

Taming Transformers for High-Resolution Image Synthesis

Patrick Esser* Robin Rombach* Björn Ommer
 Heidelberg Collaboratory for Image Processing
 IWR, Heidelberg University, Germany



Figure 1. Our approach enables transformers to synthesize high-resolution images like this one, which contains 1280x460 pixels.

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 [46, 35, 36, 4]

and are increasingly adapted in other areas such as audio [9] and vision [7, 1]. 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 [1] 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 spa-

tial 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 [31]*: 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 constituent 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 [46] is that it models interactions between its inputs solely through attention [2, 22, 32] 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 [26] or images [7], 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

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

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. [34] and [47] 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. [9] and [17] 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 3×3 . 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 [33, 50, 42, 54, 53] suggest that this bias is often too restrictive.

Convolutional architectures have been used for autoregressive modeling of images [43, 44, 8] but, for low-resolution images, previous works [34, 9, 17] 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. [10] demonstrated both theoretical and empirical evidence on the advantages of first learning a data representation with a Variational Autoencoder (VAE) [23, 39], and then again learning its distribution with a VAE. [13, 48] demonstrate similar gains when using an unconditional normalizing flow for the second stage, and [40] when using a conditional normalizing flow. To improve training efficiency of Generative Adversarial Networks (GANs), [28]

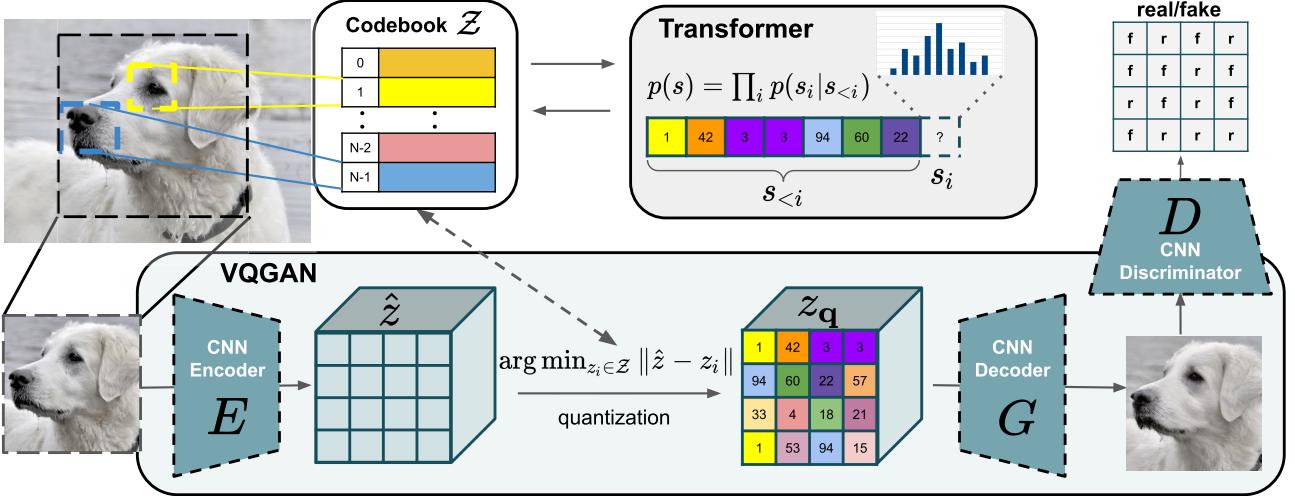


Figure 2. Our approach uses a convolutional *VQGAN* to learn a codebook of context-rich visual parts, whose composition is subsequently modeled with an autoregressive transformer architecture. A discrete codebook provides the interface between these architectures and a patch-based discriminator enables strong compression while retaining high perceptual quality. This method introduces the efficiency of convolutional approaches to transformer based high resolution image synthesis.

learns a GAN [14] on representations of an autoencoder and [15] on low-resolution wavelet coefficients which are then decoded to images with a learned generator.

[45] presents the Vector Quantised Variational Autoencoder (VQVAE), an approach to learn discrete representations of images, and models their distribution autoregressively with a convolutional architecture. [38] extends this approach to use a hierarchy of learned representations. However, these methods still rely on convolutional density estimation, which makes it difficult to capture long-range interactions in high-resolution images. [7] models images autoregressively with transformers in order to evaluate the suitability of generative pretraining to learn image representations for downstream tasks. Since input resolutions of 32×32 pixels are still quite computationally expensive [7], a VQVAE is used to encode images up to a resolution of 192×192 . In an effort to keep the learned discrete representation as spatially invariant as possible with respect to the pixels, a shallow VQVAE with small receptive field is employed. In contrast, we demonstrate that a powerful first stage, which captures as much context as possible in the learned representation, is critical to enable efficient high-resolution image synthesis with transformers.

3. Approach

Our goal is to exploit the highly promising learning capabilities of transformer models [46] and introduce them to high-resolution image synthesis up to the megapixel range. Previous work [34, 7] which applied transformers to image generation demonstrated promising results for images up to a size of 64×64 pixels but, due to the quadratically increasing cost in sequence length, cannot simply be scaled

to higher resolutions.

High-resolution image synthesis requires a model that understands the global composition of images, enabling it to generate locally realistic as well as globally consistent patterns. Therefore, instead of representing an image with pixels, we represent it as a composition of perceptually rich image constituents from a codebook. By learning an effective code, as described in Sec. 3.1, we can significantly reduce the description length of compositions, which allows us to efficiently model their global interrelations within images with a transformer architecture as described in Sec. 3.2. This approach, summarized in Fig. 2, is able to generate realistic and consistent high resolution images both in an unconditional and a conditional setting.

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 $h \cdot w$ 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 [45]. 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 $\hat{x} = G(z_{\mathbf{q}})$. We obtain $z_{\mathbf{q}}$ using the encoding $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$ and a subsequent element-wise quantization $\mathbf{q}(\cdot)$ of each spatial code $\hat{z}_{ij} \in \mathbb{R}^{n_z}$ onto its closest codebook entry z_k :

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg \min_{z_k \in \mathcal{Z}} \|\hat{z}_{ij} - z_k\| \right) \in \mathbb{R}^{h \times w \times n_z}. \quad (2)$$

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

$$\hat{x} = G(z_{\mathbf{q}}) = G(\mathbf{q}(E(x))). \quad (3)$$

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

$$\begin{aligned} \mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) = & \|x - \hat{x}\|_2^2 + \|\text{sg}[E(x)] - z_{\mathbf{q}}\|_2^2 \\ & + \beta \|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2. \end{aligned} \quad (4)$$

Here, $\|x - \hat{x}\|_2^2$ is a reconstruction loss, $\text{sg}[\cdot]$ denotes the stop-gradient operation, and $\|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2$ is the so-called ‘‘commitment loss’’ with weighting factor β [45].

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 [25, 19, 24, 12] to keep good perceptual quality at increased compression rate. Note that this is in contrast to previous works which applied pixel-based [44, 38] and transformer-based autoregressive models [7] on top of only a shallow quantization model. More specifically, we replace the L_2 reconstruction in Eq. (4) by a perceptual loss and introduce an adversarial training procedure with a patch-based discriminator D [18] that aims to differentiate between real and reconstructed images:

$$\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = [\log D(x) + \log(1 - D(\hat{x}))] \quad (5)$$

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

$$\begin{aligned} \mathcal{Q}^* = \arg \min_{E, G, \mathcal{Z}} \max_D \mathbb{E}_{x \sim p(x)} \Big[& \mathcal{L}_{\text{VQ}}(E, G, \mathcal{Z}) \\ & + \lambda \mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) \Big], \end{aligned} \quad (6)$$

where we compute the adaptive weight λ according to

$$\lambda = \frac{\nabla_{G_L} [\mathcal{L}_{\text{rec}}]}{\nabla_{G_L} [\mathcal{L}_{\text{GAN}}] + \delta} \quad (7)$$

where \mathcal{L}_{rec} is the perceptual reconstruction loss [51], $\nabla_{G_L} [\cdot]$ 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 $z_{\mathbf{q}} = \mathbf{q}(E(x)) \in \mathbb{R}^{h \times w \times n_z}$ and is equivalent to a sequence $s \in \{0, \dots, |\mathcal{Z}| - 1\}^{h \times w}$ of indices from the codebook, which is obtained by replacing each code by its index in the codebook \mathcal{Z} :

$$s_{ij} = k \text{ such that } (z_{\mathbf{q}})_{ij} = z_k. \quad (8)$$

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

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:

$$\mathcal{L}_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)} [-\log p(s)]. \quad (9)$$

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 :

$$p(s | c) = \prod_i p(s_i | s_{<i}, c). \quad (10)$$

If the conditioning information c has spatial extent, we first learn another *VQGAN* to obtain again an index-based representation $r \in \{0, \dots, |\mathcal{Z}_c| - 1\}^{h_c \times w_c}$ with the newly obtained codebook \mathcal{Z}_c ¹. 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 [29].

