Taming Transformers for High-Resolution Image Synthesis

Patrick Esser* Robin Rombach* Björn Ommer
Heidelberg Collaboratory for Image Processing, IWR, Heidelberg University, Germany
*Both authors contributed equally to this work

Abstract

Designed to learn long-range interactions on sequential data, transformers continue to show state-of-the-art results on a wide variety of tasks. In contrast to CNNs, they contain no inductive bias that prioritizes local interactions. This makes them expressive, but also computationally infeasible for long sequences, such as high-resolution images. We demonstrate how combining the effectiveness of the inductive bias of CNNs with the expressivity of transformers enables them to model and thereby synthesize high-resolution images. We show how to (i) use CNNs to learn a context-rich vocabulary of image constituents, and in turn (ii) utilize transformers to efficiently model their composition within high-resolution images. Our approach is readily applied to conditional synthesis tasks, where both non-spatial information, such as object classes, and spatial information, such as segmentations, can control the generated image. In particular, we present the first results on semantically-guided synthesis of megapixel images with transformers. Project page at https://git.io/JLlvY.

1. Introduction

Transformers are on the rise—they are now the de-facto standard architecture for language tasks [64, 50, 51, 5] and are increasingly adapted in other areas such as audio [12] and vision [8, 15]. In contrast to the predominant vision architecture, convolutional neural networks (CNNs), the transformer architecture contains no built-in inductive prior on the locality of interactions and is therefore free to learn complex relationships among its inputs. However, this generality also implies that it has to learn all relationships, whereas CNNs have been designed to exploit prior knowledge about strong local correlations within images. Thus, the increased expressivity of transformers comes with quadratically increasing computational costs, because all pairwise interactions are taken into account. The resulting energy and time requirements of state-of-the-art transformer models thus pose fundamental problems for scaling them to high-resolution images with millions of pixels.

Observations that transformers tend to learn convolutional structures [15] thus beg the question: Do we have to re-learn everything we know about the local structure and regularity of images from scratch each time we train a vision model, or can we efficiently encode inductive image biases while still retaining the flexibility of transformers? We hypothesize that low-level image structure is well described by a local connectivity, i.e. a convolutional architecture, whereas this structural assumption ceases to be effective on higher semantic levels. Moreover, CNNs not only exhibit a strong locality bias, but also a bias towards...
spatial invariance through the use of shared weights across all positions. This makes them ineffective if a more holistic understanding of the input is required.

Our key insight to obtain an effective and expressive model is that, taken together, convolutional and transformer architectures can model the compositional nature of our visual world [44]. We use a convolutional approach to efficiently learn a codebook of context-rich visual parts and, subsequently, learn a model of their global compositions. The long-range interactions within these compositions require an expressive transformer architecture to model distributions over their consituent visual parts. Furthermore, we utilize an adversarial approach to ensure that the dictionary of local parts captures perceptually important local structure to alleviate the need for modeling low-level statistics with the transformer architecture. Allowing transformers to concentrate on their unique strength — modeling long-range relations — enables them to generate high-resolution images as in Fig. 1, a feat which previously has been out of reach. Our formulation directly gives control over the generated images by means of conditioning information regarding desired object classes or spatial layouts. Finally, experiments demonstrate that our approach retains the advantages of transformers by outperforming previous codebook-based state-of-the-art approaches based on convolutional architectures.

2. Related Work

The Transformer Family The defining characteristic of the transformer architecture [64] is that it models interactions between its inputs solely through attention [2, 32, 45] which enables them to faithfully handle interactions between inputs regardless of their relative position to one another. Originally applied to language tasks, inputs to the transformer were given by tokens, but other signals, such as those obtained from audio [37] or images [8], can be used. Each layer of the transformer then consists of an attention mechanism, which allows for interaction between inputs at different positions, followed by a position-wise fully connected network, which is applied to all positions independently. More specifically, the (self-)attention mechanism can be described by mapping an intermediate representation with three position-wise linear layers into three representations, query \( Q \in \mathbb{R}^{N \times d_k} \), key \( K \in \mathbb{R}^{N \times d_k} \) and value \( V \in \mathbb{R}^{N \times d_v} \), to compute the output as

\[
\text{Attn}(Q, K, V) = \text{softmax} \left( \frac{QK^t}{\sqrt{d_k}} \right)V \in \mathbb{R}^{N \times d_v}. \tag{1}
\]

