

# Outpainting by Queries

Kai Yao<sup>1,2</sup>, Penglei Gao<sup>1,2</sup>, Xi Yang<sup>2</sup>, Kaizhu Huang<sup>3</sup>, Jie Sun<sup>2</sup>, and Rui Zhang<sup>2</sup>

<sup>1</sup> University of Liverpool, Liverpool L69 7ZX, U.K.  
{Kai.Yao,P.Gao}@liverpool.ac.uk

<sup>2</sup> Xi'an Jiaotong Liverpool University, Suzhou, 215123, China

<sup>3</sup> Duke Kunshan University, Kunshan 215316, China

**Abstract.** Image outpainting, which is well studied with Convolution Neural Network (CNN) based framework, has recently drawn more attention in computer vision. However, CNNs rely on inherent inductive biases to achieve effective sample learning, which may degrade the performance ceiling. In this paper, motivated by the flexible self-attention mechanism with minimal inductive biases in transformer architecture, we reframe the generalised image outpainting problem as a patch-wise sequence-to-sequence autoregression problem, enabling query-based image outpainting. Specifically, we propose a novel hybrid vision-transformer-based encoder-decoder framework, named **Query Outpainting TRansformer (QueryOTR)**, for extrapolating visual context all-side around a given image. Patch-wise model's global modeling capacity allows us to extrapolate images from the attention mechanism's query standpoint. A novel Query Expansion Module (QEM) is designed to integrate information from the predicted queries based on the encoder's output, hence accelerating the convergence of the pure transformer even with a relatively small dataset. To further enhance connectivity between each patch, the proposed Patch Smoothing Module (PSM) re-allocates and averages the overlapped regions, thus providing seamless predicted images. We experimentally show that QueryOTR could generate visually appealing results smoothly and realistically against the state-of-the-art image outpainting approaches. Code is available at <https://github.com/Kaiseem/QueryOTR>.

**Keywords:** Image Outpainting, Transformer, Query Expanding

## 1 Introduction

Image outpainting, usually known as image extrapolation, is a challenging task that requires extending image boundaries by generating new visually harmonious contents with semantically meaningful structure from a restricted input image. It could be widely applied in the real world to enrich humans' social lives based on limited visual content, such as automatic creative image, virtual reality, and video generation [31]. Different from image inpainting [3,2,35,45], which could

---

K. Yao and P. Gao — Equal Contribution.

take advantage of visual contexts surrounding an inpainting area, generalised image outpainting should extrapolate the unknown regions in all directions around the sub-image. As the unknown pixels farther from the image borders are less constrained, they have a greater chance of accumulating expanded-errors or generating repetitive patterns than those closer to the borders. Consequently, the challenges of this task include: (a) determining where the missing features should be located relative to the output’s spatial locations for both nearby and faraway features; (b) guaranteeing that the extrapolated image has a realistic appearance with reasonable content and a consistent structural layout with the conditional sub-image; and (c) the borders between extrapolated regions and the original sub-image should be smooth and seamless.

Convolutional architectures have been proven successful for computer vision tasks nowadays. Existing image outpainting methods utilize kinds of variants of CNN-based methods to conduct image extrapolation. CNNs rely on inherent inductive biases to achieve effective sample learning, which may degrade the performance ceiling. Although the existing CNN-based outpainting methods achieve solid performance [40,44,43,22,31], they still suffer from blunt structures and abrupt colours when extrapolating the unknown regions of the images. The potential reason might be that the inductive biases of convolution in such CNN-based architectures are hard-coded in the form of two strong constraints on the weights: locality and weight sharing [6]. These constraints may degrade the model’s ability to represent global features and capture long-range dependencies.

Transformer architectures have competitive performance in areas such as image and video recognition. The transformer dispenses with the convolutional inductive bias by performing self-attention across embeddings of patches of pixels, which breaks through the limitation of capturing long-range dependencies. However, in the pure transformer, the model converges very slowly with a relatively small dataset [6]. On the ImageNet benchmark, Dosovitskiy et al. [9] developed the Vision Transformer (ViT) interpreting a picture as a sequence of tokens, which can achieve comparable image classification accuracy while requiring less computational budgets. ViT relies on globally-contextualized representation, in which each patch is attended to all patches of the same image, as opposed to local-connectivity in CNNs. ViT and its variants have shown promising superiority in modeling non-local contextual relationships as well as good efficiency and scalability, though they are still in their infancy. In light of the global interaction and the generation of distant features with conditional sub-image, these benefits could enhance image extrapolation in a beneficial fashion.

To better cope with image long-range dependencies and spatial relationships between predicted regions and conditional sub-images, we reconsider the outpainting problem as a patch-wise sequence-to-sequence autoregression problem inspired by the original transformer [41] in natural language processing. We develop a novel hybrid query-based encoder-decoder transformer framework, named **Query Outpainting TRansformer (QueryOTR)**, to extrapolate visual context all-side around a given image taking advantages of both ViT [9] and pure transformer [41] in the image outpainting task, as shown in Fig. S1. Specifically,

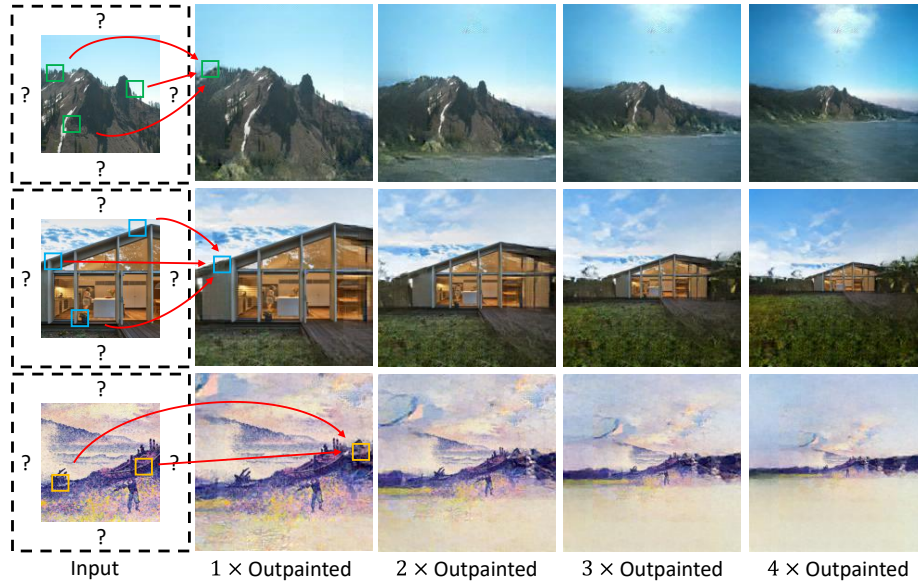


Fig. S1: Demonstration of recursive outpainting by our QueryOTR. Our method generates a sequence of extrapolated image patches by querying the sequence of input image patches, enabling a remarkable perceptual consistency.

we design two special modules, **Query Expansion Module (QEM)** and **Patch Smoothing Module (PSM)**, to conduct feature forecasting from the perspective of the query in the attention mechanism. In contrast to the query learning in pure transformer, our designed query in QEM is predicted by the stacked CNN-based blocks based on the output of the transformer encoder. The predicted query is easy to learn and has better flexibility by drawing on the advantages of CNNs’ inductive biases to accelerate query prediction converge in pure transformer for approximately three times faster than that without QEM in training, which is shown in Fig. S2(a). The developed PSM re-allocates the predicted patches around the conditional sub-image and averages the overlapping parts to make the generated image smoothly and seamlessly. Also, PSM contributes to alleviate the problem of checkerboard artifact caused by the independent procession among the output image patches. In this way, the model could focus more on the connections between each patch and enhance the representing ability as shown in Fig. S2(b) and (c). Our **QueryOTR** is the first hybrid transformer as a **sequence-to-sequence modeling**, which is able to extend image borders seamlessly and generate unseen images smoothly and realistically.