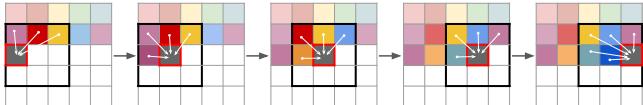


Figure 3. Sliding attention window.

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 *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 *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, *i.e.* unconditional image synthesis on aligned data, we can simply condition on image coordinates, similar to [27].

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, we investigate how codebook quality (Sec. 4.3) and codebook orderings (Sec. 4.4) affect our approach.

Based on initial experiments, we choose the size of the codebook as $|\mathcal{Z}| = 1024$ 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) [36] on a GPU with 12GB VRAM. More details on architectures and hyperparameters can be found in the appendix (Tab. 2 and Tab. 3).

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 [34, 9, 17], or to deliberately shallow pixel encodings [7]. This raises the question if our approach retains the advantages of transformers over convolutional approaches.

¹Note that this is completely analogous to the procedure that generates the data representation s .

Data / # params	Negative Log-Likelihood (NLL)		
	Transformer <i>P-SNAIL steps</i>	Transformer <i>P-SNAIL time</i>	PixelSNAIL fixed time
RIN / 85M	4.78	4.84	4.96
LSUN-CT / 310M	4.63	4.69	4.89
IN / 310M	4.78	4.83	4.96
D-RIN / 180 M	4.70	4.78	4.88
S-FLCKR / 310 M	4.49	4.57	4.64

Table 1. Comparing Transformer and PixelSNAIL architectures across different datasets and model sizes. For all settings, transformers outperform the state-of-the-art model from the PixelCNN family, PixelSNAIL in terms of NLL. This holds both when comparing NLL at fixed times (PixelSNAIL trains roughly 2 times faster) and when trained for a fixed number of steps. See Sec. 4.1 for the abbreviations.

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 *VQGAN* with $m = 4$ downsampling blocks, and, if needed, another one for the conditioning information, and then train both a transformer and a PixelSNAIL [8] model on the same representations, as the latter has been used in previous state-of-the-art two-stage approaches [38]. 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 (*P-SNAIL time*) and for the same amount of training steps (*P-SNAIL steps*).

Results Tab. 1 reports results for unconditional image modeling on *ImageNet* (IN) [11], *Restricted ImageNet* (RIN) [41], consisting of a subset of animal classes from ImageNet, *LSUN Churches and Towers* (LSUN-CT) [49], and for conditional image modeling of RIN conditioned on depth maps obtained with the approach of [37] (D-RIN) and of landscape images collected from Flickr conditioned on semantic layouts (S-FLCKR) obtained with the approach of [6]. 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 synthe-

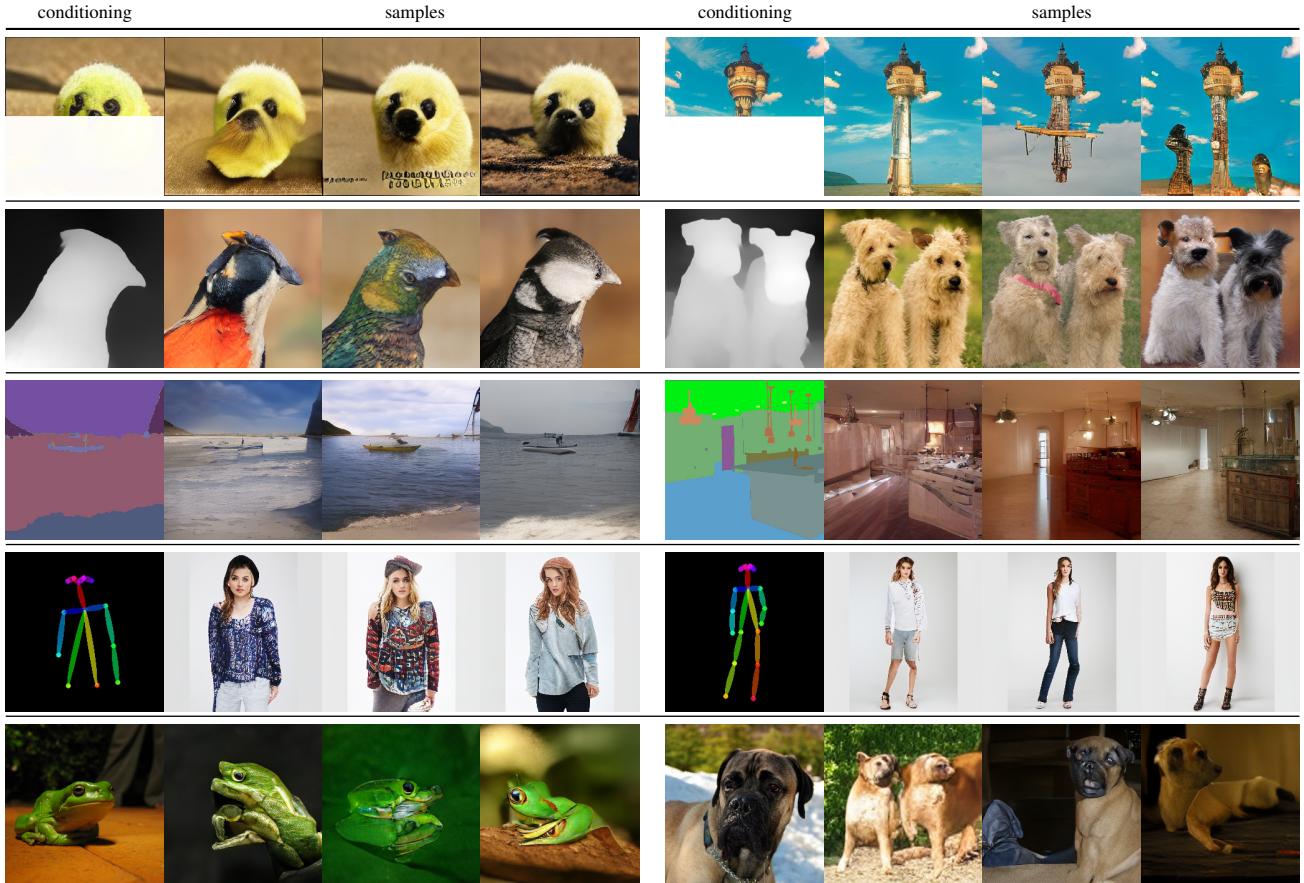


Figure 4. Transformers within our setting unify a wide range of image synthesis tasks. We show 256×256 synthesis results across different conditioning inputs and datasets, all obtained with the same approach to exploit inductive biases of effective CNN based *VQGAN* architectures in combination with the expressivity of transformer architectures. Top row: Completions from unconditional training on ImageNet. 2nd row: Depth-to-Image on RIN. 3rd row: Semantically guided synthesis on COCO-Stuff (left) and ADE20K (right). 4th row: Pose-guided person generation on DeepFashion. Bottom row: Class-conditional samples on RIN.

sis. 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 (i.e. image size 256×256 , latent size 16×16), we perform various conditional image synthesis experiments:

(i): **Semantic image synthesis**, where we condition on semantic segmentation masks of ADE20K [52], a web-scraped landscapes dataset (S-FLCKR) and COCO-Stuff [5]. 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 [30] 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.



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 13.

High-Resolution Synthesis The sliding window approach introduced in Sec. 3.2 enables image synthesis beyond a resolution of 256×256 pixels. We evaluate this approach on unconditional image generation on LSUN-CT and FacesHQ (see Sec. 4.3) and conditional synthesis on DRIN, COCO-Stuff and S-FLCKR, where we show results in Fig. 1, 6 and the supplementary (Fig. 13-23). 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

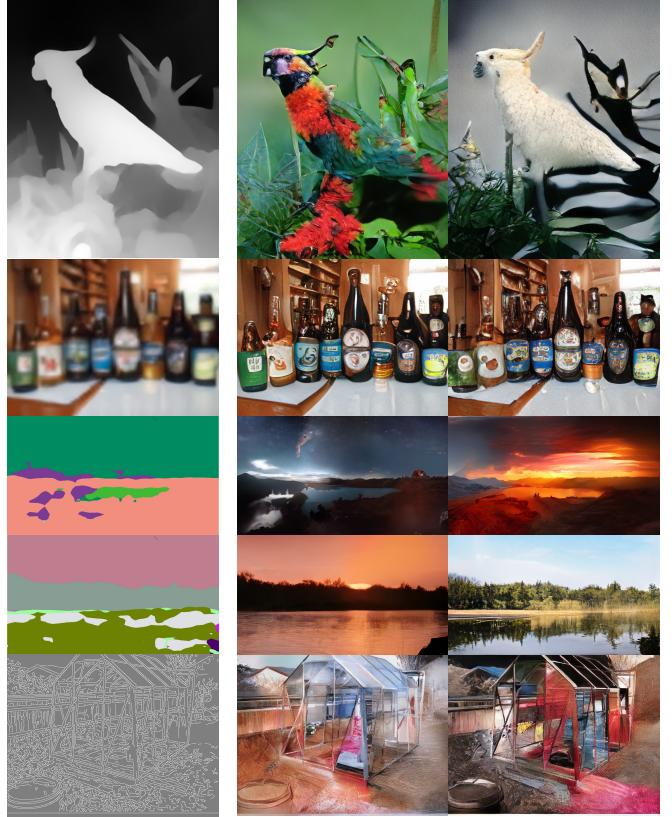


Figure 6. Applying the sliding attention window approach (Fig. 3) to various conditional image synthesis tasks. Top: Depth-to-image on RIN, 2nd row: Stochastic superresolution on IN, 3rd and 4th row: Semantic synthesis on S-FLCKR, bottom: Edge-guided synthesis on IN. The resulting images vary between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

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 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 [7] 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 16×16 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.