When performing autoregressive maximum-likelihood learning, non-causal entries of \( QK^t \), i.e., all entries below its diagonal, are set to \(-\infty\) and the final output of the transformer is given after a linear, point-wise transformation to predict logits of the next sequence element. Since the attention mechanism relies on the computation of inner products between all pairs of elements in the sequence, its computational complexity increases quadratically with the sequence length. While the ability to consider interactions between all elements is the reason transformers efficiently learn long-range interactions, it is also the reason transformers quickly become infeasible, especially on images, where the sequence length itself scales quadratically with the resolution. Different approaches have been proposed to reduce the computational requirements to make transformers feasible for longer sequences. [48] and [66] restrict the receptive fields of the attention modules, which reduces the expressivity and, especially for high-resolution images, introduces unjustified assumptions on the independence of pixels. [12] and [24] retain the full receptive field but can reduce costs for a sequence of length \( n \) only from \( n^2 \) to \( n\sqrt{n} \), which makes resolutions beyond 64 pixels still prohibitively expensive.

Convolutional Approaches The two-dimensional structure of images suggests that local interactions are particularly important. CNNs exploit this structure by restricting interactions between input variables to a local neighborhood defined by the kernel size of the convolutional kernel. Applying a kernel thus results in costs that scale linearly with the overall sequence length (the number of pixels in the case of images) and quadratically in the kernel size, which, in modern CNN architectures, is often fixed to a small constant such as \( 3 \times 3 \). This inductive bias towards local interactions thus leads to efficient computations, but the wide range of specialized layers which are introduced into CNNs to handle different synthesis tasks [46, 70, 59, 74, 73] suggest that this bias is often too restrictive.

Convolutional architectures have been used for autoregressive modeling of images [61, 62, 10] but, for low-resolution images, previous works [48, 12, 24] demonstrated that transformers consistently outperform their convolutional counterparts. Our approach allows us to efficiently model high-resolution images with transformers while retaining their advantages over state-of-the-art convolutional approaches.

Two-Stage Approaches Closest to ours are two-stage approaches which first learn an encoding of data and afterwards learn, in a second stage, a probabilistic model of this encoding. [13] demonstrated both theoretical and empirical evidence on the advantages of first learning a data representation with a Variational Autoencoder (VAE) [34, 54], and then again learning its distribution with a VAE. [17, 68] demonstrate similar gains when using an unconditional normalizing flow for the second stage, and [55, 56] when using a conditional normalizing flow. To improve training efficiency of Generative Adversarial Networks (GANs), [39]
Figure 2. Our approach uses a convolutional VQGAN to learn a codebook of context-rich visual parts, whose composition is subsequently modeled with an autoregressive transformer architecture. A discrete codebook provides the interface between these architectures and a patch-based discriminator enables strong compression while retaining high perceptual quality. This method introduces the efficiency of convolutional approaches to transformer based high resolution image synthesis.

3.1. Learning an Effective Codebook of Image Constituents for Use in Transformers

To utilize the highly expressive transformer architecture for image synthesis, we need to express the constituents of an image in the form of a sequence. Instead of building on individual pixels, complexity necessitates an approach that uses a discrete codebook of learned representations, such that any image \( x \in \mathbb{R}^{H \times W \times 3} \) can be represented by a spatial collection of codebook entries \( z_q \in \mathbb{R}^{h \times w \times n_z} \), where \( n_z \) is the dimensionality of codes. An equivalent representation is a sequence of \( h \times w \) indices which specify the respective entries in the learned codebook. To effectively learn such a discrete spatial codebook, we propose to directly incorporate the inductive biases of CNNs and incorporate ideas from neural discrete representation learning [63]. First, we learn a convolutional model consisting of an encoder \( E \) and a decoder \( G \), such that taken together, they learn to repre-
sent images with codes from a learned, discrete codebook \( \mathcal{Z} = \{z_k\}_{k=1}^K \subset \mathbb{R}^{n_z} \) (see Fig. 2 for an overview). More precisely, we approximate a given image \( x \) by \( \hat{x} = G(z_q) \).