The main contributions of this work are three-fold:

- We rephrase the image outpainting problem as a patch-wise sequence-to-sequence autoregression problem and develop a novel hybrid transformer encoder-decoder framework, named **QueryOTR**, for query-based prediction

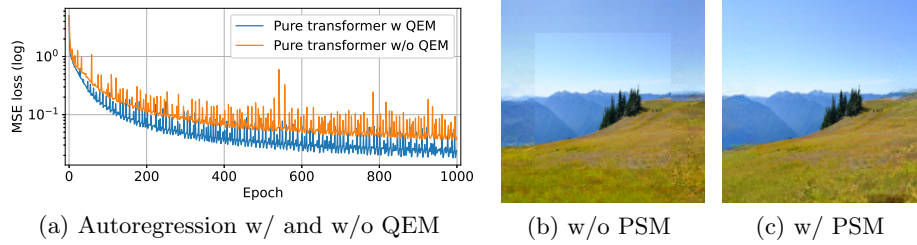


Fig. S2: (a) Training a pure transformer encoder-decoder with and without QEM to regress unseen image patches. QEM significantly speeds up the convergence (about 3.3 times faster than that without QEM: w/ QEM at 300 epoch v.s. w/o QEM at 1,000 epoch). (b) QueryOTR without PSM. (c) QueryOTR with PSM.

of extrapolated images, and minimization of degradation from the inductive biases in CNN-structures.

- We propose *Query Expansion Module* and *Patch Smoothing Module* to solve the slow convergence problem in pure transformers and to generate realistic extrapolated images smoothly and seamlessly.
- Experimental results show that the proposed method achieves state-of-the-art one-step and multi-step outpainting performance as compared to recent image outpainting methods.

## 2 Related Work

### 2.1 Image Outpainting

Generative Adversarial Networks (GANs) [12] have been widely applied in many research fields, such as image super-resolution, image synthesis, and image denoising [14,25,4,32,15]. Efforts have been made for image generation with GAN under certain conditions. Image extrapolation aims to generate the surrounding regions from the visual content, which can be considered as an image-conditioned generation task [16]. Sabini and Rusak [36] brought the image outpainting task into public attention with a deep neural network framework inspired by the image inpainting methods. This effort focused on enhancing the quality of generated images smoothly by using GANs and the post-processing methods to perform horizontal outpainting. Van et al. [40] designed a CNN-based encoder-to-decoder framework by using GAN for image outpainting. Wang et al. [43] proposed a Semantic Regeneration Network to directly learn the semantic features from the conditional sub-image. Han et al. [28] developed a 3-stage deep learning model with an edge-guided generative network to produce semantically consistent output from a small image input. Although these methods avoid the bias in the general padding and up-sampling pattern, they still suffer from blunt structures and abrupt colours issues, which tend to ignore the spatial and semantic consistency. To tackle these issues, Yang et al. [44] proposed a Recurrent Content

Transfer (RCT) block for temporal content prediction with Long Short Term Memory (LSTM) networks as the bottleneck. To increase the contextual information, Lu et al. [30] and Kim et. al. [22] rearranged the boundary region by switching the outer area of the image into its inner area. These latest models are based on convolutional neural networks. As global information is not well captured, they all have limitations in explicitly modelling long-range dependency.

## 2.2 Transformer

Recently, transformer has attracted much attention in computer vision. Transformer was first proposed to solve NLP tasks by replacing the traditional CNN and Recurrent Neural Network (RNN) structures [41]. The Self-Attention mechanism helps the model learn the global representation from the input which could improve the performance for basic visual feature extraction [41]. Jacob et al. [8] introduced a very deep network to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. It can be fine-tuned with just one additional output layer for better performance. ViT [9] is a convolution-free Transformer that conducts image classification over a sequence of image patches. The superiority of the Transformer architecture is presented in ViT fully utilizing the advantage of pretraining on large-scale datasets compared with the CNN-based methods. Many ViT-based variants also demonstrated the success in computer vision tasks [47, 19, 13], such as object detection [5], video recognition [1], and image synthesis [26]. Moreover, Liu et al. [29] proposed Swin Transformer to extend vision tasks for object detection and semantic segmentation. Gao et al. [11] designed a transformer-based framework for image outpainting with an encoder-decoder architecture. They used Swin Transformer which involved shifted window attention to bridge the windows of the preceding layer, which significantly enhanced modelling power as well as achieved lower latency.

## 3 Methodology

### 3.1 Problem Statement

Given an image  $x \in \mathbb{R}^{H \times W \times 3}$ , we aim to extrapolate outside contents beyond the image boundary with extra  $M$ -pixels. The generator will produce a visually convincing image  $\hat{x} \in \mathbb{R}^{(H+2M) \times (W+2M) \times 3}$ . Different from previous work which is almost based on convolutional operations, we rephrase the problem as a patch-wise sequence-to-sequence autoregression problem. In particular, we partition the image  $x$  into regular non-overlapping patches with the patch size  $P \times P$  ( $P$  is typically empirically set to 16), resulting in a sequence of patch tokens  $\{x_p^1, x_p^2, \dots, x_p^L\}$ , where  $x_p^i \in \mathbb{R}^{(P^2 \cdot 3)}$  and the sequence length is  $L = \frac{H \times W}{P^2}$ . Our goal is to predict the extra sequence  $\{x_p^{L+1}, x_p^{L+2}, \dots, x_p^{L+R}\}$  representing the extrapolated regions, where  $x_p^i \in \mathbb{R}^{(P^2 \cdot 3)}$  and the expanded sequence length is  $R = \frac{(H+2M) \times (W+2M) - H \times W}{P^2}$ . The extrapolated image  $\hat{x}$  can

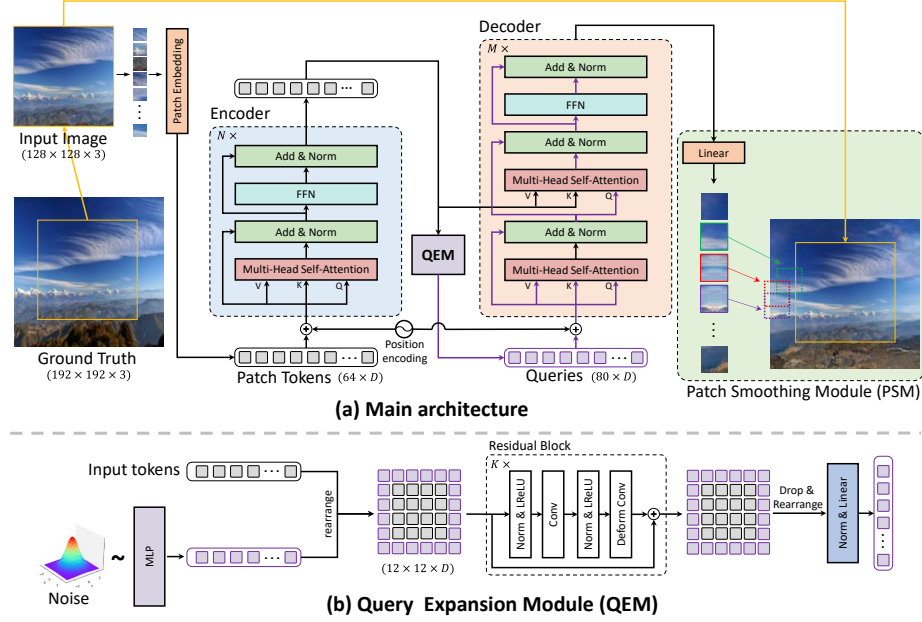


Fig.S3: (a) Main architecture of hybrid transformer generator in QueryOTR consists of transformer encoder and decoder, QEM and PSM. (b) Structure of Query Expansion Module.

be obtained by reshaping the new sequence of patch tokens into image patches, and then rearranging the image patches around the input image, leading to  $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x}, \{x_p^{L+1}, x_p^{L+2}, \dots, x_p^{L+R}\})$ .

