, \tag{3}$$

where $\hat{z}_{ij}^{(\cdot)} \in \mathbb{R}^{n_z/c}$. Then, each chunk is quantized to its closest entry z_k in the codebook using the Eq. (2), and the equivalent index representation becomes c-channel, rather than a single-channel representation as in existing approaches [26], [64], [66]. The quantized features can be recomposed and reused on the spatial positions, resulting in the same length of the index sequence, *i.e.* $h \times w$.



(a) Original (b) VQGAN [26] (c) Ours

0.6773 / 19.96 / 0.1509 **0.8500** / **24.07** / **0.0645**

Fig. 6. **Reconstructed results of different quantizers**. The scores are SSIMs, PSNR, and LPIPS, respectively. (a) Input image. (b) Results of VQGAN [26]. While the image quality is high, some details are lost. (c) In contrast, our proposed code-shared method achieves much better reconstruction quality under the same feature resolution.

While the concurrent work RQVAE [68] also represents an image as a multichannel codebook index, they achieve this by calculating the residual information between the quantized feature and the continuous feature, resulting in a long time of processing, due to the recursive quantization. Furthermore, our proposed multi-channel codebook learning has also been used for image generation in [31], [75].

Loss Functions. Following VQ-GAN [26], multiple loss functions are applied to learn a perceptually rich codebook:

$$\mathcal{L}_{VQ}(\mathcal{E}_\theta, \mathcal{G}_\phi, \mathbb{Z}) = \mathcal{L}_{rec} + \mathcal{L}_{per} + \mathcal{L}_{vq} + \mathcal{L}_{adv}. \tag{4}$$

The first and second terms are the data terms to measure the pixel-level reconstruction loss and feature-level perceptual loss, respectively. The third term in (4) measures the distance between the encoded feature $\hat{\mathbf{z}}$ and the quantized feature $\mathbf{z}_q$:

$$\mathcal{L}_{vq} = \|\text{sg}[\mathcal{E}_\theta(x)] - \mathbf{z}_q\|^2 + \beta\|\mathcal{E}_\theta(x) - \text{sg}[\mathbf{z}_q]\|^2, \tag{5}$$

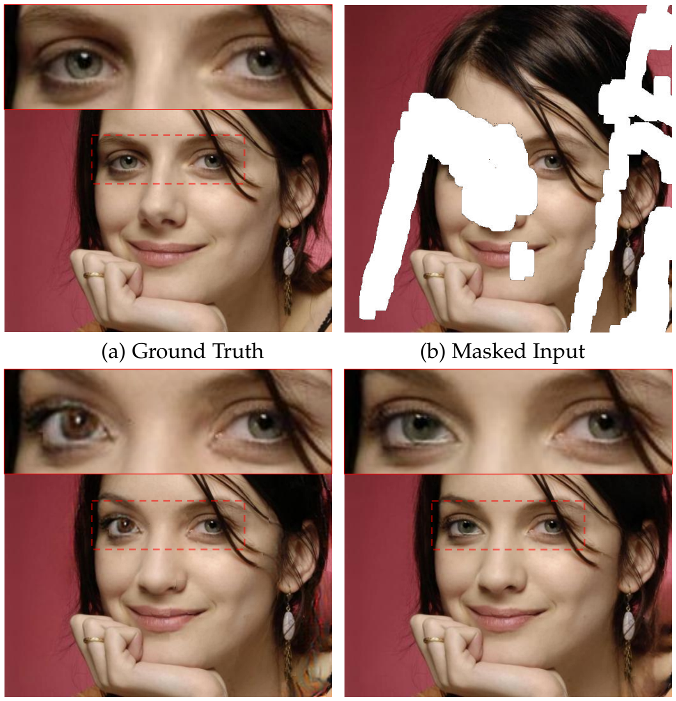
where "sg" stands for the stop-gradient operator [64], which forces the corresponding operand to be non-updated. Therefore, this loss encourages the learned quantized features $\mathbf{z}_q$ to move towards the encoded features $\mathcal{E}_\theta(x)$ in the first part of (5), while concurrently making the encoded features commit to the codebook space via the second part of (5). To further improve image quality, an adversarial training loss [43] is introduced in VQ-GAN [26]:

$$\mathcal{L}_{adv} = \arg\min_{\mathcal{E}_\theta, \mathcal{G}_\phi, \mathbb{Z}} \max_{\mathcal{D}} \mathbb{E}_{x\sim p(x)}[\log D(x) + \log(1 - D(\hat{x}))], \tag{6}$$

where $\hat{x}$ is the output image as in (1). In practice, we optimize the hinge version of this adversarial loss [76].

3.2 Diverse Content Inference: PICFormer-*Coarse*

With the above-trained codebook $\mathbb{Z}$, an image can now be represented as a set of code indices in low resolution, *i.e.* $h \times w = 16 \times 16$, which can be further flattened into a sequence $\mathbf{s} = [s^1; s^2; \dots; s^N]$, where $N = 16 \times 16$. With this practical sequence length of N, we implement the transformer encoder [20] for image completion, by modeling the global context interactions in every attention layer. In particular, given an index sequence $\mathbf{s}$, we first project each of its elements into a feature to get the sequence $\mathbf{E} = [\mathbf{e}^1; \mathbf{e}^2; \dots; \mathbf{e}^N]$ through a learnable embedding. Note that, due to our code-sharing mechanism, each sequence symbol s^n has multiple channels, *i.e.* 4 as shown in Fig. 5.



(a) Ground Truth (b) Masked Input

(c) PICFormer-*Coarse* (d) PICFormer-*Refined*

Fig. 7. **Coarse and refined results**. (a) Ground truth. (b) Masked input. (c) Coarse output. (d) Refined output. The refinement network not only increases image quality to a high resolution (256^2 *vs* 512^2), but also encourages the left eyeball to be consistent with the visible right eyeball.

Weighted Bi-directional Self-attention. To further *bias* the important visible values, we implement the self-attention with a *weighted* attention layer. In particular, the initial weight $w^{(1)} \in (0.02, 1.0]$ is obtained by calculating the fraction of visible pixels in a small patch, *e.g.* $192/(16 \times 16)$ means 192 pixels in the 16×16 patch are visible, which explicitly indicates the significance of embedded discrete tokens. Then, the feature and weight are passed into the transformer with a weighted bidirectional attention, where the original attention score will be scaled by the weight $\mathbf{w}_\ell$. The weight $\mathbf{w} \in (0.02, 1]$ is updated as $\sqrt{\mathbf{w}_{\ell-1}} \rightarrow \mathbf{w}_\ell$ after each transformer block, to reflect visible information flow.

Loss Function. Given indices $\mathbf{s}_m$, quantized from the masked image x_m, the transformer $\mathcal{T}_\psi$ learns to predict the distribution of possible indices, *i.e.* $p(\mathbf{s}|\mathbf{s}_m)$, with the target indices $\mathbf{s}_{gt}$ that are quantized from the original image x. The learning is guided by minimizing the following loss:

$$\mathcal{L}(\mathcal{T}_\psi) = \mathbb{E}_{x\sim p(x)}[-\log p(\mathbf{s}|\mathbf{s}_m)]. \tag{7}$$

Sampling Strategy. Given the estimated discrete distribution, we can sample the indices at different chunks. Specifically, the transformer predicts the probability of K codebook entries for all chunks, and then we sample tokens based on their confidence scores. As opposed to prior works [12], [17], [26], we observe that *independently sampling all positions can also produce superior results*, which dramatically reduces the inference time. This is because the global dependencies have been modeled in every transformer layer through the attention module, and the generated tokens are constrained by visible tokens. A similar strategy has been employed in the concurrent work MaskGIT [77] and MoVQ [75].

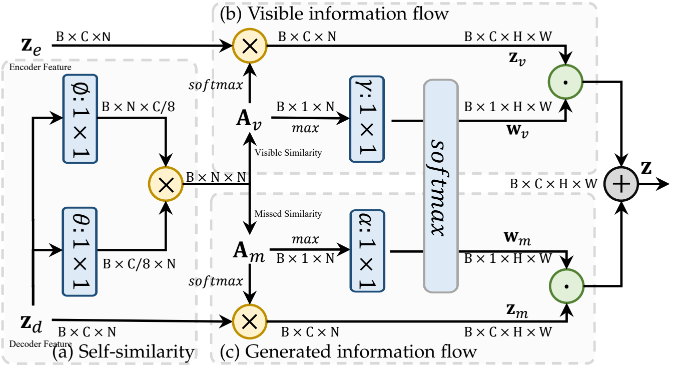


Fig. 8. **Attention-aware layer.** The feature maps are shown as tensors. "$\otimes$" denotes matrix multiplication, "$\odot$" denotes element-wise multiplication and "$\oplus$" is element-wise sum. The blue boxes denote 1×1 convolution filters that are learned.

3.3 Appearance Refinement: PICFormer-*Refined*

Through the local codebook learning and the global context modeling, our PICFormer-*Coarse* (Fig. 2(b)) correctly infers *reasonable* and *diverse* content by equally utilizing the global visible context in every layer. However, the coarse pipeline has several limitations: i) While our code-shared method significantly improves the performance of the codebook representation (as shown in Fig. 6), the quantization still discards some visual details. ii) The diverse contents are inferred using embedded low-resolution indices, *i.e.* 16×16 scale. The realistic completed results may *not* be fully consistent with the original visible appearances, *e.g.* the completed eye in Fig. 7 (c). iii) The coarse pipeline is *not* suitable for high-resolution input due to the fixed length position embedding in the transformer.