We obtain \( z_q \) using the encoding \( \hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z} \) and a subsequent element-wise quantization \( q(\cdot) \) of each spatial code \( \hat{z}_{ij} \in \mathbb{R}^{n_z} \) onto its closest codebook entry \( z_k \):

\[
z_q = q(\hat{z}) := \left( \arg \min_{z_k \in \mathcal{Z}} \| \hat{z}_{ij} - z_k \| \right) \in \mathbb{R}^{h \times w \times n_z}.
\]  

(2)

The reconstruction \( \hat{x} \approx x \) is then given by

\[
\hat{x} = G(z_q) = G(q(E(x))) .
\]  

(3)

Backpropagation through the non-differentiable quantization operation in Eq. (3) is achieved by a straight-through gradient estimator, which simply copies the gradients from the decoder to the encoder [3], such that the model and codebook can be trained end-to-end via the loss function

\[
\mathcal{L}_{VQ}(E, G, \mathcal{Z}) = \| x - \hat{x} \|^2 + \| \text{sg}[E(x)] - z_q \|^2 \nonumber + \beta \| \text{sg}[z_q] - E(x) \|^2 .
\]  

(4)

Here, \( \mathcal{L}_{\text{rec}} = \| x - \hat{x} \|^2 \) is a reconstruction loss, \( \text{sg}[\cdot] \) denotes the stop-gradient operation, and \( \| \text{sg}[z_q] - E(x) \|^2 \) is the so-called “commitment loss” with weighting factor \( \beta \) [63].

Learning a Perceptually Rich Codebook Using transformers to represent images as a distribution over latent image constituents requires us to push the limits of compression and learn a rich codebook. To do so, we propose VQGAN, a variant of the original VQVAE, and use a discriminator and perceptual loss [36, 26, 35, 16] to keep good perceptual quality at increased compression rate. Note that this is in contrast to previous works which applied pixel-based [62, 53] and transformer-based autoregressive models [8] on top of only a shallow quantization model. More specifically, we replace the \( L_2 \) loss used in [53] for \( \mathcal{L}_{\text{rec}} \) by a perceptual loss and introduce an adversarial training procedure with a patch-based discriminator \( D \) [25] that aims to differentiate between real and reconstructed images:

\[
\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = \| \log D(x) + \log (1 - D(\hat{x})) \|.
\]  

(5)

The complete objective for finding the optimal compression model \( Q^* = \{E^*, G^*, \mathcal{Z}^*\} \) then reads

\[
Q^* = \arg \min_{E,G,Z} \max_D \mathbb{E}_{x \sim p(x)} \left[ \mathcal{L}_{VQ}(E, G, \mathcal{Z}) + \lambda \mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) \right] .
\]  

(6)

where we compute the adaptive weight \( \lambda \) according to

\[
\lambda = \frac{\nabla_{G_L} [\mathcal{L}_{\text{rec}}]}{\nabla_{G_L} [\mathcal{L}_{\text{GAN}}]} + \delta
\]  

(7)

where \( \mathcal{L}_{\text{rec}} \) is the perceptual reconstruction loss [71], \( \nabla_{G_L} [\cdot] \) denotes the gradient of its input w.r.t. the last layer \( L \) of the decoder, and \( \delta = 10^{-6} \) is used for numerical stability. To aggregate context from everywhere, we apply a single attention layer on the lowest resolution. This training procedure significantly reduces the sequence length when unrolling the latent code and thereby enables the application of powerful transformer models.

3.2. Learning the Composition of Images with Transformers

Latent Transformers With \( E \) and \( G \) available, we can now represent images in terms of the codebook-indices of their encodings. More precisely, the quantized encoding of an image \( x \) is given by \( z_q = q(E(x)) \in \mathbb{R}^{h \times w \times n_z} \) and is equivalent to a sequence \( s \in \{0, \ldots, |\mathcal{Z}|-1\}^{h \times w} \) of indices from the codebook, which is obtained by replacing each code by its index in the codebook \( \mathcal{Z} \):

\[
s_{ij} = k \text{ such that } (z_q)_{ij} = z_k .
\]  

(8)

By mapping indices of a sequence \( s \) back to their corresponding codebook entries, \( z_q = (z_{s_{ij}}) \) is readily recovered and decoded to an image \( \hat{x} = G(z_q) \).