### 3.2 Hybrid Transformer Autoencoder

The architecture of the proposed QueryOTR generator is presented in Fig. S3, which is a hybrid transformer autoencoder. The overall architecture is composed of four major components: a transformer encoder extracting patch tokens' representation, a CNN-based Query Expansion Module (QEM) predicting the expanded queries, a transformer decoder processing the expanded queries, and a Patch Smoothing Module (PSM) generating the expanded patches and rearranging them around the original images.

**Transformer Encoder** Our encoder is a standard ViT [9]. Inspired from ViT, the input image is first converted to several non-overlapping patches represented as a sequence of patch tokens  $X_p$ . The encoder module embeds the patch tokens through a linear projection  $\mathbf{E}$  with the added positional embeddings  $\mathbf{E}_{pos}$ . Then the encoder processes the set of patch tokens via a series of Transformer Blocks

with a length of  $N$ . The transformer-based encoder can be described as follows:

$$h_0 = [x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^L \mathbf{E}] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot 3) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{L \times D} \quad (1)$$

$$h'_n = \text{MSA}(\text{LN}(h_{n-1})) + h_{n-1}, \quad n = 1, \dots, N \quad (2)$$

$$h_n = \text{FFN}(\text{LN}(h'_n)) + h'_n, \quad n = 1, \dots, N \quad (3)$$

$$h_{enc} = \text{LN}(h_N), \quad (4)$$

where  $D$  is the hidden dimension of transformer block, FFN is a feed forward network, LN denotes layer normalization,  $h_n$  are the intermediate tokens' representations,  $h_{enc}$  denotes the output patch tokens of the transformer encoder, and MSA represents the multi-headed self-attention.

Given the learnable matrices  $\mathbf{W}_q$ ,  $\mathbf{W}_k$ ,  $\mathbf{W}_v$  corresponding to query, key, and value representations, a single self-attention head (indexed with  $h$ ) is computed:

$$\text{Attention}_h(\mathbf{X}, \mathbf{Y}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{d_h})\mathbf{V}, \quad (5)$$

where  $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$ ,  $\mathbf{K} = \mathbf{Y}\mathbf{W}_k$ ,  $\mathbf{V} = \mathbf{Y}\mathbf{W}_v$ . Multi-headed self-attention aggregates information with linear projection operation on the concatenation of the  $H$  self-attention heads:

$$\text{MSA}(\mathbf{X}) = \text{concat}_{h=1}^H [\text{Attention}_h(\mathbf{X}, \mathbf{X})] \mathbf{W} + \mathbf{b}, \quad (6)$$

where  $\mathbf{W}$  and  $\mathbf{b}$  are learnable matrices for the aggregated features.

**Query Expansion Module** The proposed QEM is designed to speed up the convergence of pure transformer by **generating the expanded queries for the transformer decoder**. We predict the decoders' queries conditioned on encoders' features, and take advantage of CNN's inductive bias to accelerate the convergence. As shown in Fig. S3(b), the input tokens  $h_{enc}$  are first reshaped to the feature map with the size of  $\frac{H}{P} \times \frac{W}{P} \times D$ . Then the reshaped feature maps are extrapolated with extra  $\frac{M}{P}$  pixels along width and height, where the padded tokens are generated by Multi-layer Perceptual (MLP) with uniform input noise. After that, we utilize stacked residual blocks [18] **equipped with deformable convolutional layers** [48] **to process the queries**, which is commonly practiced to capture local and long-term dependencies. Finally, **the expanded queries are extracted and transformed as sequence**, followed by one Normalization Layer and one Linear Layer. This process can be described as:

$$q_{expand} = \text{QEM}(h_{enc}, z), \quad z \sim \mathcal{N}(0, 1). \quad (7)$$

**Transformer Decoder** Inspired from the original transformer [41], **the decoder equips one extra sub-layer which performs the multi-head cross attention (MCA) similar to the encoder with two sub-layers**. Specifically, in MCA the queries come from the previous decoder layer and the keys and values come from the output of the encoder. This allows each position in the decoder to attend over all positions



in the input sequence, leading to significant improvements of the generating performance. The process can be described as follows:

$$q_0 = q_{expand} + \mathbf{E}'_{pos}, \quad \mathbf{E}'_{pos} \in \mathbb{R}^{R \times D} \quad (8)$$

$$q'_m = \text{MSA}(\text{LN}(q_{m-1})) + q_{m-1}, \quad m = 1, \dots, M \quad (9)$$

$$q''_m = \text{MCA}(\text{LN}(q'_m), h_{enc}) + q'_m, \quad m = 1, \dots, M \quad (10)$$

$$q_m = \text{FFN}(\text{LN}(q''_m)) + q''_m, \quad m = 1, \dots, M \quad (11)$$

The multi-headed cross-attention in Eq. 10 aggregates information from  $H$  cross attention heads, as follows:

$$\text{MCA}(\mathbf{X}, \mathbf{Y}) = \text{concat}_{h=1}^H [\text{Attention}_h(\mathbf{X}, \mathbf{Y})] \mathbf{W} + \mathbf{b}. \quad (12)$$

**Patch Smoothing Module** The linear module is prone to generate artifacts if predicting output patches using predefined patch size of  $P \times P$ . The reason is that the output tokens are processed independently without explicit constraints. These arbitrary grid partitions could make the image contents discontinuous across the border edge of each patch. In order to mitigate this issue, we allow some overlaps among image patches. For each border edge of one patch, we extend it by  $o$  pixels generating the output image patch size as  $(P+2o) \times (P+2o)$ . This operation involves the decoder with the neighboring patches' content having a better sense of locality in the transformer architecture, thus enabling the output sequence to have same length but less effect as the predefined grids. PSM can be described as:

$$\hat{x} = \mathcal{S}(x, q_M W_{proj}), \quad W_{proj} \in \mathbb{R}^{D \times ((P+2o)^2 \cdot 3)}, \quad (13)$$

where  $\mathcal{S}$  is a function to place the extrapolated overlapped patches around the input image, and average the pixel values in the overlapped areas.

### 3.3 Loss Functions

Our loss function consists of three parts: a patch-wise reconstruction loss, a perceptual loss, and an adversarial loss. The reconstruction loss is responsible for capturing the overall structure of predicted patches, whilst the perceptual loss and adversarial loss are coupled to maintain good perceptual quality and promote more realistic prediction.

**Patch-wise Reconstruction Loss** We utilize an L2 distance between the sequence of ground truth image patches  $\{y_p^{L+1}, y_p^{L+2}, \dots, y_p^{L+R}\}$  and the sequence of predicted image patches  $\{x_p^{L+1}, x_p^{L+2}, \dots, x_p^{L+R}\}$ :

$$\mathcal{L}_{rec} = \frac{1}{R} \sum_{i=L+1}^{L+R} \|y_p^i - x_p^i\|_2^2, \quad (14)$$

where the patch size is  $(P+2o) \times (P+2o)$ . We engage a per-patch normalization to enhance the patch contrast locally, where the mean and std of the image patches are pre-computed.



**Perceptual Loss** Perceptual loss provides a supervision on the intermediate features that can help retain more semantic information. Following previous work [10,21,24], we extract the features from a VGG-19 [38] network pretrained on ImageNet [7], which is denoted as  $\phi$ . The perceptual loss is devised as follows:

$$\mathcal{L}_{perceptual} = \frac{1}{5} \sum_{j=1}^5 (w^j \times (\phi^j(\hat{x}) - \phi^j(y))), \quad (15)$$

where the superscript  $j$  is the index of feature map scales from  $\phi$ , and  $w^j$  is set to 1/32, 1/16, 1/8, 1/4, 1 as the scale decreases.