Attention-Aware Layer (AAL). To mitigate these issues, a refinement network, trained on higher-resolution images, is proposed (Fig. 2(c)), in which an **A**ttention-**A**ware **L**ayer (AAL, Fig. 8) is designed to copy long-range context from both *encoded* and *decoded* features. In particular, the coarse image $\hat{x}$ is first upsampled to the original image resolution and recomposed with the original high-resolution pixels by:

$$x_{comp} = M \odot x_m + (1 - M) \odot \hat{x}^{\uparrow}, \tag{8}$$

where M is the initial binary mask with 0 denoting holes, x_m is the masked image and $\hat{x}^{\uparrow}$ is the corresponding output after resizing. Then, the encoder-decoder architecture is applied to get the encoded features $\mathbf{z}_e$ and the decoded features $\mathbf{z}_d$. We first calculate the attention score of:

$$\mathbf{A} = \phi(\mathbf{z}_d)^{\intercal}\theta(\mathbf{z}_d), \tag{9}$$

where $\mathbf{A}_{ij}$ represents the similarity of the i^{th} feature to the j^{th} feature, and ϕ, θ are $1{\times}1$ convolution filters.

Interestingly, we discover that using $\mathbf{A}$ directly in a standard self-attention layer is suboptimal [14], because the $\mathbf{z}_d$ features for visible regions are generally distinct from those generated for masked regions. Consequently, *the attention tends to be insular*, with masked regions preferentially attending to masked regions, and vice versa. To avoid this problem, we explicitly handled the attention to visible regions separately from masked regions. So before $\texttt{Softmax}$ normalization, $\mathbf{A}$ is split into two parts: $\mathbf{A}_v$ — similarity to *visible* regions, and $\mathbf{A}_m$ — similarity to generated *masked* regions. Next, we get long-range dependencies via:

$$\mathbf{z}_v = \texttt{softmax}(\mathbf{A}_v)\mathbf{z}_e \quad , \quad \mathbf{z}_m = \texttt{softmax}(\mathbf{A}_m)\mathbf{z}_d \tag{10}$$

where $\mathbf{z}_v$ contains features of contextual flow [10] for copying high-frequency details from the encoded features $\mathbf{z}_e$ to holes, while $\mathbf{z}_m$ has features from the self-attention that is used in SAGAN [78] for high-quality image generation.

Instead of learning fixed weights as in PIC [14] to combine $\mathbf{z}_v$ and $\mathbf{z}_m$, we further learn the *weights mapping* based on the largest attention score in each position. Specifically, we first obtain the largest attention score of $\mathbf{A}_v$ and $\mathbf{A}_m$, respectively. Then, we use the $1{\times}1$ filter γ and α to *modulate* the ratio of the weights. $\texttt{Softmax}$ normalization is applied to ensure $\mathbf{w}_v{+}\mathbf{w}_m{=}1$ in every spatial position:

$$[\mathbf{w}_v, \mathbf{w}_w] = \texttt{softmax}([\gamma(\texttt{max}(\mathbf{A}_v)), \alpha(\texttt{max}(\mathbf{A}_m)]])) \tag{11}$$

where $\texttt{max}$ is executed on the attention score channel. Finally, an attention-balanced output $\mathbf{z}$ is obtained by:

$$\mathbf{z} = \mathbf{w}_v \cdot \mathbf{z}_v + \mathbf{w}_m \cdot \mathbf{z}_m \tag{12}$$

where $\mathbf{w}_v, \mathbf{w}_m \in \mathbb{R}^{B\times1\times H\times W}$ hold different values for various positions, dependent on the largest attention scores in the visible and masked regions, respectively.

Loss Function. The network is optimized using the formula:

$$\mathcal{L}(\mathcal{R}_\phi) = \mathcal{L}_{rec} + \mathcal{L}_{per} + \mathcal{L}_{adv}, \tag{13}$$

where each term holds the same formula as in Eq. (4), except here the images $\hat{x}$ and x are in higher resolution.

4 Experiments

4.1 Experimental Details

Datasets. We evaluated the proposed PICFormer model with arbitrary mask types on various datasets, including: **FFHQ** [3], **ImageNet** [2], and **Places2** [1].

Metrics. Previous works [10], [14] have argued that it should *not* be required that the completed output be exactly the same as the original visible image, especially when holes are large. However, for the purpose of quantitative comparison, we report results on various image quality metrics, including traditional pixel-level Peak Signal-to-Noise Ratio (**PSNR**), patch-level Structural SIMilarity index (**SSIM**), the latest feature-level Learned Perceptual Image Patch Similarity (**LPIPS**) [79], and dataset-level Fréchet Inception Distance (**FID**) [80]. As our PICFormer provides multiple solutions for a masked image, we evaluated the top-1 and random of top-k results for each quantitative evaluation.

Training. Our model is trained in three stages: **a)** The code-shared codebook using the proposed restrictive CNN is first trained on a fixed resolution, *i.e.* $256{\times}256$. **b)** Then, we train the PICFormer-*Coarse* on the fixed resolution, *i.e.* 256×256, by inferring the embedded tokens via a highly expressive transformer architecture. **c)** The PICFormer-*Refined* is finally trained on higher resolution, *i.e.* 512×512. For codebook sizes, we used K=1024 for FFHQ, and K=16384 for the others. More network structures and implementation details are provided in Appendix A.

TABLE 1
Quantitative comparisons on Places2 [1] with free-form masks [44]. Without bells and whistles, the proposed method outperforms existing state-of-the-art methods on most metrics, especially for the feature-level metrics. Following established works, results are mainly reported on 256×256 resolution, except that our refined results are reported on 512×512 resolution.

	PSNR ↑			SSIM ↑			LPIPS ↓			FID ↓		
Mask Ratio	20-30%	30-40%	40-50%	20-30%	30-40%	40-50%	20-30%	30-40%	40-50%	20-30%	30-40%	40-50%
Single solution												
GL [9]$_{\text{SIGGRAPH'2017}}$	21.33	19.11	17.56	0.7672	0.6823	0.5987	0.1847	0.2535	0.3189	39.22	53.24	68.46
CA [10]$_{\text{CVPR'2018}}$	20.44	18.63	17.30	0.7652	0.6906	0.6133	0.1948	0.2490	0.3064	30.21	40.28	53.38
DeepFillv2 [54]$_{\text{ICCV'2019}}$	23.58	21.50	19.94	0.8319	0.7712	0.7074	0.1234	0.1639	0.2079	23.18	28.87	35.21
HiFill [55]$_{\text{CVPR'2020}}$	22.54	20.15	18.48	0.7838	0.7057	0.6194	0.1632	0.2258	0.3053	26.89	38.40	56.24
CRFill [58]$_{\text{ICCV'2021}}$	24.38	21.95	20.44	0.8476	0.7983	0.7217	0.1189	0.1597	0.1993	17.58	23.05	29.97
TFill [29]$_{\text{CVPR'2022}}$	25.10	22.89	21.22	**0.8686**	0.8063	0.7391	0.0918	0.1328	0.1796	**15.28**	19.99	25.88
Muliple solutions												
PIC [14]$_{\text{CVPR'2019}}$	24.44	22.32	20.71	0.8520	0.7850	0.7119	0.1183	0.1666	0.2245	21.62	29.59	41.60
CoMoGAN [62]$_{\text{ICLR'2021}}$	24.67	22.20	20.20	0.8517	0.7971	0.7155	0.1177	0.1648	0.2460	29.20	28.99	29.71
ICT [12]$_{\text{ICCV'2021}}$	24.53	22.84	21.11	0.8599	0.7995	0.7228	0.1045	0.1563	0.1974	17.13	22.39	28.18
PUT [18]$_{\text{CVPR'2022}}$	25.03	23.12	21.52	0.8667	0.8023	0.7271	0.1027	0.1508	0.1916	17.27	21.58	26.34
PICFormer-*Coarse*, Top1	24.63	22.51	21.07	0.8463	0.7791	0.7146	0.1037	0.1502	0.1891	17.08	16.92	17.27
PICFormer-*Coarse*, Random	23.71	21.63	20.12	0.8441	0.7770	0.7122	0.1035	0.1478	0.1950	16.50	16.39	16.68
PICFormer-*Refined*, Top1	**25.28**	**23.19**	**21.56**	0.8658	**0.8135**	**0.7453**	**0.0852**	0.1255	0.1676	15.65	**15.37**	**15.46**
PICFormer-*Refined*, Random	24.26	22.08	20.63	0.8631	0.8059	0.7329	0.0854	**0.1249**	**0.1662**	15.72	15.46	15.83

Inference. As codebook learning is a separate offline process, only the bottom stages **b)** (PICFormer-*Coarse*) and **c)** (PICFormer-*Refined*) will be applied to infer *multiple* and *diverse* results. In particular, given a masked image, it will first be downsampled to the fixed resolution, *i.e.* 256×256, for diverse content generation. Note that as the refinement network is fully convolutional, we can upsample the coarse result to any original resolution, rather than fixing 512×512 resolution as in training. This enables our proposed model to process various images with arbitrary sizes.