Thus, after choosing some ordering of the indices in \( s \), image-generation can be formulated as autoregressive next-index prediction: Given indices \( s_{<i} \), the transformer learns to predict the distribution of possible next indices, i.e. \( p(s_i|s_{<i}) \) to compute the likelihood of the full representation as \( p(s) = \prod_i p(s_i|s_{<i}) \). This allows us to directly maximize the log-likelihood of the data representations:

\[
\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)} \left[ -\log p(s) \right] .
\]  

(9)

Conditioned Synthesis In many image synthesis tasks a user demands control over the generation process by providing additional information from which an example shall be synthesized. This information, which we will call \( c \), could be a single label describing the overall image class or even another image itself. The task is then to learn the likelihood of the sequence given this information \( c \):

\[
p(s|c) = \prod_i p(s_i|s_{<i}, c).
\]  

(10)

If the conditioning information \( c \) has spatial extent, we first learn another VQGAN to obtain again an index-based representation \( r \in \{0, \ldots, |\mathcal{Z}|-1\}^{h \times w_c} \) with the newly obtained codebook \( \mathcal{Z}_c \). Due to the autoregressive structure of the transformer, we can then simply prepend \( r \) to \( s \) and restrict the computation of the negative log-likelihood to entries \( p(s_i|s_{<i}, r) \). This “decoder-only” strategy has also been successfully used for text-summarization tasks [40].
Generating High-Resolution Images The attention mechanism of the transformer puts limits on the sequence length \( h \cdot w \) of its inputs \( s \). While we can adapt the number of downsampling blocks \( m \) of our \textit{VQGAN} to reduce images of size \( H \times W \) to \( h = H/2^m \times w = W/2^m \), we observe degradation of the reconstruction quality beyond a critical value of \( m \), which depends on the considered dataset. To generate images in the megapixel regime, we therefore have to work patch-wise and crop images to restrict the length of \( s \) to a maximally feasible size during training. To sample images, we then use the transformer in a sliding-window manner as illustrated in Fig. 3. Our \textit{VQGAN} ensures that the available context is still sufficient to faithfully model images, as long as either the statistics of the dataset are approximately spatially invariant or spatial conditioning information is available. In practice, this is not a restrictive requirement, because when it is violated, \textit{i.e.} unconditional image synthesis on aligned data, we can simply condition on image coordinates, similar to [38].

4. Experiments
This section evaluates the ability of our approach to retain the advantages of transformers over their convolutional counterparts (Sec. 4.1) while integrating the effectiveness of convolutional architectures to enable high-resolution image synthesis (Sec. 4.2). Furthermore, in Sec. 4.3, we investigate how codebook quality affects our approach. We close the analysis by providing a quantitative comparison to a wide range of existing approaches for generative image synthesis in Sec. 4.4. Based on initial experiments, we usually set \( |Z| = 1024 \) and train all subsequent transformer models to predict sequences of length \( 16 \cdot 16 \), as this is the maximum feasible length to train a GPT2-medium architecture (307 M parameters) [51] on a GPU with 12GB VRAM. More details on architectures and hyperparameters can be found in the appendix (Tab. 6 and Tab. 7).

4.1. Attention Is All You Need in the Latent Space
Transformers show state-of-the-art results on a wide variety of tasks, including autoregressive image modeling. However, evaluations of previous works were limited to transformers working directly on (low-resolution) pixels [48, 12, 24], or to deliberately shallow pixel encodings [8]. This raises the question if our approach retains the advantages of transformers over convolutional approaches.

To answer this question, we use a variety of conditional and unconditional tasks and compare the performance between our transformer-based approach and a convolutional approach. For each task, we train a \textit{VQGAN} with \( m = 4 \) downsampling blocks, and, if needed, another one for the conditioning information, and then train both a transformer and a PixelSNAIL [10] model on the same representations, as the latter has been used in previous state-of-the-art two-stage approaches [53]. For a thorough comparison, we vary the model capacities between 85M and 310M parameters and adjust the number of layers in each model to match one another. We observe that PixelSNAIL trains roughly twice as fast as the transformer and thus, for a fair comparison, report the negative log-likelihood both for the same amount of training time (\textit{P-SNAIL time}) and for the same amount of training steps (\textit{P-SNAIL steps}).