**Adversarial Loss** We utilize the same multi-scale PatchGAN discriminator  $D$  used in pix2pixHD [42] except that we replace the least squared loss term [32] with the hinge loss term [27]. Since the PatchGAN discriminator has a fixed receptive field of patch, we take the whole generated images instead of image patches to train the GAN. The extrapolated images generated by our QueryOTR should be indistinguishable from real images by the discriminator. Given the extrapolated images  $\hat{x} \sim \mathbb{P}_g$  generated by QueryOTR and real images  $y \sim \mathbb{P}_y$ , the adversarial loss for the discriminator is

$$\mathcal{L}_{adv}^D = \min_D \mathbb{E}_{\hat{x} \sim \mathbb{P}_g} (\min(1 + D(\hat{x}))) + \mathbb{E}_{y \sim \mathbb{P}_y} (\min(1 - D(y))). \quad (16)$$

Additionally, the adversarial loss for the generator is

$$\mathcal{L}_{adv}^G = \min_G -\mathbb{E}_{\hat{x} \sim \mathbb{P}_g} D(\hat{x}). \quad (17)$$

We jointly train the hybrid transformer generator and CNN discriminators and optimize the final objective as a weighted sum of the above mentioned loss terms:

$$\min_G \max_D \mathcal{L}_{adv} + \lambda_{rec} \mathcal{L}_{rec} + \lambda_{perceptual} \mathcal{L}_{perceptual}, \quad (18)$$

where  $\lambda_{rec}$ ,  $\lambda_{perceptual}$  are weights controlling the importance of loss terms. In our experiments, we set  $\lambda_{rec} = 5$ , and  $\lambda_{perceptual} = 10$ .

## 4 Experiments

### 4.1 Datasets, Implementation and Training Details

We use three datasets with {Scenery [44], Building Facades [11], and WikiArt [39]} for the experiments. Details about the three datasets could be found in the supplementary materials.

We implement our framework with PyTorch [34] equipped with a NVIDIA GeForce RTX 3090 GPU 1.9.0. Hybrid transformer generator contains 12 stacked transformer encoder layers and 4 stacked transformer decoder layers. We initialise the weights of generator encoder by utilizing the pre-trained ViT [17]. Adam [23] is used as the optimizer to minimize the objective function with the mini-batch of 64,  $\beta_1 = 0.0$ ,  $\beta_2 = 0.99$ , and weight decay of 0.0001. The  $\alpha$  is

	Methods	Scenery			Building Facades			WikiArt		
		FID↓	IS↑	PSNR↑	FID↓	IS↑	PSNR↑	FID↓	IS↑	PSNR↑
1×	SRN	47.781	2.981	22.440	38.644	3.862	18.588	76.749	3.629	<u>20.072</u>
	NSIPO	25.977	3.059	21.089	30.465	4.153	18.314	22.242	5.600	18.592
	IOH	32.107	2.886	22.286	49.481	3.924	18.431	40.184	4.835	19.403
	Uformer	<u>20.575</u>	<u>3.249</u>	<u>23.007</u>	<u>30.542</u>	<u>4.189</u>	<u>18.828</u>	<u>15.904</u>	<u>6.567</u>	19.610
	QueryOTR	<b>20.366</b>	<b>3.955</b>	<b>23.604</b>	<b>22.378</b>	<b>4.978</b>	<b>19.680</b>	<b>14.955</b>	<b>7.896</b>	<b>20.388</b>
2×	SRN	83.772	2.349	18.403	74.304	3.651	15.355	137.997	3.039	<u>16.646</u>
	NSIPO	45.989	2.606	17.733	<u>58.341</u>	3.669	15.262	51.668	4.591	15.679
	IOH	44.742	2.655	18.739	76.476	3.456	15.443	75.070	4.289	16.056
	Uformer	<u>39.801</u>	<u>2.920</u>	<u>18.920</u>	63.915	<u>3.798</u>	<u>15.612</u>	<b>41.107</b>	<u>5.900</u>	15.947
	QueryOTR	<b>39.237</b>	<b>3.431</b>	<b>19.358</b>	<b>41.273</b>	<b>4.547</b>	<b>16.213</b>	<u>43.757</u>	<b>6.341</b>	<b>17.074</b>
3×	SRN	115.193	2.087	16.123	110.036	2.938	13.693	181.533	2.504	<u>14.609</u>
	NSIPO	64.457	2.405	15.606	<u>81.301</u>	<u>3.431</u>	13.791	75.785	4.225	14.257
	IOH	<b>58.629</b>	2.432	16.307	95.068	2.790	13.894	108.328	3.728	13.919
	Uformer	<u>60.497</u>	<u>2.638</u>	<u>16.379</u>	93.888	3.388	<u>14.051</u>	<u>72.923</u>	<b>5.904</b>	13.464
	QueryOTR	60.977	<b>3.114</b>	<b>16.864</b>	<b>64.926</b>	<b>4.612</b>	<b>14.316</b>	<b>69.951</b>	<u>5.683</u>	<b>15.294</b>

Table S1: Quantitative results of one-step and multi-step outpainting. Best and second best results are **boldface** and underlined. 1× represents one step outpainting, while 2× and 3× denote two- and three-step outpainting respectively.

set to 8 considering the complexity and precision. Our QueryOTR is trained for 300, 200 and 120 epochs on Scenery, Building Facades, and WikiArt datasets respectively with the learning rate of 0.0001. The warm-up trick [18] is utilized in the first 10 epochs with the reconstruction loss only. For discriminator regularization, DiffAug [46] and spectral normalization [33] are used to stabilise the training dynamics.

We conduct generalised image outpainting for experimental comparison following the previous work. In the training stage, the original images are resized to the size  $192 \times 192$  as the ground truth images. Then the input images with the size  $128 \times 128$  are obtained by the center cropping operation. In the testing stage, all images are resized to  $192 \times 192$  as the ground truth, and then the input images are obtained by center cropping to the sizes  $128 \times 128$ ,  $86 \times 86$ , and  $56 \times 56$  for 1×, 2×, and 3× outpainting respectively. Excepted for horizontal flip and image normalization, no other data augmentation is used for ease of setup. The total output sizes are 2.25, 5, and 11.7 times of the input in terms of 1×, 2×, and 3× outpainting, indicating that over half of all pixels will be generated.

## 4.2 Experimental Results

We make comparisons with three SOTA CNN-based image outpainting methods, NSIPO [44], SRN [43], and IOH [40], and one transformer-based method Uformer [11] to demonstrate the effectiveness of QueryOTR. For all the experiments, we set the input and output sizes as  $128 \times 128$  and  $192 \times 192$ .

We use Inception Score (IS) [37], Fréchet Inception Distance (FID) [20], and peak signal-to-noise ratio (PSNR) to measure the generative quality objectively. The upper-bounds of IS are 4.091, 5.660 and 8.779 for Scenery, Building Facades and WikiArt, respectively, which are calculated by real images in test set.

**Quantitative Result** Tab. S1 shows quantitative results. Our QueryOTR outperforms the competition on almost all metrics on 1-step and multi-step outpainting. In particular, QueryOTR shows obvious superiority in all entries compared with CNN-based methods, e.g., SRN, NSIPO, and IOH. These results show that transformer structure succeeds in capturing global dependencies for image outpainting compared with CNN’s inductive biases. Meanwhile, our QueryOTR outperforms the very competitive Swin-based Uformer which uses an image-to-image translation approach for image extrapolation, mainly because our query-based method allows to generate image patches attended to all the visual locations, yielding a better perceptual consistency. It is noted that our results for  $1\times$  outpainting are very close to the IS upper-bound for all the datasets, indicating realistic image generation and good perceptual consistency. Extra results of replacing the center region with input sub-images are in the supplementary.

**Qualitative Result** Some examples of visual results on all the datasets are shown in Fig. S4. Our QueryOTR effectively extrapolates the images by querying the global semantic-similar image patches. Seen from the  $1\times$  outpainting results, our QueryOTR could generate more realistic images with vivid details and enrich the contents of the generated regions marked in white box. Furthermore, our method could weaken the sense of edges between the generated regions and input sub-image. Compared with other baselines, our QueryOTR could generate water containing more realistic ripples in the  $3^{rd}$  row and intact trees in the  $5^{th}$  row of Fig. S4, which could be seen in the white dotted box. In the  $7^{th}$  row of Fig. S4, the whole skyscraper generated by QueryOTR indicates the success of our query-based method which predicts the detailed contents with global information by queries. In the  $9^{th}$  row, our method could capture the global information of the green background on the corner marked in the white box. More visual results could be seen in the supplementary material.