4.2 Main Results

We first performed a thorough comparison of PICFormer to the following methods:

- **GL**[1] [9]$_{\text{SIGGRAPH'2017}}$: Globally and Locally, the first learning-based method for arbitrary regions.
- **CA**[2] [10]$_{\text{CVPR'2018}}$: Contextual Attention, the first method combining intra- and inter-images.
- **PIC**[3] [14]$_{\text{CVPR'2019}}$: Pluralistic Image Completion, the first work considering multiple solutions.
- **HiFill**[4] [55]$_{\text{CVPR'2020}}$: High resolution Fill, the first work aiming to 8k resolution image completion.
- **CoMoGAN**[5] [62]$_{\text{ICLR'2021}}$: Co-Modulation GAN, the completion work for huge holes.
- **CRFill**[6] [58]$_{\text{ICCV'2021}}$: Contextual Reconstruction Fill, the latest image completion for single solution.
- **ICT**[7] [12]$_{\text{ICCV'2021}}$ and **PUT**[8] [18]$_{\text{CVPR'2022, TPAMI'2024}}$: Image Completion with Transformer and Patch-based Un-quantized Transformer, the latest state-of-the-art works for pluralistic image completion.

For a fair comparison, we used their publicly available codes and released models on their GitHub. However, some methods are trained on different mask types, then we cannot provide an absolutely fair comparison.

Quantitative Comparison. Table 1 shows quantitative evaluation results on Places2 [1], in which the images were degraded by the free-form masks provided in the PConv [44] testing set. The mask ratio denotes the range of masking proportion applied to the images. The original mask ratios hold six levels, from 0 to 60%, increasing 10% for each level. Here, following ICT [12], we mainly compare the results on middle-level mask ratios.

Without extra bells and whistles, our model outperforms all existing approaches by a large margin. Compared with the state-of-the-art methods ICT [12], which also use a deep transformer architecture to predict the possible discrete tokens, our PICFormer achieves averaging relative 18.58% and 28.37% improvements for LPIPS and FID scores, respectively. The key difference is that *our model learns a compositionally flexible codebook in the feature domain, instead of using a pre-clustered palette at pixel-level*. Therefore, despite modeling a shorter sequence distribution, our method can achieve better image quality after decoding. While the recent PUT [18] also applied the vector quantizer and transformer for pluralistic image completion, PICFormer learns a better discrete representation through code-shared codebook learning and restrictive CNN, resulting in a significant improvement.

The proposed PICFormer also produces competitive or better results to our conference version TFill [29], which is trained to generate a single "best" solution that matches the ground truth. Although our paired evaluations on PSNR, SSIM, and LPIPS become gradually worse for larger holes, it is worth noting that our FID scores, measuring the dataset-level distribution, remained about the same on different mask ratios. This suggests that *while our completed results do not exactly match the corresponding ground truth instances, they fit well to the dataset distribution.*

[1] https://github.com/satoshiiizuka/siggraph2017_inpainting
[2] https://github.com/JiahuiYu/generative_inpainting
[3] https://github.com/lyndonzheng/Pluralistic-Inpainting
[4] https://github.com/Atlas200dk/sample-imageinpainting-HiFill
[5] https://github.com/zsyzzsoft/co-mod-gan
[6] https://github.com/zengxianyu/crfill
[7] https://github.com/raywzy/ICT
[8] https://github.com/liuqk3/PUT

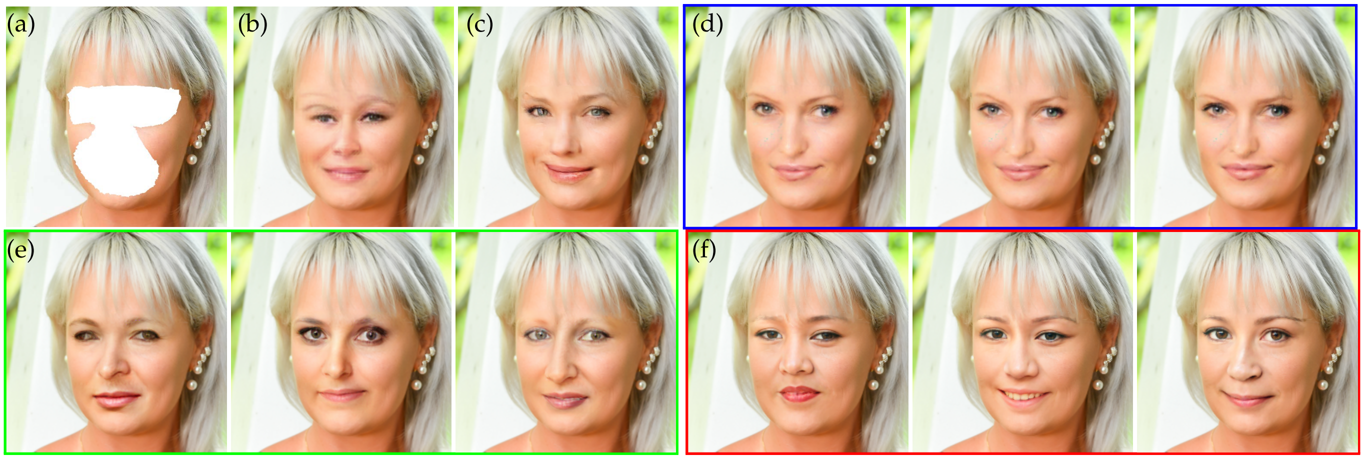
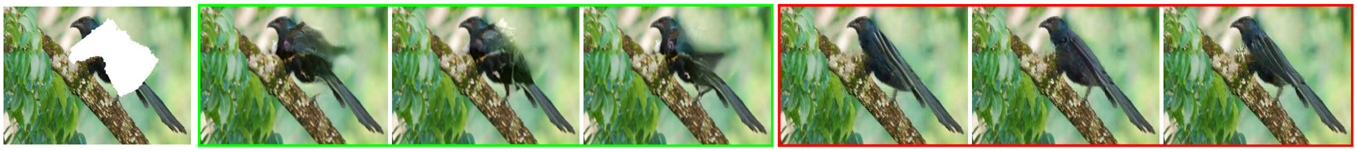


Fig. 9. **Comparisons with existing works on FFHQ [3]**. (a) Masked image. (b) CA [10] and (c) DeepFillv2 [54] generate one single solution. (d) PIC [14] provides multiple results but with limited diversity. (e) While ICT [12] improves the diversity, the generated images tend to be of reduced quality. (f) PICFormer achieves better image quality and larger diversity. More diverse solutions are presented as **animations** in the last term. Best viewed in Adobe Reader. Another 100 examples are provided in the Appendix.



(a) Masked Input (b) ICT [12]$_{\text{ICCV'2021}}$ (c) Ours

Fig. 10. **Comparisons with existing works on ImageNet [2]**. While ICT [12] provides diverse results, the heavily missed semantic content is hard to be met. In contrast, PICFormer provides some reasonable guesses for the large regions. More diverse solutions are presented as **animations** in the last columns of each case. Best viewed in Adobe Reader.

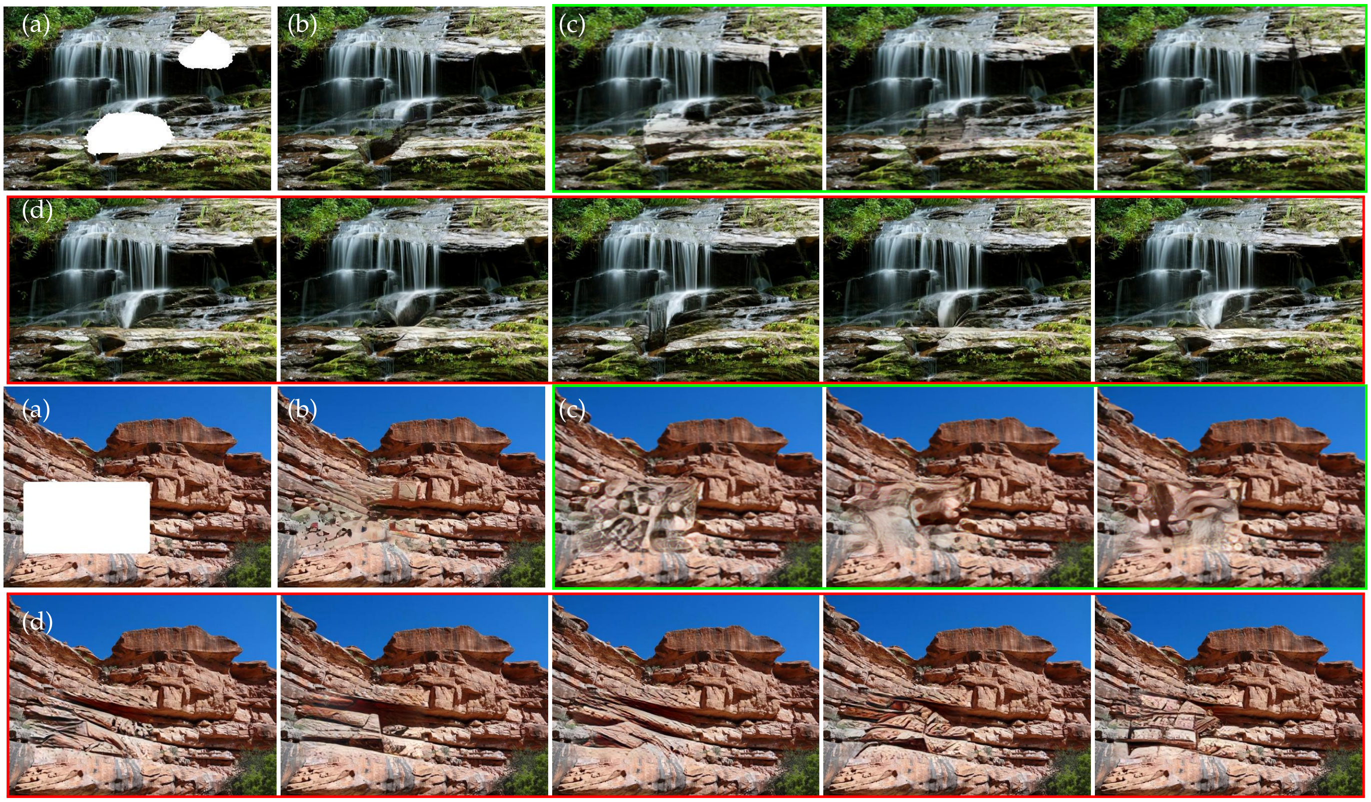


Fig. 11. **Comparisons with existing works on Places2 [1]**. (a) Masked image. (b) HiFill [55] generates only one solution. (c) While ICT [12] provides multiple solutions, the image quality is worse for large holes. (d) Our PICFormer generates high-fidelity pluralistic results. More diverse solutions are presented as **animations** in the final examples. Best viewed in Adobe Reader.