Results Tab. 1 reports results for unconditional image modeling on \textit{ImageNet} (IN) [14], \textit{Restricted ImageNet} (RIN) [57], consisting of a subset of animal classes from ImageNet, \textit{LSUN Churches and Towers} (LSUN-CT) [69], and for conditional image modeling of RIN conditioned on depth maps obtained with the approach of [52] (D-RIN) and of landscape images collected from Flickr conditioned on semantic layouts (S-FLCKR) obtained with the approach of [7]. Note that for the semantic layouts, we train the first-stage using a cross-entropy reconstruction loss due to their discrete nature. The results show that the transformer consistently outperforms PixelSNAIL across all tasks when trained for the same amount of time and the gap increases even further when trained for the same number of steps. These results demonstrate that gains of transformers carry over to our proposed two-stage setting.

4.2. A Unified Model for Image Synthesis Tasks
The versatility and generality of the transformer architecture makes it a promising candidate for image synthesis. In the conditional case, additional information \( c \) such as class labels or segmentation maps are used and the goal is to learn the distribution of images as described in Eq. (10). Using the same setting as in Sec. 4.1 (\textit{i.e.} image size \( 256 \times 256 \),
latent size $16 \times 16$), we perform various conditional image synthesis experiments:

(i): **Semantic image synthesis**, where we condition on semantic segmentation masks of ADE20K [72], a web-scraped landscapes dataset (S-FLCKR) and COCO-Stuff [6]. Results are depicted in Figure 4, 5 and Fig. 6.

(ii): **Structure-to-image**, where we use either depth or edge information to synthesize images from both RIN and IN (see Sec. 4.1). The resulting depth-to-image and edge-to-image translations are visualized in Fig. 4 and Fig. 6.

(iii): **Pose-guided synthesis**: Instead of using the semantically rich information of either segmentation or depth maps, Fig. 4 shows that the same approach as for the previous experiments can be used to build a shape-conditional generative model on the DeepFashion [41] dataset.

(iv): **Stochastic superresolution**, where low-resolution images serve as the conditioning information and are thereby upsampled. We train our model for an upsampling factor of 8 on ImageNet and show results in Fig. 6.

(v): **Class-conditional image synthesis**: Here, the conditioning information $c$ is a single index describing the class label of interest. Results on conditional sampling for the RIN dataset are demonstrated in Fig. 4.

All of these examples make use of the same methodology. Instead of requiring task specific architectures or modules, the flexibility of the transformer allows us to learn appropriate interactions for each task, while the VQGAN — which can be reused across different tasks — leads to short sequence lengths. In combination, the presented approach can be understood as an efficient, general purpose mechanism for conditional image synthesis. Note that additional results for each experiment can be found in the appendix, Sec. C.

**High-Resolution Synthesis** The sliding window approach introduced in Sec. 3.2 enables image synthesis beyond a resolution of $256 \times 256$ pixels. We evaluate this approach on unconditional image generation on LSUN-CT and FacesHQ (see Sec. 4.3) and conditional synthesis on D-RIN, COCO-Stuff and S-FLCKR, where we show results in Fig. 1, 6 and the supplementary (Fig. 17-27). Note that this approach can in principle be used to generate images of arbitrary ratio and size, given that the image statistics of the dataset of interest are approximately spatially invariant or spatial information is available. Impressive results can be achieved by applying this method to image generation from semantic layouts on S-FLCKR, where a strong VQGAN can be learned with $m = 5$, so that its codebook together with the conditioning information provides the transformer with enough context for image generation in the megapixel regime.

**4.3. Building Context-Rich Vocabularies**

How important are context-rich vocabularies? To investigate this question, we ran experiments where the transformer architecture is kept fixed while the amount of context encoded into the representation of the first stage is varied through the number of downsampling blocks of our VQGAN. We specify the amount of context encoded in terms of reduction factor in the side-length between image inputs and the resulting representations, i.e. a first stage encoding of images of size $H \times W$ into discrete codes of size $H/f \times W/f$ is denoted by a factor $f$. For $f = 1$, we reproduce the approach of [8] and replace our VQGAN by a k-means clustering of RGB values with $k = 512$.

During training, we always crop images to obtain inputs of size $16 \times 16$ for the transformer, i.e. when modeling images with a factor $f$ in the first stage, we use crops of size $16f \times 16f$. To sample from the models, we always apply them in a sliding window manner as described in Sec. 3.
Figure 5. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832, 1024×416 and 1280×240 pixels. Best viewed zoomed in. A larger visualization can be found in the appendix, see Fig. 17.