Results Fig. 7 shows results for unconditional synthesis of faces on *FacesHQ*, the combination of *CelebA-HQ* [20] and *FFHQ* [21]. It clearly demonstrates the benefits of powerful *VQGANs* by increasing the effective receptive field of the transformer. For small receptive fields, or

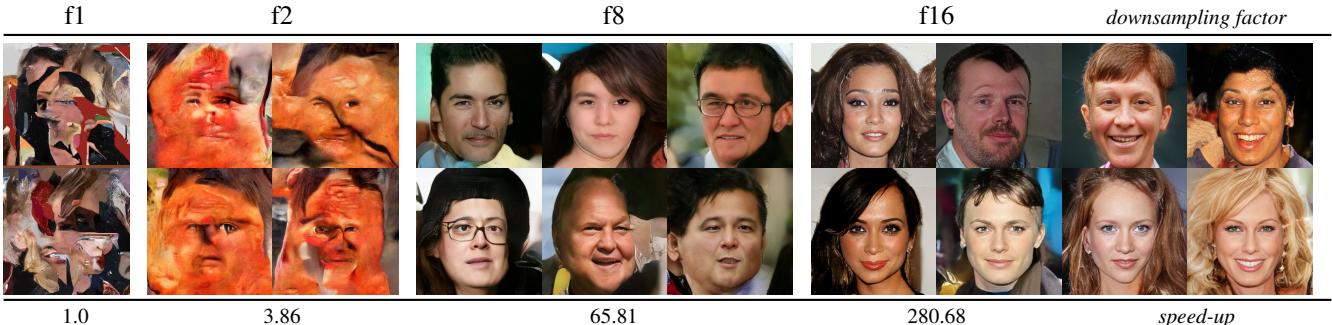


Figure 7. Evaluating the importance of effective codebook for HQ-Faces (CelebA-HQ and FFHQ) for a fixed sequence length $|s|=16 \cdot 16 = 256$. Globally consistent structures can only be modeled with a context-rich vocabulary (right). All samples are generated with temperature $t = 1.0$ and top- k sampling 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.

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 additionally 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 [7] and learn a dictionary of 512 RGB values on CIFAR10 to operate directly on pixel space and additionally 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 FID scores and 14.08 times faster sampling of images.

4.4. On the Ordering of Image Representations

For the “classical” domain of transformer models, NLP, the order of tokens is defined by the language at hand. For images and their discrete representations, in contrast, it is not clear which linear ordering to use. In particular, our sliding-window approach depends on a row-major ordering and we thus investigate the performance of the following five different permutations of the input sequence of codebook indices: (i) **row major**, or *raster scan order*, where the image representation is unrolled from top left to bottom right. (ii) **spiral out**, which incorporates the prior assumption that most images show a *centered* object. (iii) **z-curve**, also known as *z-order* or *morton curve*, which introduces the prior of *preserved locality* when mapping a 2D image representation onto a 1D sequence. (iv) **subsample**, where prefixes correspond to subsampled representations. (v) **alternate**, which is related to *row major*, but alternates the direction of unrolling every row. (vi) **spiral in**, a reversed version of *spiral out* which provides the most context for predicting

the center of the image. A graphical visualization of these permutation variants is shown in the supplement. Given a *VQGAN* trained on ImageNet, we train a transformer for each permutation in a controlled setting, i.e. we fix initialization and computational budget.

Results Sec. D contains the evolution of negative log-likelihood for each variant as a function of training iterations, with final values given by (i) 4.767, (ii) 4.889, (iii) 4.810, (iv) 5.015, (v) 4.812, (vi) 4.901. Interestingly, *row major* performs best in terms of this metric, whereas the more hierarchical *subsample* prior does not induce any helpful bias. We also include qualitative samples in the appendix (Fig. 31) and observe that the two worst performing models in terms of NLL (*subsample* and *spiral in*) tend to produce more textural samples, while the other variants synthesize samples with much more recognizable structures. Overall, we can conclude that the autoregressive codebook modeling is *not* permutation-invariant, but the common *row major* ordering [44, 7] outperforms other orderings.

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.

Taming Transformers for High-Resolution Image Synthesis

Supplementary Material

The supplementary material for our work *Taming Transformers for High-Resolution Image Synthesis* is structured as follows: First, in Sec. A, we present hyperparameters and architectures which were used to train our models. Next, extending the discussion of Sec. 4.3, Sec. B presents additional evidence for the importance of perceptually rich codebooks and its interpretation as a trade-off between reconstruction fidelity and sampling capability. Additional results on high-resolution image synthesis for a wide range of tasks are then presented in Sec. C. Finally, Sec. D contains the full results regarding the ordering of image representations, as discussed in Sec. 4.4.

A. Implementation Details

The hyperparameters for all experiments presented in the main paper and supplementary material can be found in Tab. 3. These hyperparameters are set such that each transformer model can be trained with a batch-size of at least 2 on a GPU with 12GB VRAM, but we generally train on 2-4 GPUs with an accumulated VRAM of 48 GB. If hardware permits, 16-bit precision training is enabled.

VQGAN Architecture The architecture of our convolutional encoder and decoder models used in the *VQGAN* experiments is described in Tab. 2. Note that we adopt the compression rate by tuning the number of downsampling steps m .

Transformer Architecture Our transformer model is identical to the GPT2 architecture (“small” to “medium”) [36] and we vary its capacity mainly through varying the amount of layers (see Tab. 3). Furthermore, we generally produce samples with a temperature $t = 1.0$ and a top- k cutoff at $k = 100$.

B. On Context-Rich Vocabularies

Sec. 4.3 investigated the effect of the downsampling factor f used for encoding images. As demonstrated in Fig. 7, large factors are crucial for our approach, since they enable the transformer to model long-range interactions efficiently. However, since larger f correspond to larger compression rates, the reconstruction quality of the *VQGAN* starts to decrease after a certain point, which is analyzed in Fig. 8. The left part shows the reconstruction error (measured by LPIPS [51]) versus the negative log-likelihood obtained by the transformer for values of f ranging from 1 to 64. The latter provides a measure of the ability to model the distribution of the image representation, which increases with f . The reconstruction error on the other hand decreases with f and the qualitative results on the right part show that beyond a critical value of f , in this case $f = 16$, reconstruction errors become severe. At this point, even when the image representations are modeled faithfully, as suggested by a low negative log-likelihood, sampled images are of low-fidelity, because the reconstruction capabilities provide an upper bound on the quality that can be achieved.

Encoder	Decoder
$x \in \mathbb{R}^{H \times W \times C}$ $\text{Conv2D} \rightarrow \mathbb{R}^{H \times W \times C'}$ $m \times \{ \text{Residual Block}, \text{Downsample Block} \} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\quad \text{Residual Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\quad \text{Non-Local Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\quad \text{Residual Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\quad \text{GroupNorm, Swish, Conv2D} \rightarrow \mathbb{R}^{h \times w \times n_z}$	$z_q \in \mathbb{R}^{h \times w \times n_z}$ $\text{Conv2D} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\text{Residual Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\text{Non-Local Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $\text{Residual Block} \rightarrow \mathbb{R}^{h \times w \times C''}$ $m \times \{ \text{Residual Block}, \text{Upsample Block} \} \rightarrow \mathbb{R}^{H \times W \times C'}$ $\quad \text{GroupNorm, Swish, Conv2D} \rightarrow \mathbb{R}^{H \times W \times C}$

Table 2. High-level architecture of the encoder and decoder of our *VQGAN*. The design of the networks follows the architecture presented in [16] with no skip-connections. For the discriminator, we use a patch-based model as in [18]. Note that $h = \frac{H}{2^m}$, $w = \frac{W}{2^m}$ and $f = 2^m$.