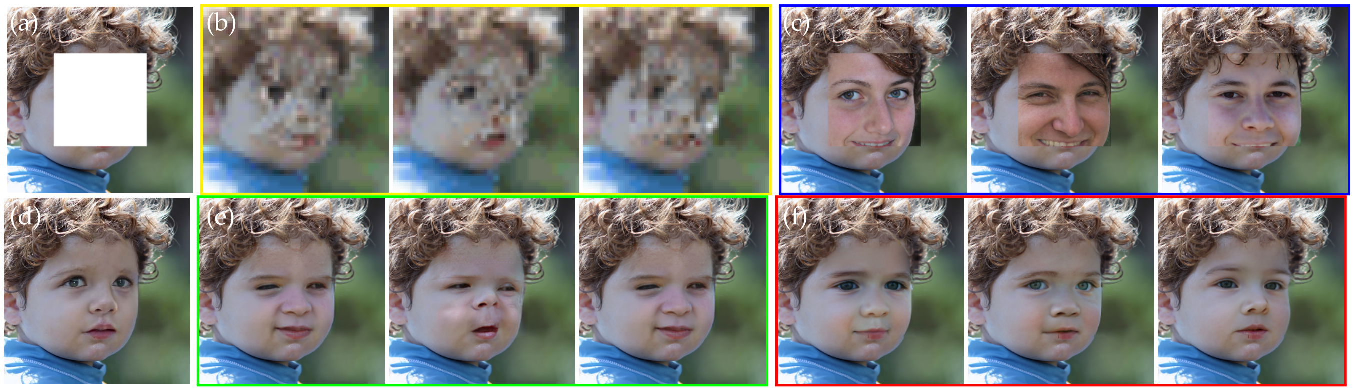


Fig. 12. **Comparison of training with different token embedding methods for transformer**. (a) Masked input. (b) Results of iGPT [16]. (c) Results of VQGAN [26]. (d) The single solution of our conference version TFill [29]. (e) Results of the latest ICT [12]. (f) With a better discrete codebook available, our method provides different sizes and colors of eyes and mouth.

TABLE 2
The trade-off between diversity and quality. The larger LPIPS denotes larger diversity between two generated images. All scores are reported on 256×256 resolution for the center-masked FFHQ images.

Method	LPIPS ↑	FID ↓
PIC [14]$_{\text{CVPR'2019}}$	0.024	6.43
UCTGAN [11]$_{\text{CVPR'2020}}$	0.036	5.12
Co-Mod-GAN [62]$_{\text{ICLR'2021}}$	0.020	1.88
ICT [12]$_{\text{ICCV'2021}}$	0.053	4.42
PUT [18]$_{\text{CVPR'2022}}$	0.056	3.98
PICFormer-*Coarse*	**0.062**	1.53
PICFormer-*Refined*	0.058	**1.41**

TABLE 3
The effect of different token representations on FFHQ dataset. "Mem" denotes the memory (GB) cost during testing, and "Time" is the average testing time (s) for one center-masked image.

Method	LPIPS ↓	FID ↓	Mem ↓	Time ↓
IGPT [16]$_{\text{ICML'2020}}$	0.609	148.42	3.16	26.45
VIT [70]$_{\text{ICLR'2021}}$	0.062	5.09	1.16	0.167
VQGAN [26]$_{\text{CVPR'2021}}$	0.226	11.92	2.36	4.29
ICT [12]$_{\text{ICCV'2021}}$	0.061	4.24	3.87	152.48
TFill-*Coarse* [29]$_{\text{CVPR'2022}}$	0.057	3.63	**1.15**	**0.02**
PICFormer-*Coarse*, Top1	**0.042**	2.19	3.83	0.03
PICFormer-*Coarse*, Random	0.044	**1.53**	3.83	0.03

Qualitative Comparison. The qualitative results are visualized in Figs. 9 to 11 for faces, objects, and natural scenes, respectively. Our model PICFormer achieves good results even under challenging scenarios. In Fig. 9, we can see that PICFormer not only fills in reasonable content with visually realistic appearance but also provides multiple and diverse choices for the masked face. In Fig. 10, we further evaluated PICFormer on a more challenging ImageNet dataset. For the state-of-the-art ICT [12], although it can generate multiple and diverse results, it had some difficulty creating plausible completions for arbitrary animals. In contrast, our PICFormer provides multiple results for heavily masked animals. Finally, the comparison is conducted on natural scenes in Fig. 11. While some existing approaches can generate visually reasonable results for background completion, most are geared towards providing only a single result. While ICT [12] can provide multiple and diverse results, it appears to suffer from reduced quality. In comparison, our PICFormer provides more diverse plausible results.

Diversity Comparison. Following the existing work [11], [12], [14], [18], we use the LPIPS to estimate the diversity score between the completed results on FFHQ. In particular, for each center-masked image, we produce 50 pairs of samples and calculate their paired feature distance. We compare various methods in Table 2. The PIC [14] and UCTGAN [11] carefully pursue the balance between diversity and quality, resulting in unsatisfied results for both metrics. While Co-Mod-GAN [62] achieves high-quality completion results (1.88 FID), the diversity (0.020 LPIPS) is limited, as their model is designed for large holes, and the diversity is also sampled from a fixed distribution. In contrast, ICT [12] learns to estimate the underlying distribution through the likelihood model, which achieves large diversity (0.053 LPIPS), but with limited quality (4.42 FID) due to the downsampled *pixel-level* representation. PUT [18], [19] improves both using the learned codebook presentation in *feature-level*. Compared with these methods, our PICFormer estimates the underlying prior distribution in a discrete space with a semantically rich codebook, resulting in a large diversity (*10.7% relative improvement*), while maintaining high image quality (*25% relative improvement*). This suggests that once a good representation is provided, the underlying prior distribution is easier to achieve.

4.3 Ablation Studies

We ran comprehensive ablation studies to analyse the effectiveness of each key point presented in our PICFormer. Results are reported on FFHQ dataset and shown in Tables 3 to 5 and Figs. 12 to 14.

Effect of Token Representation. We first investigated the influence of the different token embedding methods in Table 3 and Fig. 12. Here, to highlight our key target that building a high-quality image completion system, we only report the results on the second stage, leaving the comparison of offline trained image reconstruction in Appendix B.

In Table 3, all methods utilize an iGPT-based [16] transformer to predict the tokens. iGPT downsamples the image to a fixed scale, *i.e.* 32 × 32, and embeds each *pixel* to a token. While this may not impact the classification [63], it

TABLE 4
The effect of various attention layers. "center" and "random" denote mask types. These attention layers were implemented within our refinement framework, while using the same content generator.

	LPIPS↓		FID↓	
Mask Type	center	random	center	random
SA [78]$_{\text{ICML'19}}$	0.0584	0.0469	3.62	2.69
CA [10]$_{\text{CVPR'2018}}$	0.0608	0.0443	3.86	2.66
SLTA [14]$_{\text{CVPR'2019}}$	0.0561	0.0452	3.61	2.64
Ours-AAL	**0.0533**	**0.0412**	**3.50**	**2.57**

has a large negative effect on image generation (Fig. 12(b)). Furthermore, it cannot generate semantically consistent content due to the single-directional attention module. ICT [12] achieved diverse reasonable content through the bidirectional attention module, along with a 3× super-resolution network. Although the large diversity is met, the generated image is blurry (Fig. 12(e)). In contrast, VIT [70] embeds *each patch in a token*, which can achieve relatively good results. However, some details are perceptually poor. Finally, VQGAN [26] employs a large RF CNN to embed the image. It generates a visually realistic completion, but when pasted to the original input, there is an obvious gap between generated and visible pixels (Fig. 12 (c)). Using the same transformer architecture, our PICFormer outperformed these models, even by using only coarse results as shown in Table 3. Compared with the conference version TFill [29], the new PICFormer trained on discrete space seems to be able to infer more reasonable content. We believe this is because *a more compact discrete space is much easier for distribution modeling* and transformer learning involves optimizing a log-likelihood function, instead of seeking a balance in the adversarial learning in TFill. Interestingly, our random results also achieved higher image quality than the deterministic result in TFill. Even more surprisingly, our random sampling results led to better FID scores than all other methods. This phenomenon suggests that *the distribution inferred by our model is close to the true data distribution*, as our randomly sampled solutions fit well to it, and we are able to avoid out-of-distribution noisy samples.