### 4.3 Ablation Study

We ablate several critical factors in QueryOTR by progressively adjusting each factor here. It can be seen that each factor contributes to the final success of QueryOTR. We conducted all the ablation experiments on the Scenery dataset.

**Transformer Encoder and Decoder** We compare the impact of the pre-trained ViT-based encoder and the number of transformer decoder layers  $M$ . As shown in Tab. S2(a), utilizing a pretrained ViT encoder contributes to the improvements of FID and IS by 2.418 and 0.204, respectively. The main reason is that the small datasets might not be sufficient to train the model for performance saturation. The pretrained ViT encoder is capable of capturing the long-term dependencies, which may benefit the patch prediction. Additionally, our QueryOTR performs optimally in both FID and IS when the number of

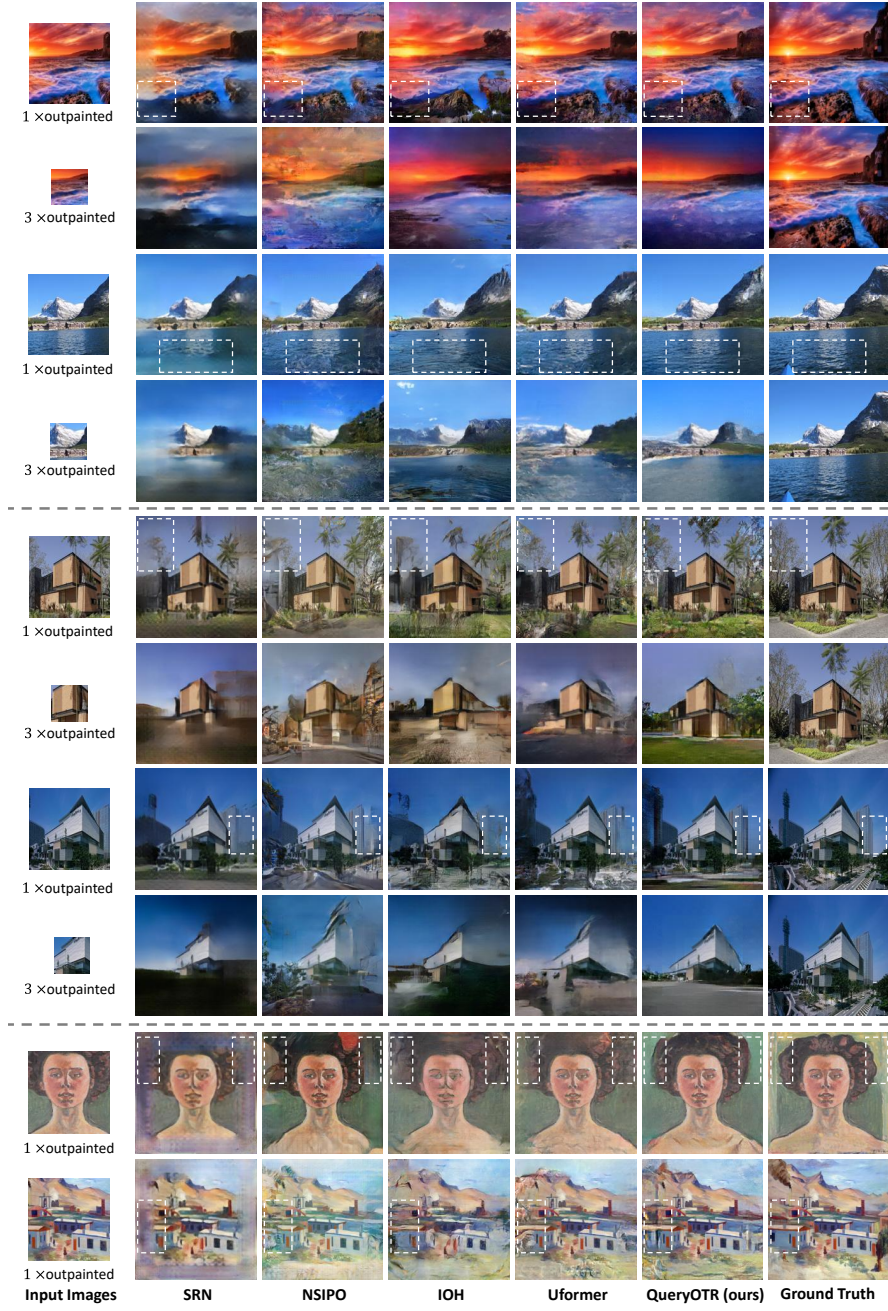


Fig. S4: Comparisons on 1-step and multi-step outpainting with the state-of-the-art methods. Our QueryOTR achieves the best image quality.

Pretrained <i>Enc.</i>	<i>M</i>	FID↓	IS↑
-	4	22.784	3.751
✓	2	20.731	3.931
✓	4	<b>20.366</b>	<b>3.955</b>
✓	8	20.373	3.852

(a) Ablation of the pretrained ViT-base encoder and the number of transformer decoder layers  $M$ .

	FID↓	IS↑
w/o QEM	36.967	3.642
QEM w/o Noise	23.444	3.728
QEM w/o DC [48]	23.530	<b>3.775</b>
w QEM	<b>22.784</b>	3.751

(c) Impact of proposed Query Expansion Module (QEM) and its key internal components.

	FID↓	IS↑
w/o $\mathcal{L}_{rec}$ & $\mathcal{L}_{perceptual}$	38.009	3.433
w/o $\mathcal{L}_{rec}$	31.282	3.744
w/o $\mathcal{L}_{perceptual}$	33.380	3.510
QueryOTR (baseline)	<b>20.366</b>	<b>3.955</b>

(b) Impact of  $\mathcal{L}_{rec}$  and  $\mathcal{L}_{perceptual}$  contribute to the overall performance. The model is default trained with three losses.

PSM	Per-Patch <i>Norm.</i>	FID↓	IS↑
-	-	51.945	3.801
-	✓	31.073	3.753
✓	-	22.501	3.707
✓	✓	<b>20.366</b>	<b>3.955</b>

(d) Effect of the proposed Patch Smoothing Module (PSM) and per-patch image normalization.

Table S2: Ablation studies validated on Scenery dataset.

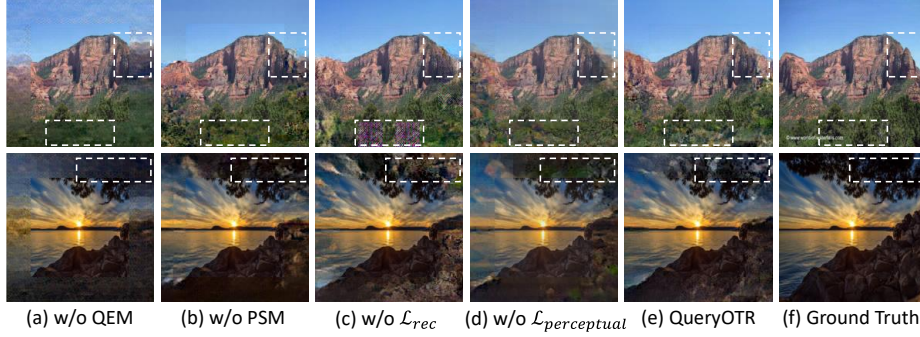


Fig. S5: Visualisation of ablation study.

decoder layers is set to 4. Further increasing the depth of decoder indefinitely will not improve the performance of our QueryOTR.

**Loss Terms** We investigate the impact of patch-wise reconstruction loss  $\mathcal{L}_{rec}$  and perceptual loss  $\mathcal{L}_{perceptual}$  in Tab. S2(b). We first train the model with only adversarial loss, which is equivalent to training the model unpaired, resulting in a FID of 38.009 and IS of 3.433. On the basis of adversarial training, using either  $\mathcal{L}_{rec}$  or  $\mathcal{L}_{perceptual}$  could improve the overall performance. Fig. S5(c) and (d) show that high-frequency checkerboard artifacts occur when trained without  $\mathcal{L}_{rec}$ , and the details cannot be generated without  $\mathcal{L}_{perceptual}$ .