Results Fig. 7 shows results for unconditional synthesis of faces on FacesHQ, the combination of CelebA-HQ [27] and FFHQ [29]. It clearly demonstrates the benefits of powerful VQGANs by increasing the effective receptive field of the transformer. For small receptive fields, or equivalently small \( f \), the model cannot capture coherent structures. For an intermediate value of \( f = 8 \), the overall structure of images can be approximated, but inconsistencies of facial features such as a half-bearded face and of viewpoints in different parts of the image arise. Only our full setting of \( f = 16 \) can synthesize high-fidelity samples. For analogous results in the conditional setting on S-FLCKR, we refer to the appendix (Fig. 10 and Sec. B).

To assess the effectiveness of our approach quantitatively, we compare results between training a transformer directly on pixels, and training it on top of a VQGAN’s latent code with \( f = 2 \), given a fixed computational budget. Again, we follow [8] and learn a dictionary of 512 RGB values on CIFAR10 to operate directly on pixel space and train the same transformer architecture on top of our VQGAN with a latent code of size \( 16 \times 16 = 256 \). We observe improvements of 18.63% for FIDs and 14.08× faster sampling of images.

Figure 6. Applying the sliding attention window approach (Fig. 3) to various conditional image synthesis tasks. Top: Depth-to-image on RIN, 2nd row: Stochastic superresolution on IN, 3rd and 4th row: Semantic synthesis on S-FLCKR, bottom: Edge-guided synthesis on IN. The resulting images vary between 368×496 and 1024×576, hence they are best viewed zoomed in.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COCO-Stuff</td>
<td>22.4</td>
<td>22.6/23.9(*)</td>
<td>111.5 (54.2)</td>
<td>70.4</td>
</tr>
<tr>
<td>ADE20K</td>
<td>35.5</td>
<td>33.9/35.7(*)</td>
<td>81.8 (41.5)</td>
<td>73.3</td>
</tr>
</tbody>
</table>

Table 2. FID score comparison for semantic image synthesis (256×256 pixels). (*): Recalculated with our evaluation protocol based on [43] on the validation splits of each dataset.

4.4. Quantitative Comparison to Existing Models

In this section we investigate how our approach quantitatively compares to existing models for generative image synthesis. In particular, we assess the performance of our model in terms of FID and compare to a variety of established models (GANs, VAEs, Flows, AR, Hybrid) on (i) semantic synthesis in Tab. 2 (where we compare to [46, 65, 31, 9]) and (ii) unconditional face synthesis in Tab. 3. Furthermore, to address a direct comparison to the original VQVAE-2 model [53], we train a class conditional ImageNet transformer on 256×256 images, using a VQGAN with \( \dim Z = 16384 \) and \( f = 16 \), and additionally compare to BigGAN [4] and MSP [18] in Tab. 4. Note that our model uses \( \simeq 10 \times \text{less} \) parameters than VQVAE-2,
Figure 7. Evaluating the importance of effective codebook for HQ-Faces (CelebA-HQ and FFHQ) for a fixed sequence length $|s| = 16 \cdot 16 = 256$. Globally consistent structures can only be modeled with a context-rich vocabulary (right). All samples are generated with temperature $t = 1.0$ and top-$k$ sampling with $k = 100$. Last row reports the speedup over the f1 baseline which operates directly on pixels and takes 7258 seconds to produce a sample on a NVIDIA GeForce GTX Titan X.

<table>
<thead>
<tr>
<th>CelebA-HQ $256 \times 256$</th>
<th>FFHQ $256 \times 256$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td><strong>FID ↓</strong></td>
</tr>
<tr>
<td>GLOW [33]</td>
<td>69.0</td>
</tr>
<tr>
<td>NVAE [63]</td>
<td>40.3</td>
</tr>
<tr>
<td>PIONEER (B.) [21]</td>
<td>39.2 (25.3)</td>
</tr>
<tr>
<td>NCPVQVAE [1]</td>
<td>24.8</td>
</tr>
<tr>
<td>VAE BM [67]</td>
<td>20.4</td>
</tr>
<tr>
<td>Style ALAE [49]</td>
<td>19.2</td>
</tr>
<tr>
<td>DC-VAE [47]</td>
<td>15.8</td>
</tr>
<tr>
<td><strong>ours</strong></td>
<td>10.7</td>
</tr>
<tr>
<td>PGGAN [57]</td>
<td>8.0</td>
</tr>
</tbody>
</table>

Table 3. FID score comparison for face image synthesis. CelebA-HQ results reproduced from [1, 47, 67, 22], FFHQ from [58, 28].