Effect of Attention Aware Layer: An evaluation of our proposed *AAL* is shown in Table 4. Here, the ablation study was performed on our original single "best" solution, *i.e.* TFill model, and then we directly used the best *AAL* on refinement network for multiple and diverse solutions, *i.e.* PICFormer. As can be seen, even using the same content, the proposed AAL reduces LPIPS and FID scores by averaging relative 6.0% and 2.8%, over the existing works [10], [14], [78]. This is likely due to our AAL selects features based on the largest attention scores, using weights *dynamically mapped* during inference, instead of depending on *fixed* weights to copy features as in PIC [14].

The qualitative comparison is visualized in Fig. 13. CA [10], PIC [14], and CRFill [58] used different context attention in image completion. Here, we directly use their public models for visualization. As can be seen in Fig. 13, these methods cannot handle large holes. While ours-*SA* used the good but lower-resolution (256 × 256) coarse content from ours-*Coarse*, the mouth exhibits artifacts with inconsistent color. Our-*AAL* shows no such artifacts.

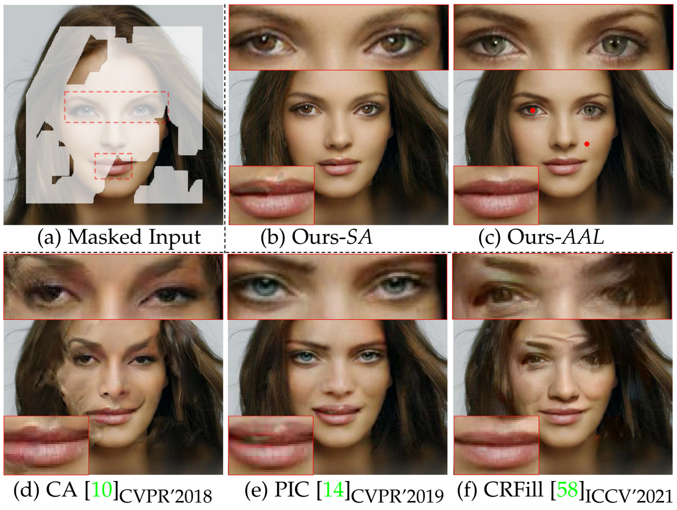

(a) Masked Input (b) Ours-*SA* (c) Ours-*AAL*
(d) CA [10]$_{\text{CVPR'2018}}$ (e) PIC [14]$_{\text{CVPR'2019}}$ (f) CRFill [58]$_{\text{ICCV'2021}}$

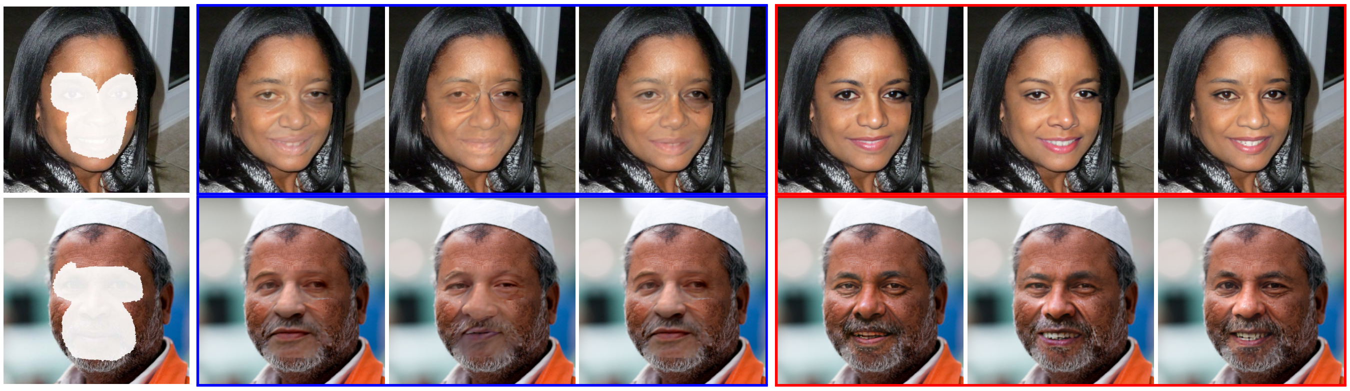
Fig. 13. **Results with different attention modules** in various methods. Our attention-aware layer is able to adaptively select the features from both visible and generated content.

TABLE 5
The effect of different sampling strategies. Top-$\mathcal{K}$ is the number of candidates. "Autoregressive" sampling needs to predict each token one-by-one via an expensive loop. "One-time" denotes to independently sample all tokens at one time. Here, the LPIPS is for diversity as in [12], [14], where larger value denotes lager diversity.

Method	Numbers	LPIPS ↑	FID ↓	Time ↓
Autoregressive	Top-1	-	5.60	3.532
	Top-20	0.073	5.59	
	Top-40	0.097	6.53	
	Top-100	**0.151**	6.26	
One-time	Top-1	-	2.19	**0.033**
	Top-20	0.062	**1.53**	
	Top-40	0.088	1.98	
	Top-100	0.124	1.77	

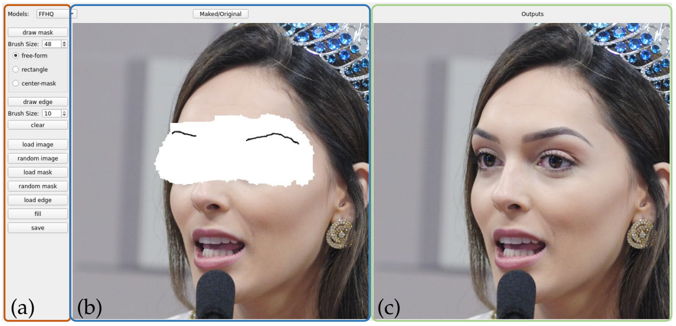
Effect of sampling strategy. The autoregressive sampling is a default setting in most existing discrete token-based image synthesis models [12], [26]. While they achieved excellent performance, the running time is ruinously expensive (average 22.32s/img for iGPT [16] and 131.32s/img for ICT [12] on an NVIDIA 3090 GPU). Is sequence sampling necessary for image completion? To answer this question, we ran several comparisons in Table 5 and Fig. 14. Here, except for different sampling strategies, we used the same trained model for the evaluation. Compared to autoregressive sampling, simultaneous sampling not only achieves more impressive results in our setting but also runs much faster with more than 100x speed-up. This is quite surprising. Our conjecture is that the transformer has learned the global context well within the image, and the sampling is appropriately conditioned on the visible regions.

Effect of sampling numbers. We also evaluated our model with different numbers of candidates. For this experiment, we first selected the top-$\mathcal{K}$ candidates from the predicted token distribution. We then sample the tokens based on their confidence scores. As can be seen in Table 5, more candidates result in larger diversity, but with worse image quality, which is still a trade-off.



(a) Masked Input (b) Ours Sequence (c) Ours One-time

Fig. 14. **Comparison of different sampling strategies during testing.** As our training directly predicts all tokens at one time, instead of sequentially depending on the previous scanning line, the sequential generation in our model performs worse than sampling at one time.



Models: FFHQ

Brush Size: 48

● free-form

○ rectangle

○ center-mask

Brush Size: 10

Fig. 15. **A screenshot of our interface for free-form image editing.** (b) and (c) are the original image and the modified output, respectively. (a) is the control function panel for user input.

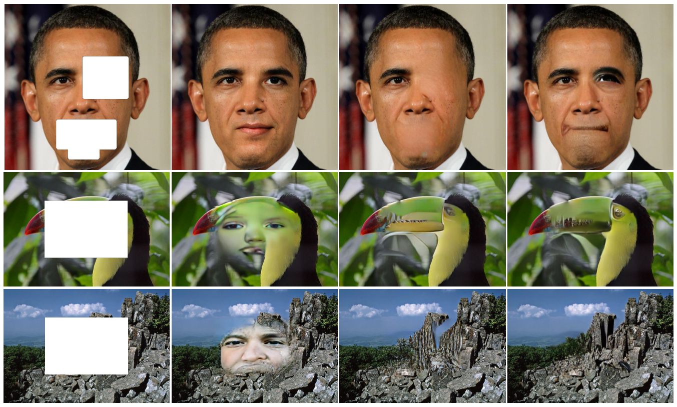
5 Application

Our trained model can be applied to a wide range of applications, including object removal and free-form image editing, see examples in Appendix Sections C and D.

Image Editing Interface We designed a real-time mask-sketch-based user interface (Fig. 15), enabling image modifications via masks and auxiliary sketches. The control panel (Fig. 15 (a)) consists of some necessary tools such as model selection, image selection, manual input mask, sketch, etc. For a given masked image (Fig. 15 (b)), our system generates completed images with diverse results in real-time on a GPU, by simply clicking the "Fill" button. Users can then select the best result according to their preferences.

6 Conclusion

In this paper, we have presented a novel framework for pluralistic image completion that produces multiple and diverse plausible results for a single masked image. Unlike recent vision transformer models that either use shallow projections or large receptive fields for token representation, our *code-shared* codebook leaning through the *restrictive CNN projection* provides the necessary separation between explicit *global* attention modeling and implicit *local* patch correlation that leads to substantial improvement in results. Through comprehensive experiments and thorough ablation studies, we have demonstrated that it is easier to tame a transformer to infer the correct tokens for missing regions by *learning a compact and expressive token representation.* We also introduced a novel attention-aware layer that adaptively balances the attention for visible and masked regions, further improving the completed image quality.