Experiment	n_{layer}	# params [M]	n_z	$ \mathcal{Z} $	dropout	length(s)	m
RIN	12	85	64	768	0.0	512	4
c-RIN	18	128	64	768	0.0	257	4
D-RINV1	14	180	256	1024	0.0	512	4
D-RINV2	24	307	256	1024	0.0	512	4
IN	24	307	256	1024	0.0	256	4
c-IN	24	307	256	1024	0.0	257	4
IN-Edges	24	307	256	1024	0.0	512	3
IN-SR	12	153	256	1024	0.0	512	3
S-FLCKR, $f = 4$	24	307	256	1024	0.0	512	2
S-FLCKR, $f = 16$	24	307	256	1024	0.0	512	4
S-FLCKR, $f = 32$	24	307	256	1024	0.0	512	5
(FacesHQ, $f = 1$) [*]	24	307	—	512	0.0	512	—
FacesHQ, $f = 2$	24	307	256	1024	0.0	512	1
FacesHQ, $f = 4$	24	307	256	1024	0.0	512	2
FacesHQ, $f = 8$	24	307	256	1024	0.0	512	3
FacesHQ, $f = 16$	24	307	256	1024	0.0	512	4
COCO-Stuff	36	470	256	4096	0.1	512	4
ADE20K	28	405	256	4096	0.1	512	4
DeepFashion	18	129	256	1024	0.0	340	4
LSUN-CT	24	307	256	1024	0.0	256	4
CIFAR-10	24	307	256	1024	0.0	256	1

Table 3. Hyperparameters. For every experiment, we set the number of attention heads in the transformer to $n_h = 16$ and the embedding dimension to $n_e = 1024$ (except for c-RIN, D-RINV1 and DeepFashion, which use $n_e = 768$). D-RINV1 is the experiment which compares to Pixel-SNAIL in Sec. 4.1. Note that the experiment (FacesHQ, $f = 1$)^{*} does not use a learned *VQGAN* but a fixed k-means clustering algorithm as in [7] with $K = 512$ centroids. The “commitment factor” β in Eq. (4) is always set to $\beta = 0.25$. A prefix “c” refers to a class-conditional model. # params refers to the number of parameters of the transformer model; we round 307 M to 310 M in the main text.

Hence, Fig. 8 shows that we must learn perceptually rich encodings, *i.e.* encodings with a large f and perceptually faithful reconstructions. This is the goal of our *VQGAN* and Fig. 9 compares its reconstruction capabilities against a *VQVAE* [45]. Both approaches use the same architecture of Tab. 2 and a factor $f = 16$, which demonstrates how the *VQGAN* provides high-fidelity reconstructions at large factors, and thereby enables efficient high-resolution image synthesis with transformers.

To illustrate how the choice of f depends on the dataset, Fig. 10 presents results on S-FLCKR. In the left part, it shows, analogous to Fig. 7, how the quality of samples increases with increasing f . However, in the right part, it shows that reconstructions remain faithful perceptually faithful even for $f=32$, which is in contrast to the corresponding results on faces in Fig. 8. These results might be explained by a higher perceptual sensitivity to facial features as compared to textures, and allow us to generate high-resolution landscapes even more efficiently with $f = 32$.

C. Additional Results

Comparison to Image-GPT To further evaluate the effectiveness of our approach, we compare to the state-of-the-art generative transformer model on images, ImageGPT [7]. By using immense amounts of compute the authors demonstrated that transformer models can be applied to the pixel-representation of images and thereby achieved impressive results both in representation learning and image synthesis. However, as their approach is confined to pixel-space, it does not scale beyond a resolution of 192×192 . As our approach leverages a strong compression method to obtain context-rich representations of images and *then* learns a transformer model, we can synthesize images of much higher resolution. We compare both approaches in Fig. 11 and Fig. 12, where completions of images are depicted. Both plots show that our approach is able to synthesize consistent completions of dramatically increased fidelity. The results of [7] are obtained from <https://openai.com/blog/image-gpt/>.

Additional High-Resolution Results Fig. 13, 14, 15 and Fig. 16 contain additional HR results on the S-FLCKR dataset for both $f = 16$ ($m = 4$) and $f = 32$ ($m = 5$) (semantically guided). In particular, we provide an enlarged version of Fig. 5

from the main text, which had to be scaled down due to space constraints. Additionally, we use our sliding window approach (see Sec. 3) to produce high-resolution samples for the depth-to-image setting on RIN in Fig. 17 and Fig. 18, edge-to-image on IN in Fig. 19, stochastic superresolution on IN in Fig. 20, more examples on semantically guided landscape synthesis on S-FLCKR in Fig. 21 with $f = 16$ and in Fig. 22 with $f = 32$, and unconditional image generation on LSUN-CT (see Sec. 4.1) in Fig. 23. Moreover, for images of size 256×256 , we provide results for generation from semantic layout on (i) ADE20K in Fig. 24 and (ii) COCO-Stuff in Fig. 25, depth-to-image on IN in Fig. 26, pose-guided person generation in Fig. 27 and class-conditional synthesis on RIN and IN in Fig. 28 and Fig. 29, respectively.

D. On the Ordering of Image Representations

Sec. 4.4 discussed the ordering of image representations used for autoregressive prediction thereof. Fig. 30 shows an illustration of all the different variants that were analyzed, together with the evolution of negative log-likelihood for each variant. Fig. 31 contains corresponding samples of each model.

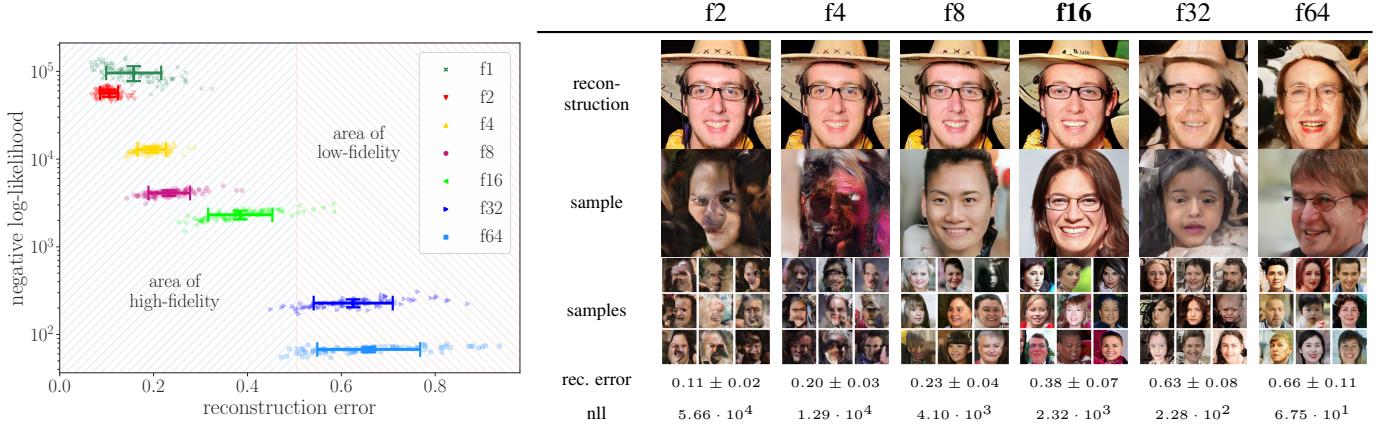


Figure 8. Trade-off between negative log-likelihood (nll) and reconstruction error. While context-rich encodings obtained with large factors f allow the transformer to effectively model long-range interactions, the reconstructions capabilities and hence quality of samples suffer after a critical value (here, $f = 16$). For more details, see Sec. B.

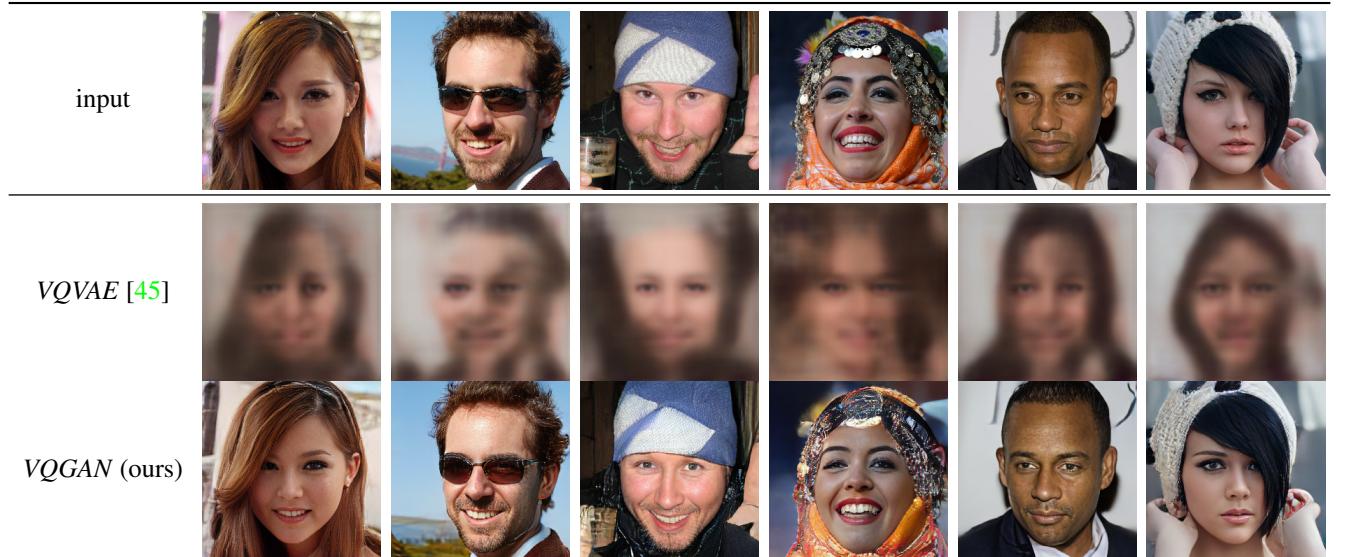


Figure 9. We compare the ability of VQVAEs and VQGANs to learn perceptually rich encodings, which allow for high-fidelity reconstructions with large factors f . Here, using the same architecture and $f = 16$, VQVAE reconstructions are blurry and contain little information about the image, whereas VQGAN recovers images faithfully. See also Sec. B.