<table>
<thead>
<tr>
<th>Dataset</th>
<th><strong>ours (+R)</strong></th>
<th><strong>VQVAE-2 (+R)</strong></th>
<th><strong>BigGAN (-deep)</strong></th>
<th>MSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN 256, 50K</td>
<td>19.8 (11.2)</td>
<td>38.1 (~ 10)</td>
<td>7.1 (7.3)</td>
<td>n.a.</td>
</tr>
<tr>
<td>IN 256, 18K</td>
<td>23.5</td>
<td>n.a.</td>
<td>9.6 (9.7)</td>
<td>50.4</td>
</tr>
</tbody>
</table>

Table 4. FID score comparison for class-conditioned synthesis. "+R": classifier-based rejection sampling as proposed in VQVAE-2. FID*-values (calculated on reconstructed data, analogous to [53]): ours: 13.5 (8.1), VQVAE-2: 19 (5). BigGAN (-deep) evaluated via https://tfhub.dev/deepmind truncated at 1.0.

which has an estimated parameter count of 13.5B (estimation based on https://github.com/rosinality/vq-vae-2-pytorch). While some task-specialized GAN models report better FID scores, our approach provides a unified model that works well across a wide range of tasks while retaining the ability to encode and reconstruct images. It thereby bridges the gap between purely adversarial and likelihood-based approaches. Fig. 11, 12, 13 and Fig. 14 contain qualitative samples corresponding to the quantitative analysis in Tab. 4.

**How good is the VQGAN?** Reconstruction FIDs obtained via the codebook provide a lower bound on the achievable FID of the generative model trained on it. To quantify the performance gains of our VQGAN over VQVAE-2, we evaluate this metric on ImageNet and report results in Tab. 5. Our VQGAN outperforms VQVAE-2 while providing significantly more compression (seq. length of 256 vs. $5120 = 32^2 + 64^2$). As expected, larger versions of VQGAN (either in terms of larger codebook sizes or increased code lengths) further improve performance. Using the same hierarchical codebook setting as in VQVAE-2 with our model provides the best reconstruction FID, albeit at the cost of a very long and thus impractical sequence. Furthermore, Fig. 9 qualitatively shows that a standard VQGAN cannot achieve such compressions; the corresponding reconstruction-FIDs read: VQVAE 254.4; VQGAN 5.7. Sampling from this VQVAE cannot achieve FIDs below 254.4, whereas our VQGAN achieves 21.93 with PixelSNAIL and 11.44 with a transformer (see Tab. 3).

**5. Conclusion**

This paper addressed the fundamental challenges that previously confined transformers to low-resolution images. We proposed an approach which represents images as a composition of perceptually rich image constituents and thereby overcomes the infeasible quadratic complexity when modeling images directly in pixel space. Modeling constituents with a CNN architecture and their compositions with a transformer architecture taps into the full potential of their complementary strengths and thereby allowed us to represent the first results on high-resolution image synthesis with a transformer-based architecture. In experiments, our approach demonstrates the efficiency of convolutional inductive biases and the expressivity of transformers by synthesizing images in the megapixel range and outperforming state-of-the-art convolutional approaches. Equipped with a general mechanism for conditional synthesis, it offers many opportunities for novel neural rendering approaches.

This work has been supported by the German Research Foundation (DFG) projects 371923335, 421703927 and a hardware donation from NVIDIA corporation.
References


[31] Prateek Katiyar and Anna Khoreva. Improving augmentation and evaluation schemes for semantic image synthesis, 2021. 7


[38] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. COCOGAN: generation by parts via conditional coordinating. In ICCV, pages 4511–4520. IEEE, 2019. 5


[47] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contrastivegenerative autoencoder, 2020. 8


[50] A. Radford. Improving language understanding by generative pre-training. 2018. 1