(a) Masked Input (b) Ours FFHQ (c) Ours ImageNet (d) Ours Places2

Fig. 16. **Examples of models trained on different datasets.** We test the model on website images. The special model works well only for a special dataset, instead of completing arbitrary images with one model.

Limitations. Although PICFormer provides diverse plausible results for a masked image, we need to train different models for different data types, *e.g.* faces, animals, objects, and natural scenes. In Fig. 16, we evaluated the trained model on natural images from websites. As can be seen, the model worked well when images contained the general content found in the dataset, but failed when tested on out-of-distribution images. Therefore, a long-term goal is to train a general codebook, and then tame a network to generate reasonable content for any arbitrary images. The latest diffusion-based approaches [45], [46] is able to achieve this goal by learning the priors from billions of images.

Acknowledgments

This research was supported by Monash FIT Grant. This study was also supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).

References

[1] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

[2] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[5] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, vol. 2, 2003, pp. II–II.

[6] C. Zheng, D.-S. Dao, G. Song, T.-J. Cham, and J. Cai, "Visiting the invisible: Layer-by-layer completed scene decomposition," *International Journal of Computer Vision*, vol. 129, no. 12, pp. 3195–3215, 2021.

[7] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF IEEE International Conference on Computer Vision (ICCV) Workshops*, 2019.

[8] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 2536–2544.

[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 107, 2017.

[10] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5505–5514.

[11] L. Zhao, Q. Mo, S. Lin, *et al.*, "Uctgan: Diverse image inpainting based on unsupervised cross-space translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 4692–4701.

[13] W. Wang, L. Niu, J. Zhang, X. Yang, and L. Zhang, "Dual-path image inpainting with auxiliary gan inversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 421–11 430.

[14] C. Zheng, T.-J. Cham, and J. Cai, "Pluralistic image completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] C. Zheng, T.-J. Cham, and J. Cai, "Pluralistic free-form image completion," *International Journal of Computer Vision*, vol. 129, no. 10, pp. 2786–2805, 2021.

[16] M. Chen, A. Radford, R. Child, *et al.*, "Generative pretraining from pixels," in *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, 2020, pp. 1691–1703.

[17] Y. Yu, F. Zhan, R. Wu, *et al.*, "Diverse image inpainting with bidirectional and autoregressive transformers," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[18] Q. Liu, Z. Tan, D. Chen, *et al.*, "Reduce information loss in transformers for pluralistic image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 347–11 357.

[19] Q. Liu, Y. Jiang, Z. Tan, *et al.*, "Transformer based pluralistic image completion with reduced information loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[20] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pretraining of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[23] C. Saharia, W. Chan, S. Saxena, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 213–229.

[25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.

[26] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 873–12 883.

[27] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 392–30 400, 2021.

[28] S. Zheng, J. Lu, H. Zhao, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6881–6890.

[29] C. Zheng, T.-J. Cham, J. Cai, and D. Phung, "Bridging global context interactions for high-fidelity image completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 512–11 522.

[30] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "Mat: Mask-aware transformer for large hole image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 758–10 768.

[31] C. Zheng and A. Vedaldi, "Online clustered codebook," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 798–22 807.

[32] M. Hu, C. Zheng, Z. Yang, *et al.*, "Unified discrete diffusion for simultaneous vision-language generation," in *The Eleventh International Conference on Learning Representations*, 2023.

[33] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (TOG)*, vol. 26, p. 4, 2007.

[34] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.

[35] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics.," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, IEEE, vol. 1, 2003, pp. 305–312.

[36] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003.

[37] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[38] J. Jia and C.-K. Tang, "Inference of segmented color and texture description by tensor voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 771–786, 2004.

[39] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (ToG)*, vol. 28, p. 24, 2009.

[40] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling, "Mask-specific inpainting with deep neural networks," in *Proceedings of the German Conference on Pattern Recognition*, Springer, 2014, pp. 523–534.

[41] J. S. Ren, L. Xu, Q. Yan, and W. Sun, "Shepard convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 901–909.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.

[44] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[45] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 461–11 471.

[46] T. Wu, C. Zheng, and T.-J. Cham, "Panodiffusion: 360-degree panorama outpainting via diffusion," in *The Twelfth International Conference on Learning Representations*, 2024.

[47] R. Suvorov, E. Logacheva, A. Mashikhin, *et al.*, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2149–2159.

[48] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 5892–5900.

[49] Y. Song, C. Yang, Z. Lin, *et al.*, "Contextual-based image inpainting: Infer, match, and translate," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[50] T. Portenier, Q. Hu, A. Szabo, S. A. Bigdeli, P. Favaro, and M. Zwicker, "Faceshop: Deep sketch-based face image editing," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 99, 2018.

[51] Y. Jo and J. 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)*, 2019.

[52] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 3.

[53] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-net: Image inpainting via deep feature rearrangement," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[54] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4471–4480.

[55] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7508–7517.

[56] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 1–17.

[57] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[58] Y. Zeng, Z. Lin, H. Lu, and V. M. Patel, "Cr-fill: Generative image inpainting with auxiliary contextual reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 14 164–14 173.

[59] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Satoh, "Image inpainting guided by coherence priors of semantics and textures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6539–6548.

[60] M. Suin, K. Purohit, and A. N. Rajagopalan, "Distillation-guided image inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2481–2490.

[61] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, and J. Liao, "Pd-gan: Probabilistic diverse gan for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9371–9381.

[62] S. Zhao, J. Cui, Y. Sheng, *et al.*, "Large scale image completion via co-modulated generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[63] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[64] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6309–6318.

[65] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 14 866–14 876.

[66] P. Esser, R. Rombach, A. Blattmann, and B. Ommer, "Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis," in *Proceedings of the International Conference on Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.

[67] J. Yu, X. Li, J. Y. Koh, *et al.*, "Vector-quantized image modeling with improved VQGAN," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=pfNyExj7z2.

[68] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, "Autoregressive image generation using residual quantization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[69] J. Peng, D. Liu, S. Xu, and H. Li, "Generating diverse structure for image inpainting with hierarchical vq-vae," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 775–10 784.

[70] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[71] D. A. Hudson and L. Zitnick, "Generative adversarial transformers," in *International conference on machine learning (ICLR)*, PMLR, 2021, pp. 4487–4499.

[72] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two transformers can make one strong gan," *arXiv preprint arXiv:2102.07074*, 2021.

[73] B. Wu, C. Xu, X. Dai, *et al.*, "Visual transformers: Token-based image representation and processing for computer vision," *arXiv preprint arXiv:2006.03677*, 2020.

[74] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," *International Conference on Computer Vision (ICCV)*, 2021.

[75] C. Zheng, T.-L. Vuong, J. Cai, and D. Phung, "Movq: Modulating quantized vectors for high-fidelity image generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 412–23 425, 2022.

[76] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[77] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[78] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning (ICML)*, PMLR, 2019, pp. 7354–7363.

[79] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[80] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6626–6637.

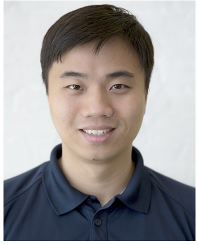
[81] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, "Photo-sketching: Inferring contour drawings from images," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 1403–1412.

Chuanxia Zheng (Member, IEEE) is currently a *Marie Skłodowska-Curie Actions (MSCA) Fellow* and a postdoctoral researcher in VGG at the University of Oxford. He received his PhD in 2021 from Nanyang Technological University, Singapore, and obtained the *NTU Outstanding PhD thesis award*. Chuanxia's research interests are broadly in computer vision and machine learning, especially for Generative AI in multi-modality (1D, 2D, 3D, and 4D) generation. He is author of more than 20 peer-reviewed publications in the top machine vision and artificial intelligence conferences and journals. His current and past research services include serving as Area Chair for ACM Multimedia, organizing the Workshop for CVPR, and being the reviewer for TPAMI, IJCV, IEEE T-IP, T-MM, and CVPR, ICCV, ECCV, NeurIPS, ICML, ICLR, SIGGRAPH.

Guoxian Song is currently a Research Scientist at ByteDance, CA, USA. He obtained a CS Ph.D. degree in 2021 from Nanyang Technological University, Singapore. Before that, he obtained a B.S degree from the School of Mathematics, University of Science and Technology of China. His interests are computer vision and graphics including image-based 3D face reconstruction/analysis, gaze estimation, human portrait relighting etc.

Tat-Jen Cham is an Associate Professor in the College of Computing and Data Science, Nanyang Technological University, Singapore. He received his BA in Engineering in 1993 and his PhD in 1996, both from the University of Cambridge. He was previously a Jesus College Research Fellow in Science (1996-97), and a research scientist at DEC/Compaq Research Labs in Cambridge, MA, USA (1998-2021). Tat-Jen received best paper prizes at PROCAMS'05, ECCV'96 and BMVC'94, and is an inventor on eight patents. Tat-Jen has been the principal investigator on projects that include those based in the Rehabilitation Research Institute of Singapore (RRIS), the Singtel Cognitive & AI Lab (SCALE@NTU), Singapore-ETH Centre's Future Cities Lab, and the NRF BeingThere / BeingTogether Centres on 3D Telepresence. His current and past research services include being an Associate Editor for IEEE T-MM, CVIU and IJCV, as well as an Area Chair for many CVPR, ICCV and ECCV conferences. He was also a General Chair for ACCV 2014. Tat-Jen's research interests are broadly in computer vision and machine learning, with a focus on deep learning generative methods that can exploit semantic and contextual cues, for applications such as 3D telepresence and metaverses.