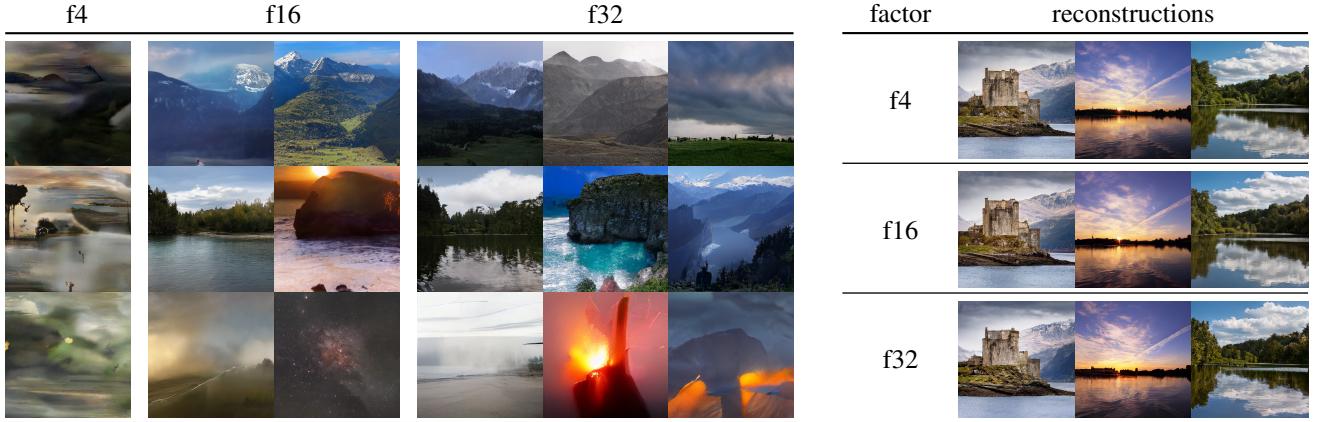


Figure 10. Samples on landscape dataset (left) obtained with different factors f , analogous to Fig. 7. In contrast to faces, a factor of $f = 32$ still allows for faithful reconstructions (right). See also Sec. B.

conditioning

ours (top) vs iGPT [7] (bottom)

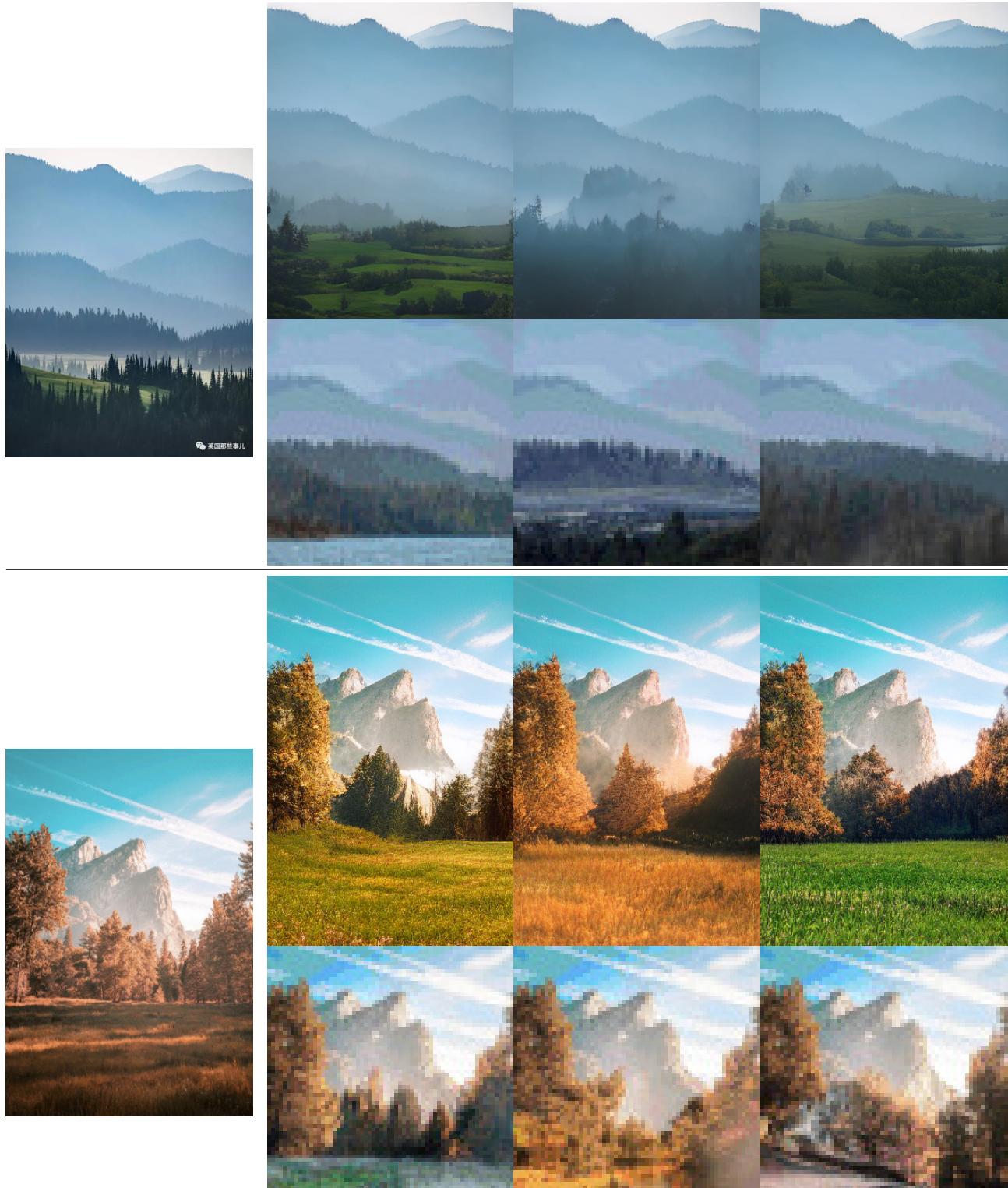


Figure 11. Comparing our approach with the pixel-based approach of [7]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [7] (bottom).