**QEM** We ablate the impact of QEM and its internal key components. In the experiments, we do not use a pretrained encoder to avoid reducing the difficulty of training learnable queries. Since training pure transformer may require larger datasets and longer time, it is hard for learnable queries to converge



well on Scenery dataset, resulting in a high FID (see Tab. S2(c)) and blurry image patches (see Fig. S5(a)). On the other hand, the proposed QEM generates queries conditioned on input images, significantly improving FID by 14.227. Meanwhile, generating queries with noise slightly improves the patch diversity, and deformable convolution enables an active long distance modeling for query generation.

To further investigate how QEM affect the convergence speed of pure transformer, we train the pure transformer with and without QEM module for 1000 epochs. As shown in Fig. S2(a), the convergence rate of the pure transformer with QEM is about 3.3 times faster than that without QEM on a relatively small dataset indicating the superiority of QEM in accelerating the model convergence. On the other hand, the loss declines slowly without QEM, which might be caused by the insufficient training data. The reason leading to this phenomenon is that the pure transformer will process almost 4 billion possibilities if the  $16 \times 16$  pixel patch is treated as a word, which needs larger semantic space for attention processing. When dealing with a small dataset, the amount of data is not enough to regress the extrapolated patches resulting in model degradation.

**PSM** Tab. S2(d) demonstrates the effect of the proposed PSM and per-patch normalization. Although using a single linear layer can generate vivid image patches, the connections between patches are unnatural, as shown in Fig. S5(b). Per-patch normalization could improve the reconstruction of high-frequency by enhancing the local contrast of patches, leading an improvement of FID 20.872. Meanwhile, PSM significantly alleviates the checkerboard artifacts caused by per-patch prediction, and improves the overall perceptual quality of the extrapolated images. PSM alleviates checkerboard artifacts via explicit constraints, while perceptual loss penalizes image discontinuity from a semantic perspective. PSM appears more effective and direct than perceptual loss. If both are applied, even better performance can be obtained.

## 5 Conclusion

In this paper, we have proposed a novel hybrid query-based encoder-decoder transformer framework, **QueryOTR**, to extrapolate visual context all-side around a given image. The transformer structure breaks through the limitation of capturing image long-rang dependencies and intrinsic locality. The special designed module QEM helps to accelerate the transformer model convergence on small datasets and PSM contributes to generate seamless extrapolated images realistically and smoothly. Extensive experiments on Scenery, Building and WikiArt datasets proved the superiority of our query-based method.

**Acknowledgments.** The work was partially supported by the following: National Natural Science Foundation of China under no.61876155; Jiangsu Science and Technology Programme under no.BE2020006-4; Key Program Special Fund in XJTU under no.KSF-T-06, no.KSF-E-26 and no.KSF-E-37; Research Development Fund in XJTU under no.RDF-19-01-21.

## References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 6836–6846 (2021)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics* **28**(3), 24 (2009)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. pp. 417–424 (2000)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: *International Conference on Learning Representations* (2019)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: *European conference on computer vision*. pp. 213–229. Springer (2020)
6. D’Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: *International Conference on Machine Learning*. pp. 2286–2296. PMLR (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. pp. 248–255. Ieee (2009)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations* (2021)
10. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems* **29** (2016)
11. Gao, P., Yang, X., Zhang, R., Huang, K., Geng, Y.: Generalised image outpainting with u-transformer. *arXiv preprint arXiv:2201.11403* (2022)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems* **27** (2014)
13. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet’s clothing for faster inference. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12259–12269 (2021)
14. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 3012–3021 (2020)
15. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. *Advances in neural information processing systems* **30** (2017)



16. Guo, D., Liu, H., Zhao, H., Cheng, Y., Song, Q., Gu, Z., Zheng, H., Zheng, B.: Spiral generative network for image extrapolation. In: European Conference on Computer Vision. pp. 701–717. Springer (2020)
17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11936–11945 (2021)
20. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017)
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
22. Kim, K., Yun, Y., Kang, K.W., Kong, K., Lee, S., Kang, S.J.: Painting outside as inside: Edge guided image outpainting via bidirectional rearrangement with progressive step learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2122–2130 (2021)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning. pp. 1558–1566. PMLR (2016)
25. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., 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. pp. 4681–4690 (2017)
26. Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., Liu, C.: Vitgan: Training gans with vision transformers. arXiv preprint arXiv:2107.04589 (2021)
27. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
28. Lin, H., Pagnucco, M., Song, Y.: Edge guided progressively generative image outpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 806–815 (2021)
29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
30. Lu, C.N., Chang, Y.C., Chiu, W.C.: Bridging the visual gap: Wide-range image blending. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 843–851 (2021)
31. Ma, Y., Ma, J., Zhou, M., Chen, Q., Ge, T., Jiang, Y., Lin, T.: Boosting image outpainting with semantic layout prediction. arXiv preprint arXiv:2110.09267 (2021)
32. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)

33. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
35. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
36. Sabini, M., Rusak, G.: Painting outside the box: Image outpainting with gans. arXiv preprint arXiv:1808.08483 (2018)
37. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. *Advances in neural information processing systems* **29** (2016)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
39. Tan, W.R., Chan, C.S., Aguirre, H.E., Tanaka, K.: Ceci n’est pas une pipe: A deep convolutional network for fine-art paintings classification. In: 2016 IEEE international conference on image processing. pp. 3703–3707. IEEE (2016)
40. Van Hoorick, B.: Image outpainting and harmonization using generative adversarial networks. arXiv preprint arXiv:1912.10960 (2019)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
42. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8798–8807 (2018)
43. Wang, Y., Tao, X., Shen, X., Jia, J.: Wide-context semantic image extrapolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1399–1408 (2019)
44. Yang, Z., Dong, J., Liu, P., Yang, Y., Yan, S.: Very long natural scenery image prediction by outpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10561–10570 (2019)
45. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
46. Zhao, S., Liu, Z., Lin, J., Zhu, J.Y., Han, S.: Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems* **33**, 7559–7570 (2020)
47. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886 (2021)
48. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9308–9316 (2019)

## A Additional Quantitative Results

	Methods	Scenery			Building Facades			WikiArt		
		FID↓	IS↑	PSNR↑	FID↓	IS↑	PSNR↑	FID↓	IS↑	PSNR↑
1×	Lower Bound	160.174	3.595	9.569	123.678	4.356	9.810	139.956	5.073	10.215
	SRN	45.296	3.540	22.433	34.058	4.722	18.839	65.675	4.933	<b>20.467</b>
	NSIPO	35.606	3.475	21.630	33.140	4.529	18.460	30.338	6.231	18.929
	IOH	23.410	3.578	22.839	33.525	<u>4.739</u>	18.812	24.539	6.679	19.808
	Uformer	<u>23.216</u>	<u>3.691</u>	<u>23.054</u>	32.228	4.651	<u>18.892</u>	<u>18.808</u>	7.466	19.708
	QueryOTR	<b>20.366</b>	<b>3.955</b>	<b>23.604</b>	<b>22.378</b>	<b>4.978</b>	<b>19.680</b>	<b>14.955</b>	<b>7.896</b>	<u>20.388</u>
2×	Lower Bound	201.871	2.097	7.868	196.650	2.875	8.047	230.893	2.477	8.557
	SRN	97.989	2.724	18.459	75.121	3.837	15.431	139.395	3.045	<u>16.759</u>
	NSIPO	69.683	<u>3.235</u>	17.701	<u>65.319</u>	3.771	15.287	67.880	4.888	15.721
	IOH	<u>45.108</u>	3.047	18.846	<u>72.053</u>	3.727	15.519	66.953	5.065	16.127
	Uformer	50.605	3.099	<u>18.934</u>	71.306	<u>3.924</u>	<u>15.626</u>	<u>51.263</u>	6.098	15.936
	QueryOTR	<b>39.237</b>	<b>3.431</b>	<b>19.358</b>	<b>41.273</b>	<b>4.547</b>	<b>16.213</b>	<b>43.757</b>	<b>6.341</b>	<b>17.074</b>
3×	Lower Bound	227.268	1.991	7.242	223.224	2.378	7.384	260.623	2.258	7.919
	SRN	141.040	2.483	16.141	114.016	3.312	13.777	181.394	2.407	<u>14.620</u>
	NSIPO	101.411	<b>3.131</b>	15.384	<u>92.041</u>	<u>3.628</u>	13.741	94.176	4.325	14.159
	IOH	<u>67.591</u>	2.723	16.351	104.337	2.956	13.913	104.032	4.190	13.943
	Uformer	76.318	2.799	<u>16.374</u>	105.539	3.315	<u>14.065</u>	<u>79.322</u>	<b>5.954</b>	13.411
	QueryOTR	<b>60.977</b>	<u>3.114</u>	<b>16.864</b>	<b>64.926</b>	<b>4.612</b>	<b>14.316</b>	<b>69.951</b>	5.683	<b>15.294</b>
	Up Bound	0	4.184	+∞	0	5.660	+∞	0	8.779	+∞