Jianfei Cai (S'98-M'02-SM'07-F'21) received his PhD degree from the University of Missouri-Columbia. He is currently a Professor at Faculty of IT, Monash University, where he had served as the inaugural Head for the Data Science & AI Department. Before that, he was Head of Visual and Interactive Computing Division and Head of Computer Communications Division in Nanyang Technological University (NTU). His major research interests include computer vision, deep learning and multimedia. He has successfully trained 30+ PhD students with three getting NTU SCSE Outstanding PhD thesis award and one getting Monash FIT Graduate Research Student Excellence Award. He is a co-recipient of paper awards in ACCV, ICCM, IEEE ICIP and MMSP. He serves or has served as an Associate Editor for TPAMI, IJCV, IEEE T-IP, T-MM, and T-CSVT as well as serving as Area Chair for CVPR, ICCV, ECCV, IJCAI, ACM Multimedia, ICME, ICIP and ISCAS. He was the Chair of IEEE CAS VSPC-TC during 2016-2018. He had served as the leading TPC Chair for IEEE ICME 2012 and the best paper award committee chair & co-chair for IEEE T-MM 2020 & 2019. He is the leading General Chair for ACM Multimedia 2024, and a Fellow of IEEE.



Linjie Luo is a Research Scientist and Manager at ByteDance on computer graphics and computer vision. His work covers a wide range of areas including the creation, driving and rendering for virtual avatars, 3D scene understanding, reconstruction and tracking, which led to many successful tech transfers empowering commercial products, such as TikTok Avatars, Douyin's Landmark AR, Snapchat Landmarkers, Marker Tracking, 3D Stickers. Before joining ByteDance, Linjie was a Lead Research Scientist at Snap Inc. working on a variety of AR research projects and products. Even before that, Linjie worked as Research Scientist at Adobe Research on 3D scanning and 3D modeling. Linjie obtained PhD from Princeton University Computer Science Department in 2013 and bachelor degree with honors from Tsinghua University School of Software in 2007.

Dinh Phung (Member, IEEE) received the B.Sc. (first-class honor) and Ph.D. degrees in computer science from Curtin University, in 2001 and 2005, respectively. He is a Professor and Head of the Department of Data Science and AI at Monash University, Australia. He has won numerous best paper and research awards, published 250+ papers, and attracted over 20 million in funding in these areas and application domains such as NLP, computer vision, cybersecurity, digital health, and AI-enabled autism research. In 2020, he was Finalist of the prestigious Australian Museum Eureka Prize for Excellence in Data Science. He is the current editor-in-chief for the 3rd/living edition of the Encyclopedia in Machine Learning and Data Science and has been an Associate Editor for the Journal of Artificial Inteligence Research since 2021.

The supplementary materials are organized as follows:

1) A video to illuminate our work and interface https://drive.google.com/file/d/18rjUTPq_hpFjehbG4MFgkv0X6nXq7bVl/view?usp=sharing.
2) More results for free-form image completion on FFHQ dataset. We directly show 1-100 index from FFHQ without curated selection, which is available at https://drive.google.com/file/d/1KLrD4NIk5j-fnS1UGLyc8jd9bEiuabCM/view?usp=sharing.
3) Experiment details in Section A.
4) Results for more image completion and editing tasks in Sections B to D.

Appendix A
Experiment Details

Our quantizer is built upon the VQGAN[9], and the transformer architecture and refinement network is adapted from our conference version TFill[10].

In the stage **a)**, we use the quantizer as in VQGAN to embed an image into discrete space, except that we used the restrictive CNN to embed the features and subdivide the continuous features into 4 chunks and then quantize each chunk to the code vector. In practice, following the default setting in VQGAN, images are downsampled by a fixed factor of 16 in all experiments, *i.e.* from $256 \times 256 \times 3$ to a grid of discrete index with size $16 \times 16 \times 4$, where 4 is the multiple channels. The number of entries in the learned codebook is the same as the original VQGAN, *i.e.* 1024 tokens for FFHQ and 16,384 tokens for others, respectively. Note that, since the original features are subdivided into 4 channels, the corresponding dimensionality is reduced to $64 = 256/4$. All hyperparameters follow the VQGAN setting, and we trained all models with batch size 96 across 8 Tesla V100 GPUs with 40 epochs for stage **a)**.

All models in this paper have the same configuration for the stage **b)**: 24 layers, 16 attention heads, 1024 embedding dimensions, and 4096 hidden dimensions for the transformer. Here, the architecture is on the top of our conference version TFill. There is only one difference, that we simultaneously predict all discrete tokens at one time, instead of predicting the original continuous features. The training hyperparameters also follow the VQGAN setting, and we trained all models with batch size 128 across 8 Tesla V100 GPUs with 50 epochs on FFHQ and 30 epochs for other datasets.

The refinement architecture is adapted from the TFill refinement network, where a fully convolutional encoder-decoder architecture is implemented to process images on arbitrary resolutions. During the training, the model is trained to refine the results of VQ on 256×256 resolution, and the image will be downsampled with a factor of 8, *i.e.* 32×32 resolution, where an adaptive attention layer is further employed to copy high-frequence details from both visible and generated regions to masked regions. Once the model is converged, we fine-tune the model on 512×512 resolution. At the inference time, in principle, the refinement model can process images on arbitrary resolution due to the fully convolutional architecture.

TABLE A.1

Quantitative results between reconstructed validation split and original validation split on ImageNet [2] (50,000 images) and FFHQ [3] (10,000 images). "Num $\mathcal{Z}$" is the number of tokens in the codebook.

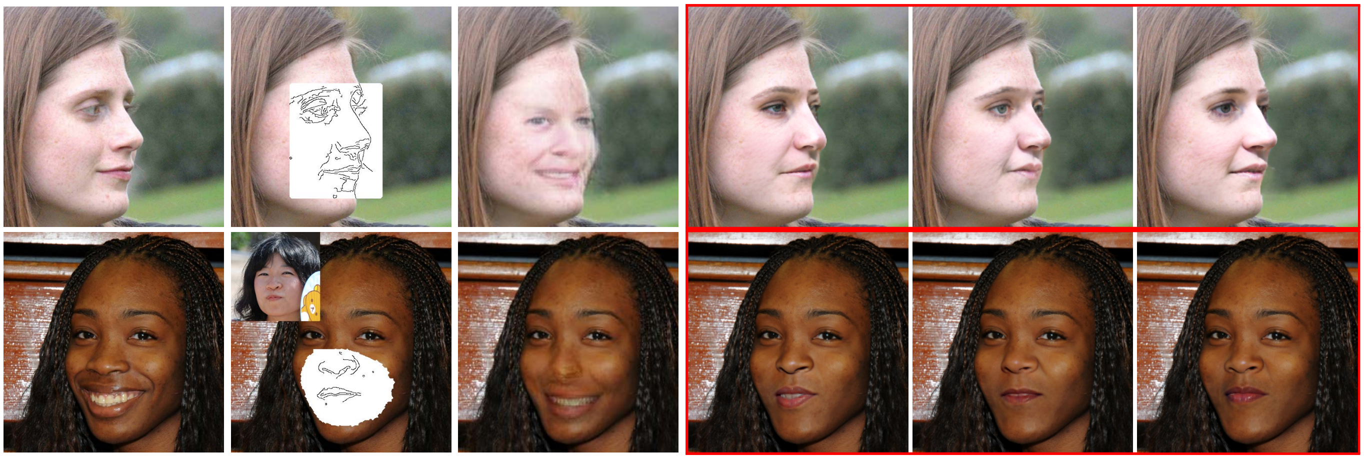
Model	Dataset	Latent Size	Num $\mathcal{Z}$	PSNR ↑	SSIM ↑	LPIPS ↓	rFID ↓
VQGAN [26]	FFHQ	16×16	1024	22.24	0.6641	0.1175	4.42
ViT-VQGAN [67]		32×32	8192	-	-	-	3.13
RQ-VAE [68]		8×8×4	2048	22.99	0.6700	0.1302	7.04
RQ-VAE [68]*		16×16×4	2048	24.53	0.7602	0.0895	3.88
Ours		16×16(×4)	1024	**25.28**	**0.7772**	**0.0688**	**2.96**
VQGAN [26]	ImageNet	16×16	1024	19.47	0.5214	0.1950	6.25
VQGAN [26]		16×16	16384	19.93	0.5424	0.1766	3.64
ViT-VQGAN [67]		32×32	8192	-	-	-	**1.28**
RQ-VAE [68]		8×8×16	16384	-	-	-	1.83
Ours		16×16(×4)	16384	**22.43**	**0.6750**	**0.1107**	1.30

Appendix B
Reconstruction Results

We compare our code shared codebook to the state-of-the-art methods in image reconstruction in Table A.1. Most instantiations of our model outperform baseline variants of previous state-of-the-art models. This includes the latest concurrent work RQ-VAE [68], which also represents images into multichannel's index. However, they obtain the multichannel representation in a recursive way to calculate the residual information in every loop, which takes more computational time. Furthermore, they also need to recursively predict multichannel indexes one-by-one, resulting in larger memory cost and expensive computational cost. Our model also achieves competitive results with the latest ViT-VQGAN [67], which includes higher resolution representation, larger encoder-decoder model, and codebook normalization. Besides, they also require a much larger model for the transformer model, due to the longer sequence.