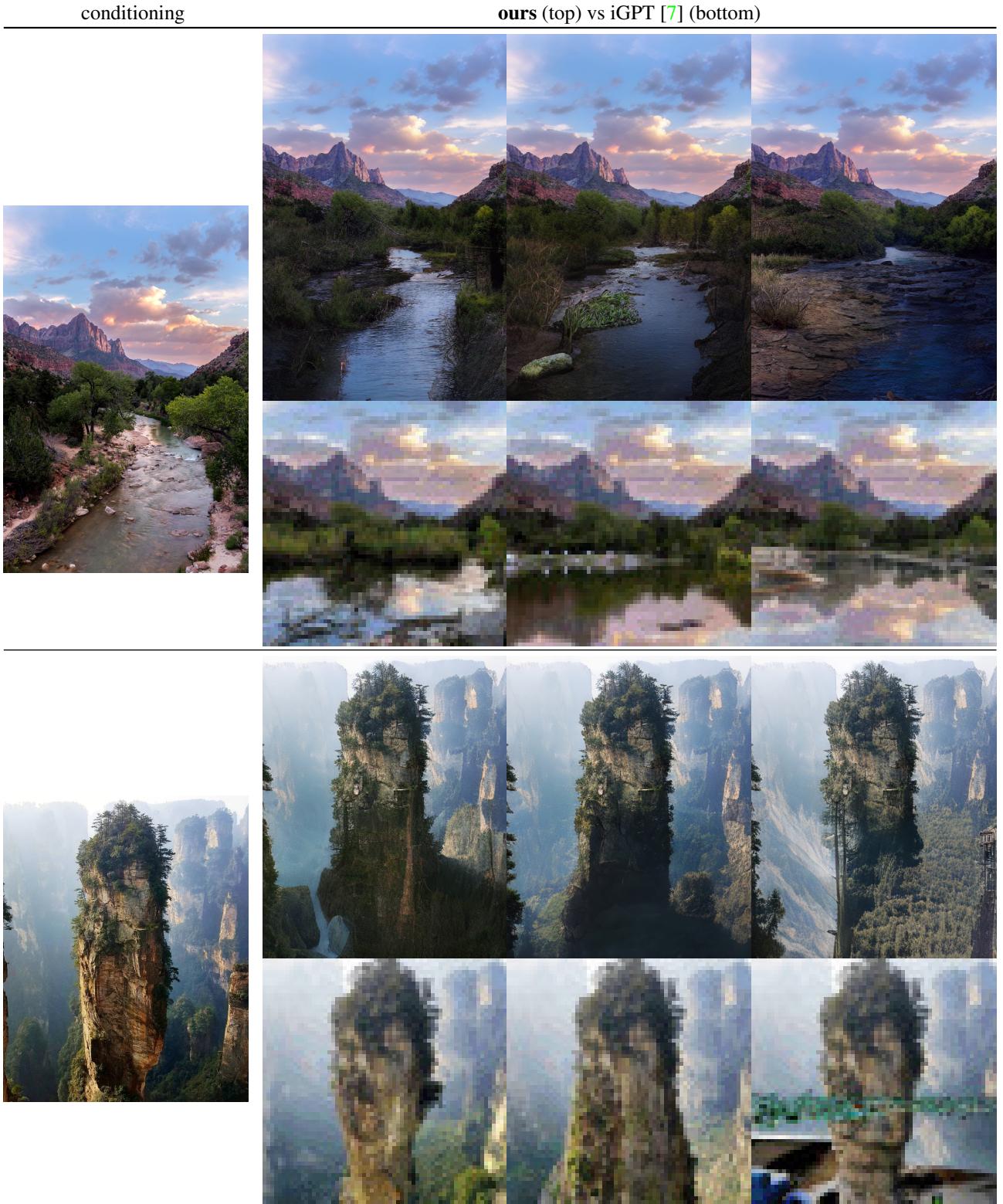


Figure 12. Comparing our approach with the pixel-based approach of [7]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [7] (bottom).

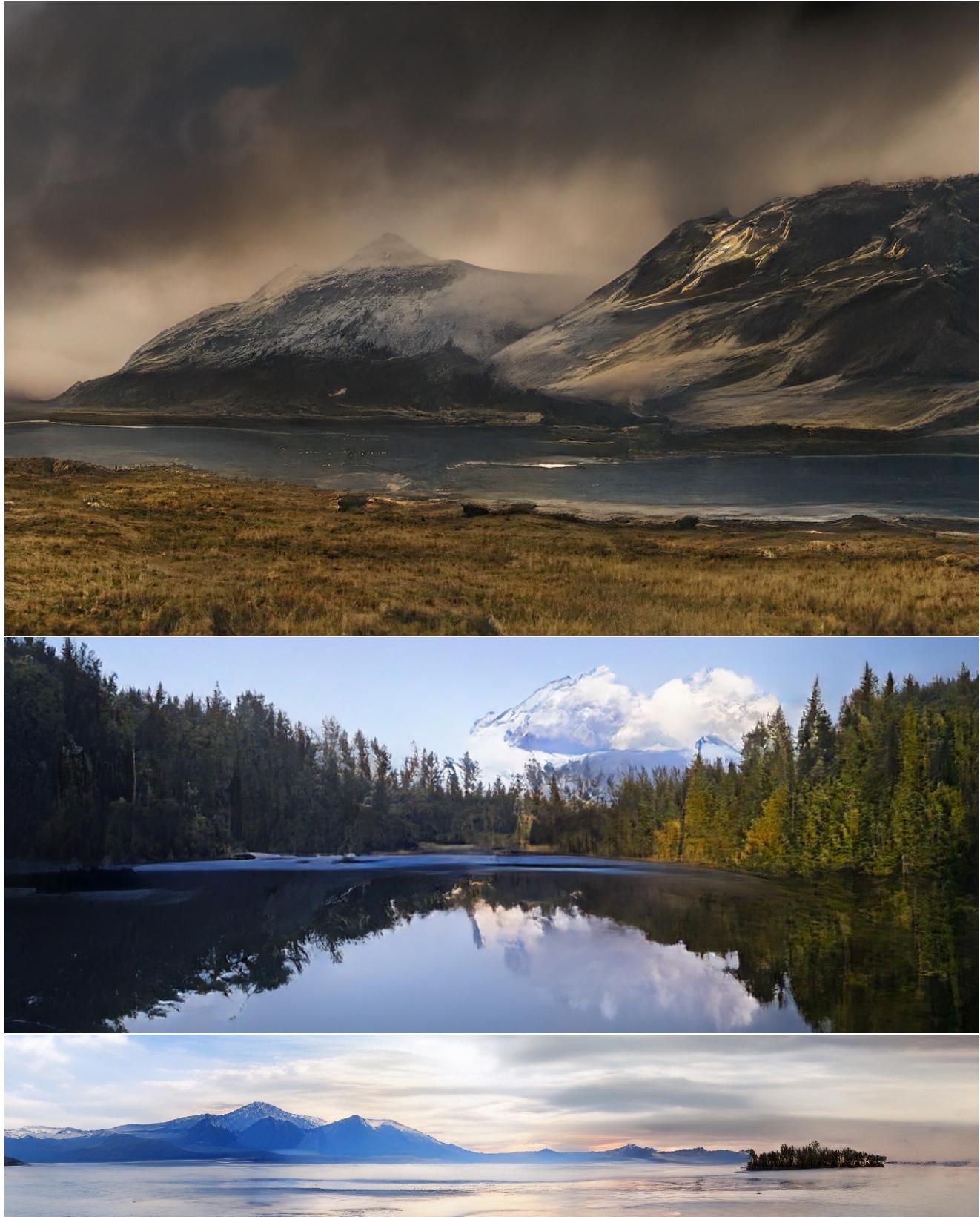


Figure 13. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832 , 1024×416 and 1280×240 pixels.

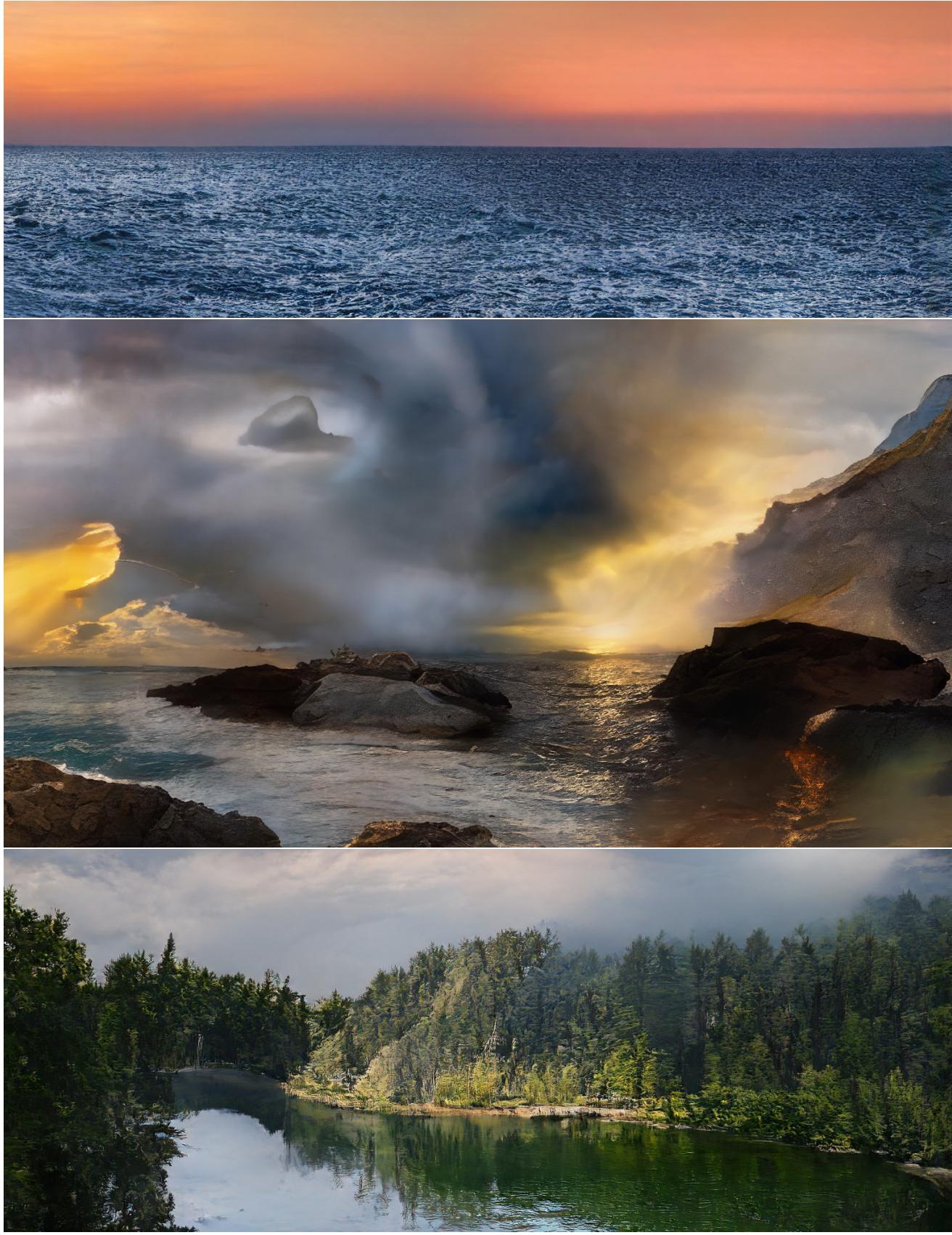


Figure 14. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1536×512 , 1840×1024 , and 1536×620 pixels.