Table S3: Quantitative results follow our setting that **replacing the center region with input sub-images surrounded by the extrapolated parts.**

The comparative methods are all based on image-to-image translation, which need to reconstruct the input sub-image, whilst the sequence-to-sequence based method QueryOTR does not need to reconstruct the input sub-image only outputting the extrapolated regions. In the main manuscript, we report the best results of comparative methods by keeping the reconstructed regions, which were consistent with their original settings. All things being equal, [Tab. S3](#) reports the results following our setting that replaces the center region with the input sub-image. We additionally report the lower bound of each metric by filling the extrapolated regions with zero pixel values. As shown in [Tab. S3](#), our proposed QueryOTR outperforms other methods in most cases, indicating that the higher performance of our method contributes little on the use of the input sub-image. Instead, the results demonstrate the superiority of our method on generating the extrapolated regions. On the other hand, all the comparative methods have an improvement of IS and PSNR metrics due to replacing the center regions with the input sub-image.

## B Details of Datasets

**Scenery** is a natural scenery dataset consisting of about 5,000 images in the training set and 1,000 images in the testing set. The images are very diverse and complicated, which contains natural scenes, e.g., snow, valley, seaside, riverbank, sky, and mountain.

**Building Facades** is a city scenes dataset consisting of about 16,000 and 1,500 images for training set and testing set respectively. It contains building architecture and city scenes.

**WikiArt** is a fine-art paintings dataset obtained from the wikiart.org website. We use the split manner of genres datasets, which contains 45,503 training images and 19,492 testing images.

## C Inference Time

The comparison of inference time can be referred in [Tab. S4](#). Due to the simple but effective design of QueryOTR, our framework is almost three times faster than Uformer which is also engaged with a vision transformer (Swin Transformer) architecture.

Method	SRN	NSIPO	IOH	Uformer	QueryOTR
Time usage (ms/image)	11.960	44.190	4.160	46.810	13.345

Table S4: Comparison on inference time.

## D Hard Examples

We illustrate some hard examples that QueryOTR can work significantly better than the other methods. As shown in [Fig. S6](#), extrapolating the images with

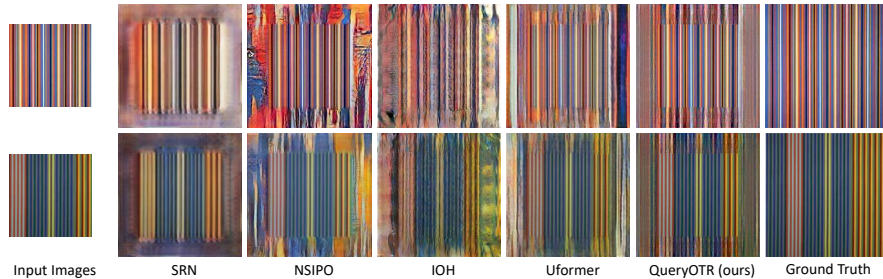


Fig. S6: Visualization of some hard examples in the test set of WikiArt dataset.

simple colour stripes is very challenging, which requires the network to recognize the pattern and mimic it, especially when such samples are not enough in the training set. The CNN-based methods have limitations to generalize well on such samples, whilst the transformer-based Uformer can generate colorful lines but not straight. In contrast, QueryOTR takes advantage of querying the input sub-image to generate color and straight lines, generating much better images.

## E More Qualitative Results

We present more comparative results for one-step and multi-step outpainting. In [Fig. S7](#), we show additional results on Scenery and WikiArt datasets. Similarly, in [Fig. S8](#), we provide more results on Building Facades dataset compared with other methods. Meanwhile, we visualize the results conducted by QueryOTR in [Fig. S9](#).

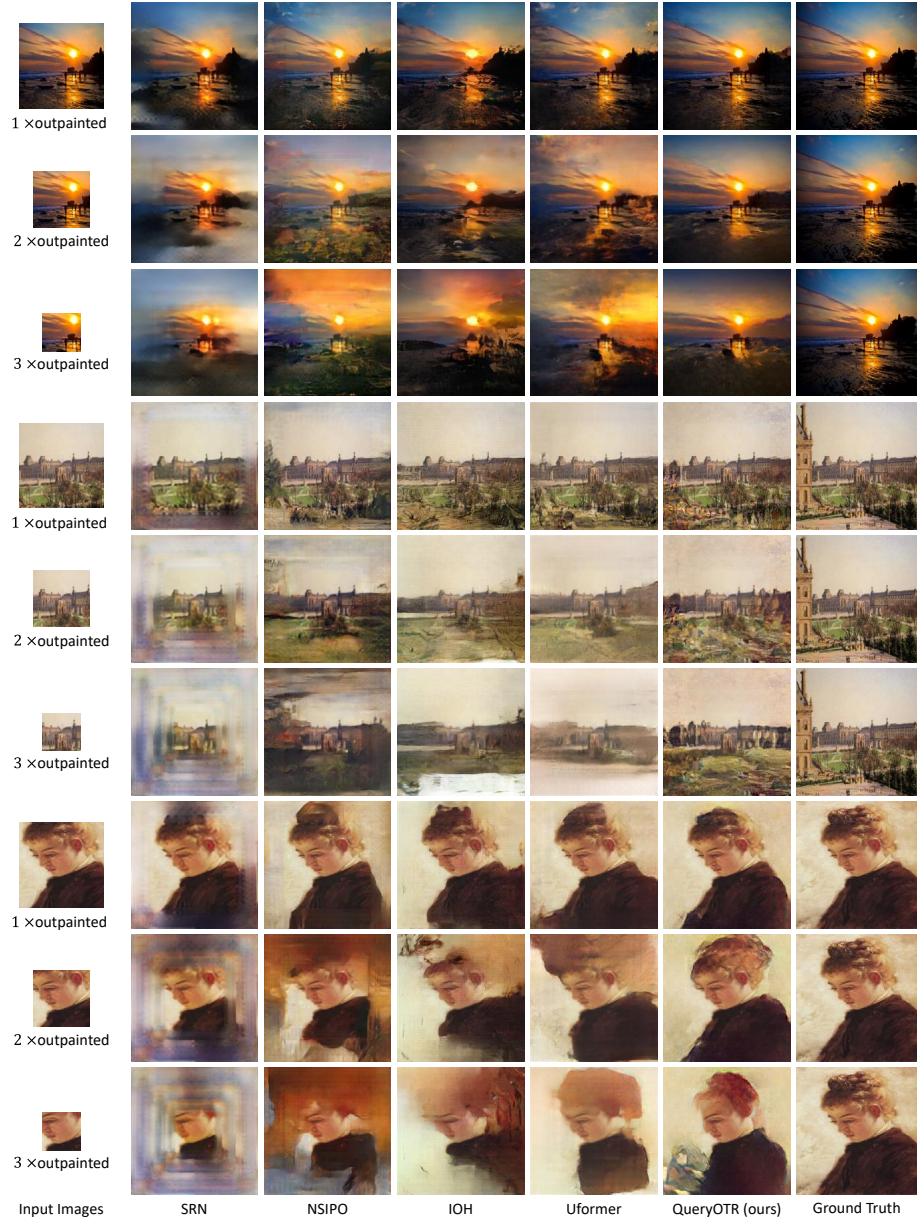


Fig. S7: Qualitative comparison results on Scenery and WikiArt datasets.