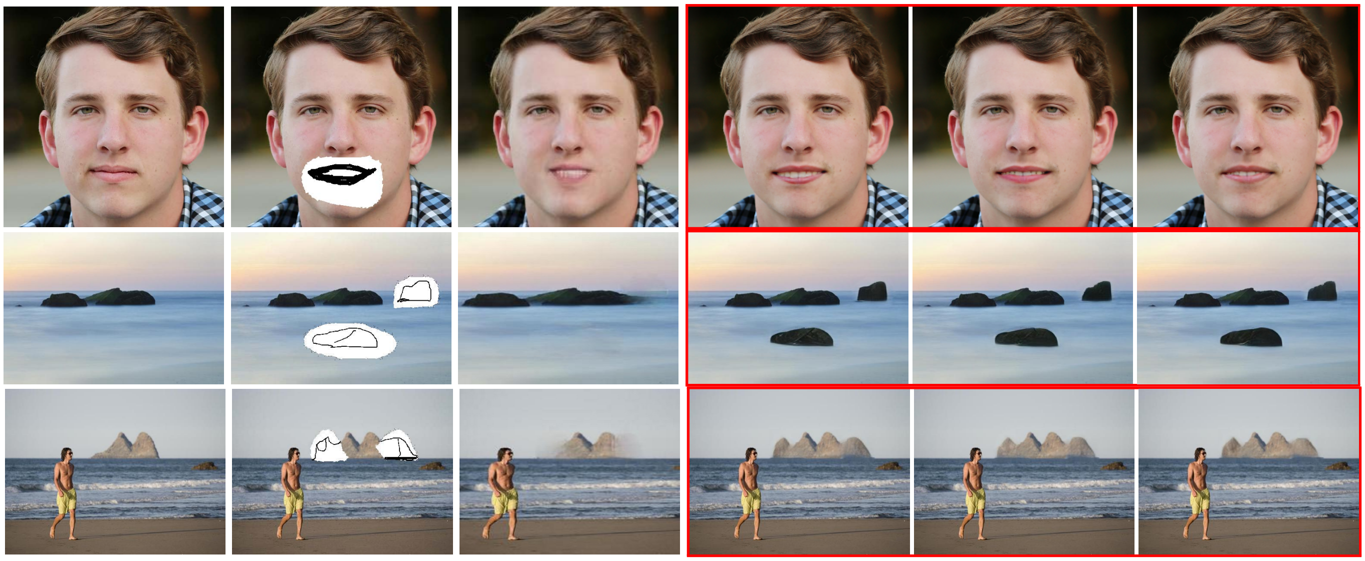
[9] https://github.com/CompVis/taming-transformers

[10] https://github.com/lyndonzheng/TFill



(a) Original (b) Masked input (c) EC [7] (d) Ours

Fig. B.1. **Comparisons of image completion given auxiliary edge information.** We can use the original canny edge to infer results under challenge scenarios. While the shape is fixed, the details are changed. Furthermore, we can also combine with other edges to recompose a new scene cheaply.



(a) Original (b) Masked input (c) EC [7] (d) Ours

Fig. B.2. **Completed results with hand-drawn sketches as auxiliary input information**. Our model works well with reasonable hand-drawn sketches. While the shape is guided by the sketch, our method still provides diverse results with different details.

Appendix C
Auxiliary Input

Our PICFormer is easily adapted to include auxiliary input guidance, such as simple user-drawn sketches. Here, we first transform the images in the dataset to sketches using a traditional Canny edge detector and the latest learning-based PhotoSketch [81]. During training, a proportion (we use 40%) of masked images contain corresponding sketches interposed within the masked regions. These are then passed through the encoder and transformer to generate realistic outputs.

We compared PICFormer to the state-of-the-art EdgeConnect [7] in Fig. B.1. Here, we use extracted sketches either from the corresponding image or from the other images. PICFormer outperforms EdgeConnect by providing better content and consistent appearance. When combined with sketches from other images, the proposed method is able to create multiple new scenes that adapt to the input guidance. In these instances, the diversity is more limited to changing local attributes, while the global structure has been established by the sketches.

Lastly, by combining our masking and sketch-based interface, users can freely edit images by masking the target regions and drawing on the corresponding sketches. We show several editing examples on faces and natural scenes in Fig. B.2, and a qualitative comparison with EdgeConnect [7] is provided. EdgeConnect cannot provide reasonable content, and exhibits large artifacts on imperfect manual sketches, suggesting that it has difficulty in adapting to arbitrary random manual sketches. Our proposed system mitigates this issue by quantizing manually drawn sketches to the closest tokens, resulting in only small artifacts.

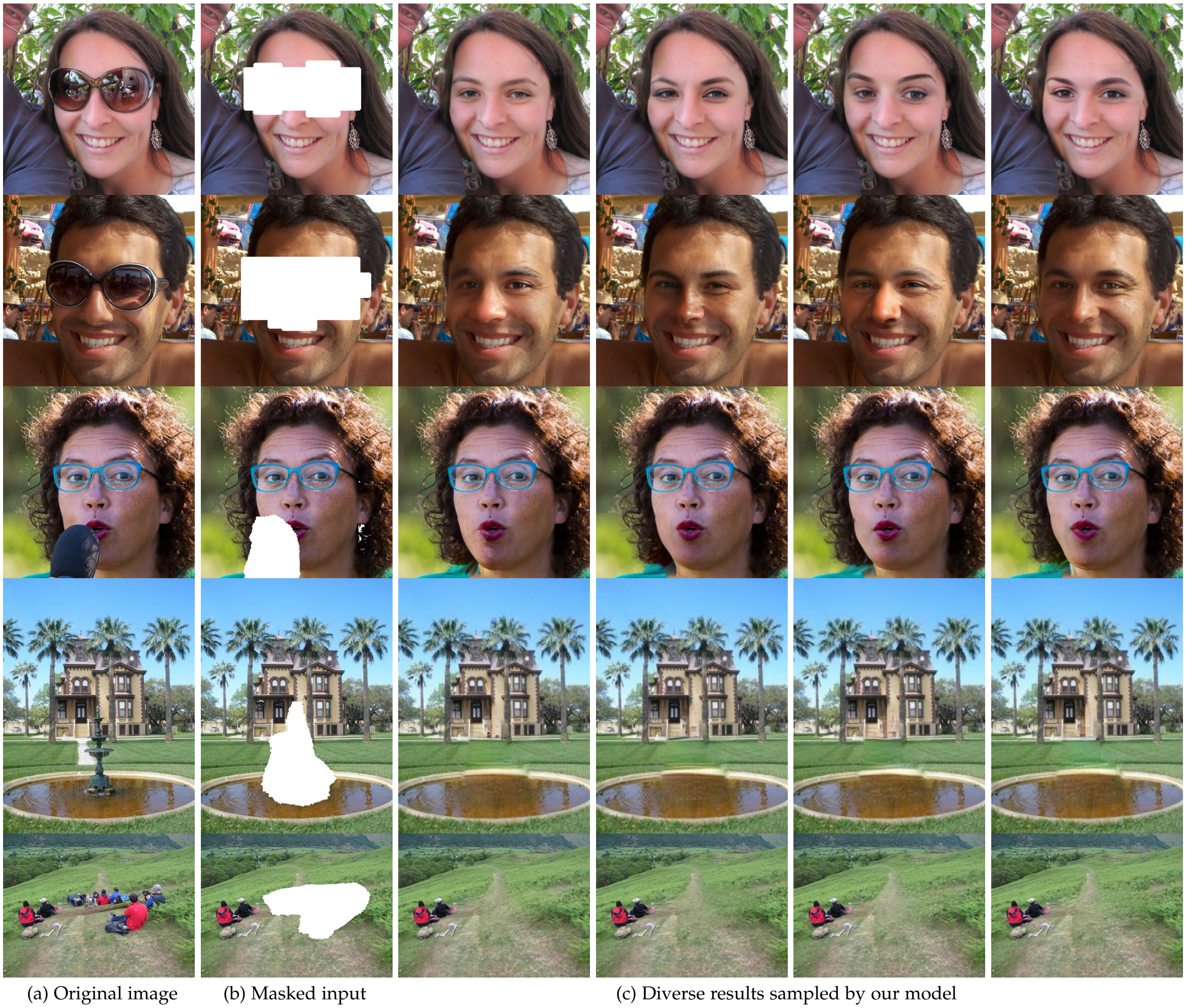


Fig. C.3. **Examples of object removal by PICFormer on faces and natural scenes**. The last column shows diverse results as **animations**. It will be more obvious to capture the difference between different solutions. If foreground objects are fully masked, our method fills in with background contents, because it only captures the background visible information.

Appendix D
Image Editing

With our designed interface as in Fig. 15, we can now freely edit an image by inputting a mask. The main applications include object removal and more advanced foreground object completion and manipulation.

Free-form image editing. Instead of filling background pixels into masks for object removal, it is more challenging to generate diverse plausible results for partially visible content. This needs the model to hallucinate new content based on what it has observed, rather than purely completing background textures within an image. As shown in Fig. B.2, through masking the mouth of a face, we can synthesize different target expressions. In particular, the generated mouths maintain the same shape as the guided masks, while achieve diverse visual appearances. In addition, the model also can generate different shapes for the mountains, after masking the target regions.

Object removal. Object removal is a related sample task in semantic image completion because it only requires copying and propagating similar background information to the masked regions. Since this is not the main of our paper, we just show a few completed examples in video and Fig. C.3. Our method generally works very well for large object removal by correctly inferring the content based on the partially visible context.