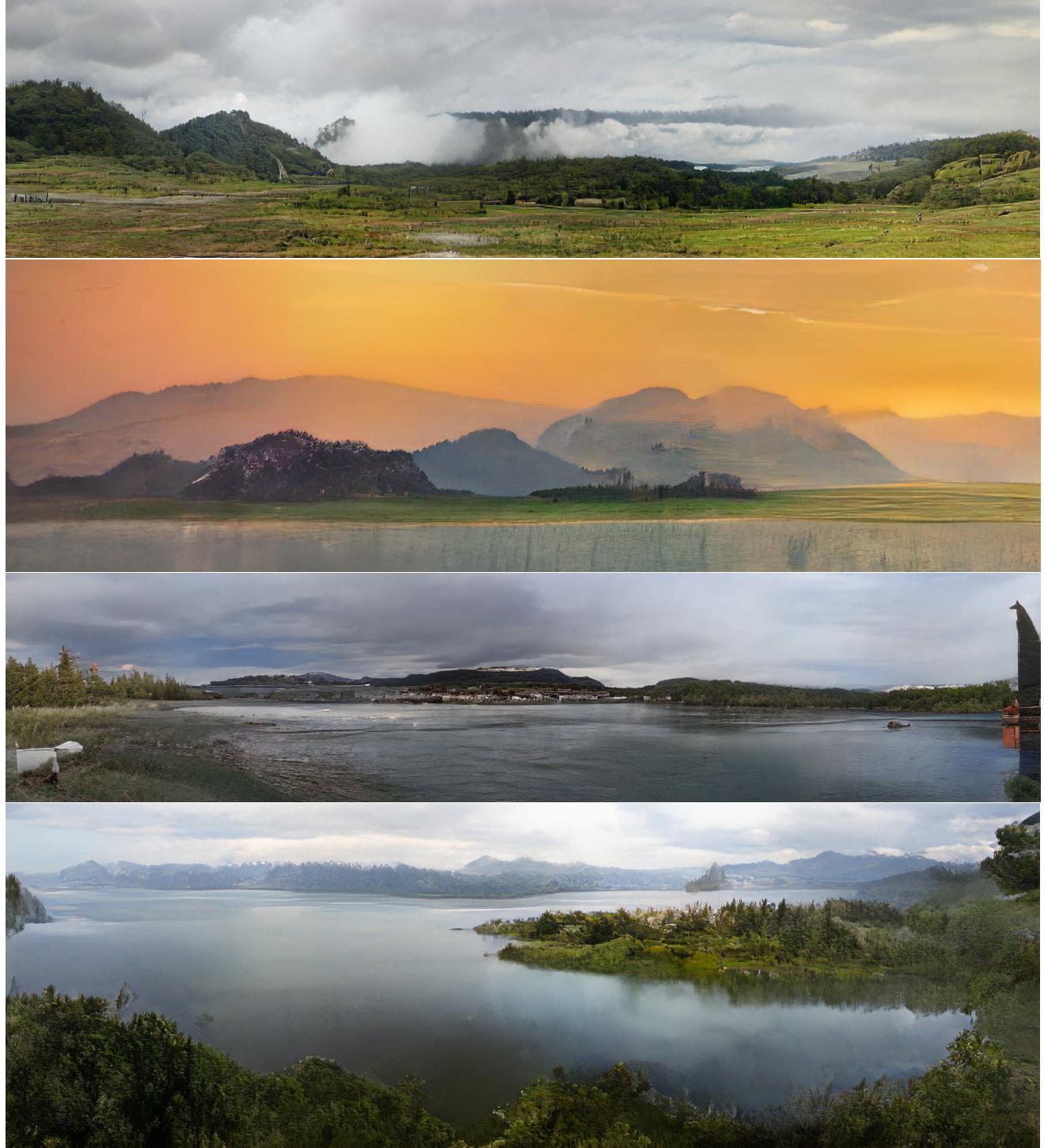


Figure 15. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 2048×512 , 1460×440 , 2032×448 and 2016×672 pixels.

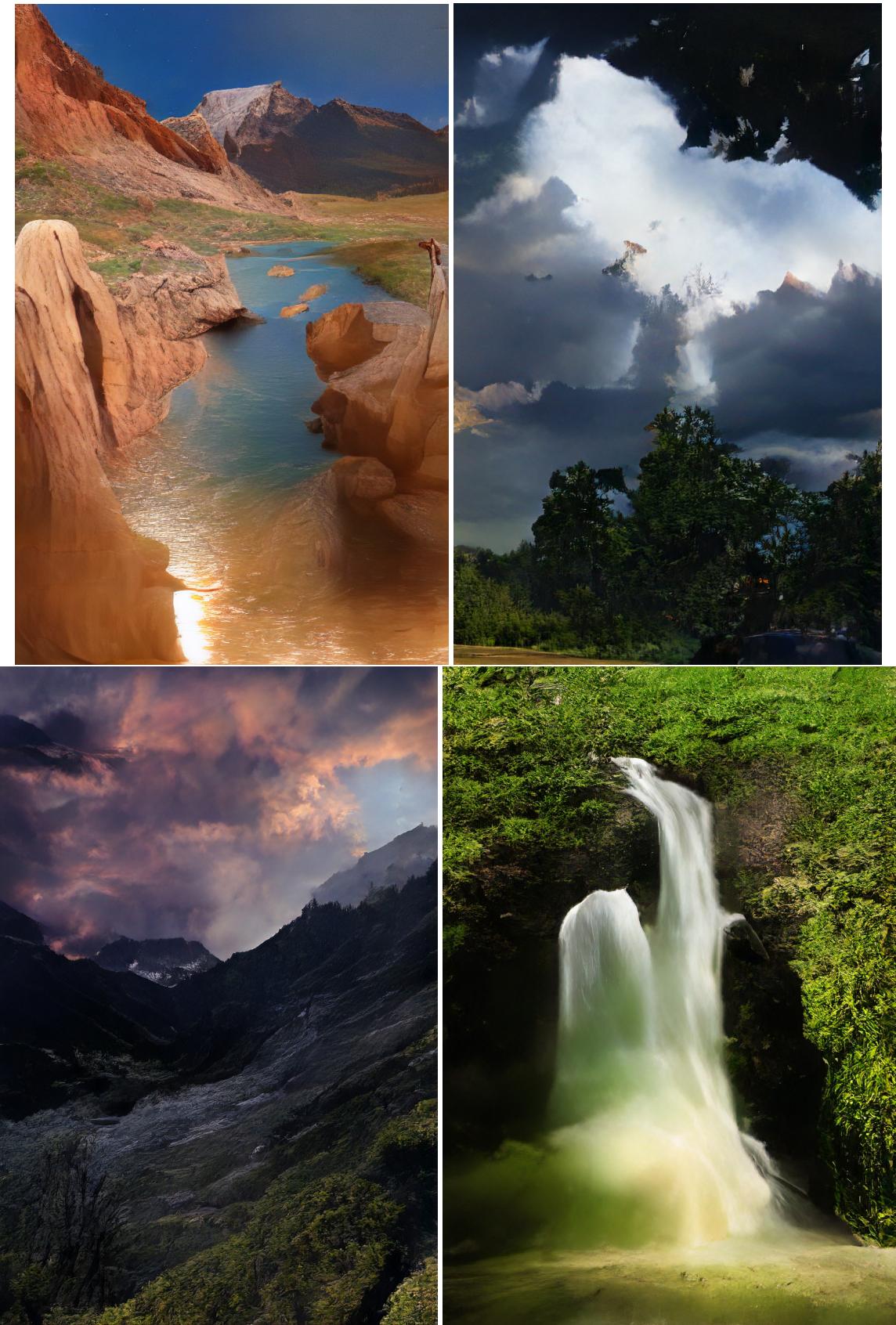


Figure 16. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832 , 1024×416 and 1280×240 pixels.