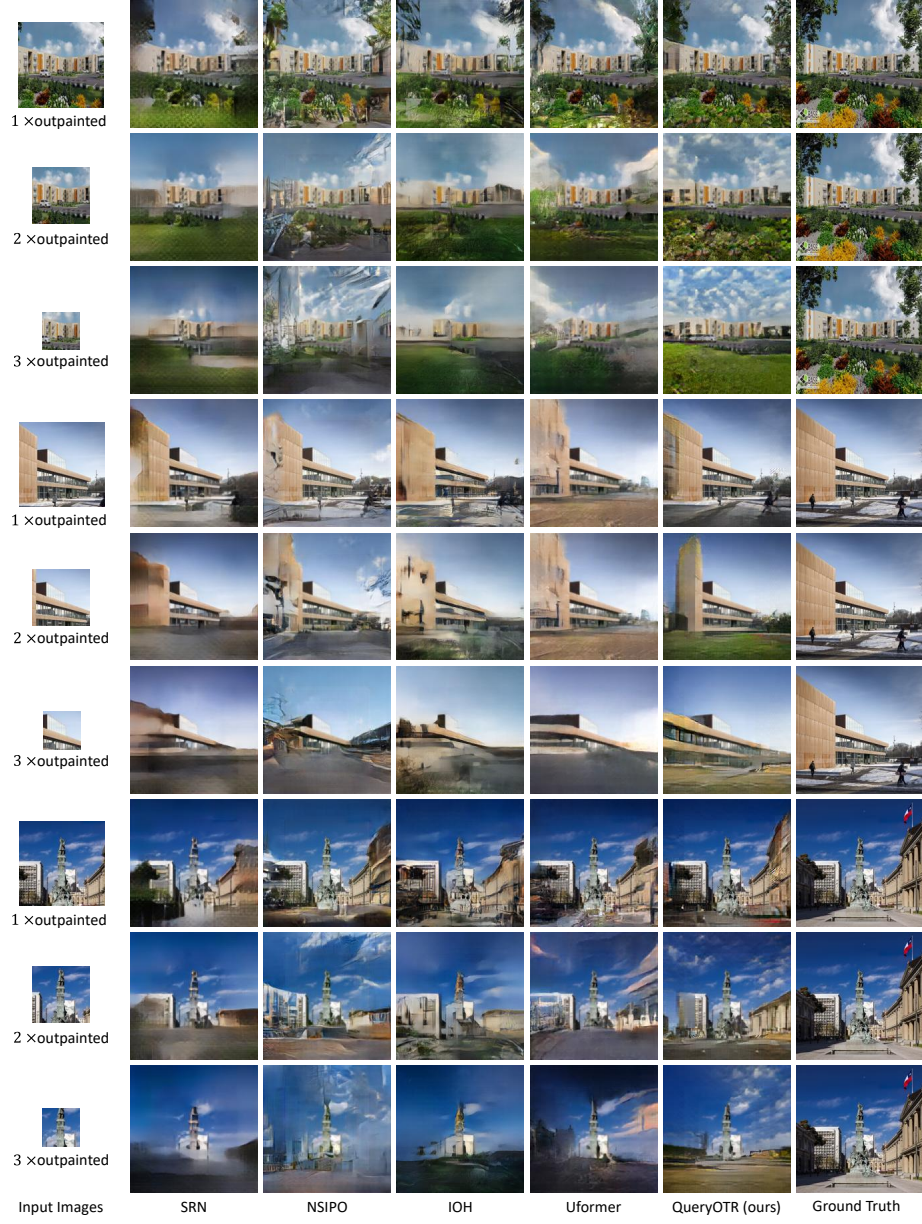


Fig. S8: Qualitative comparison results on Building Facades dataset.



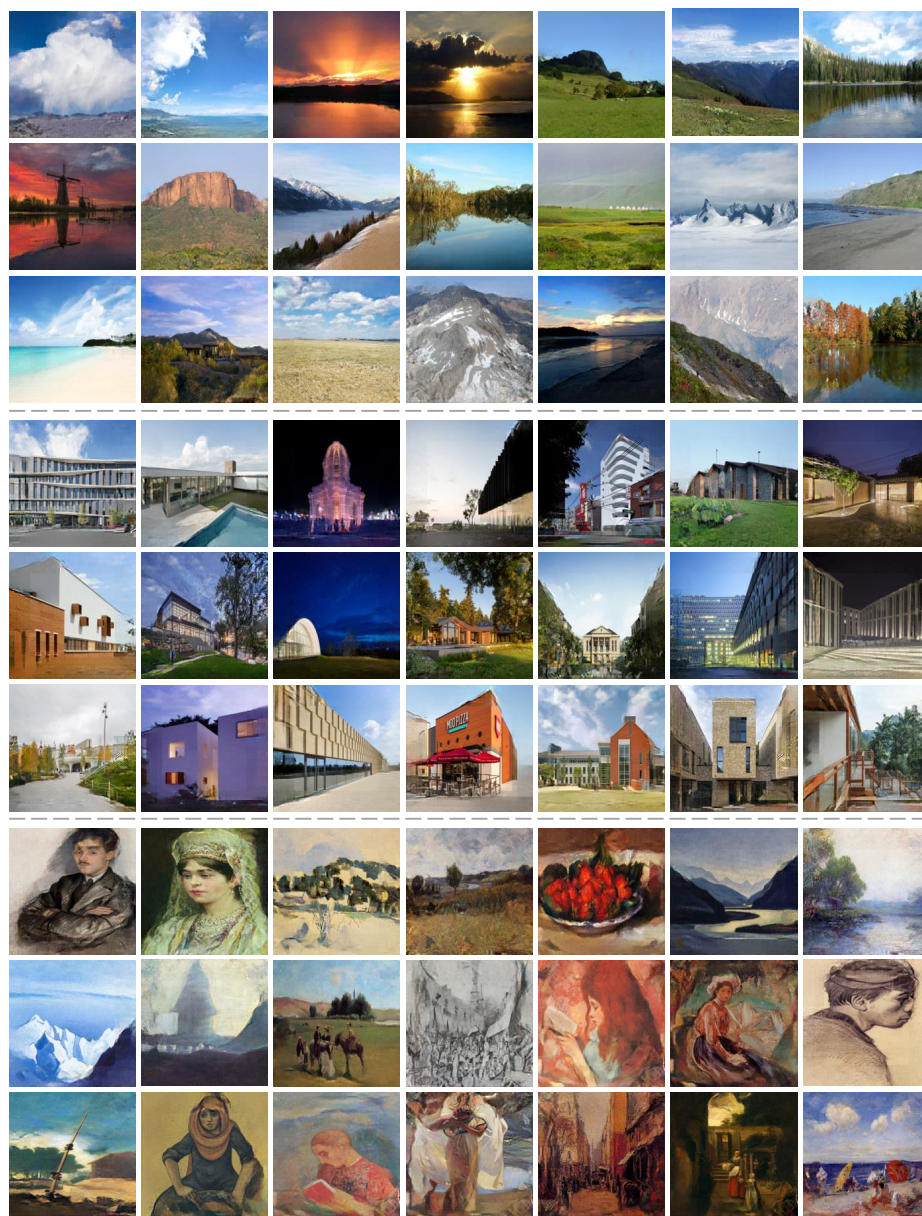


Fig. S9: Visualization of QueryOTR one-step outpainting.