conditioning

samples

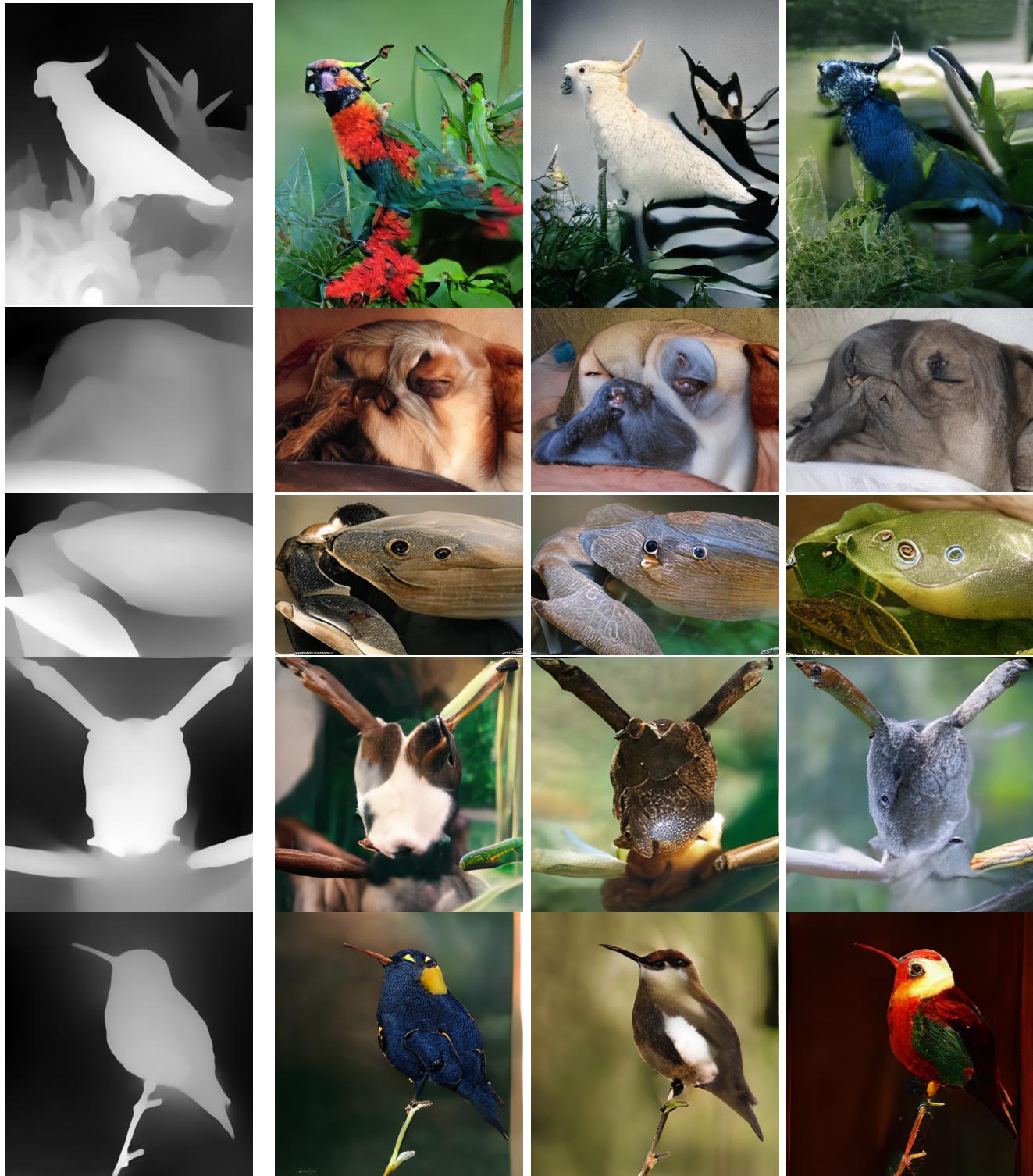
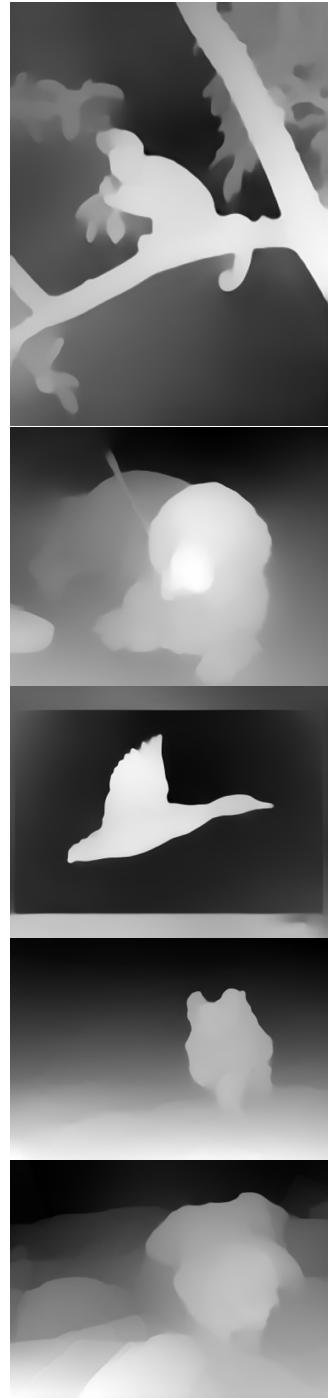


Figure 17. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.

conditioning



samples

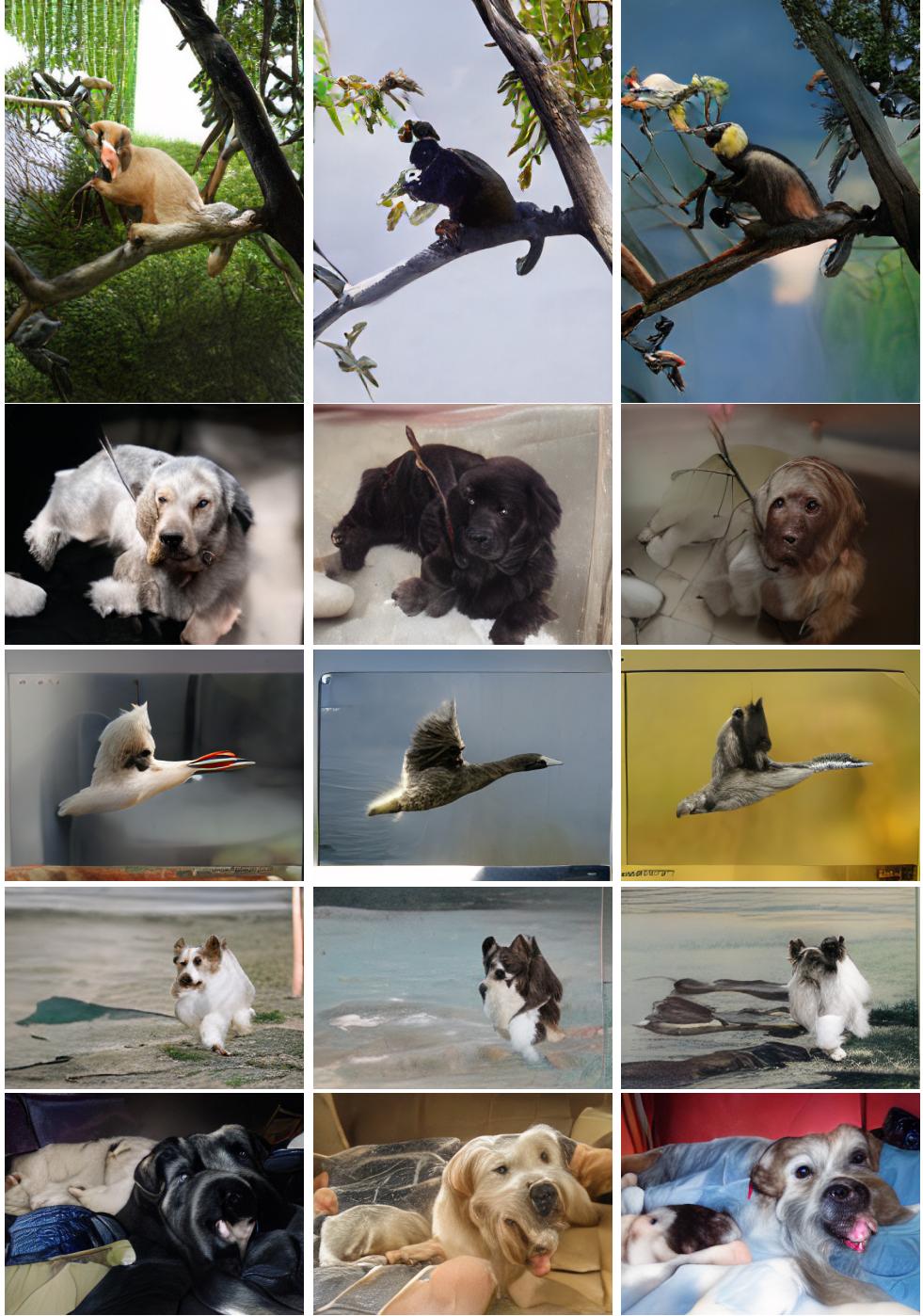
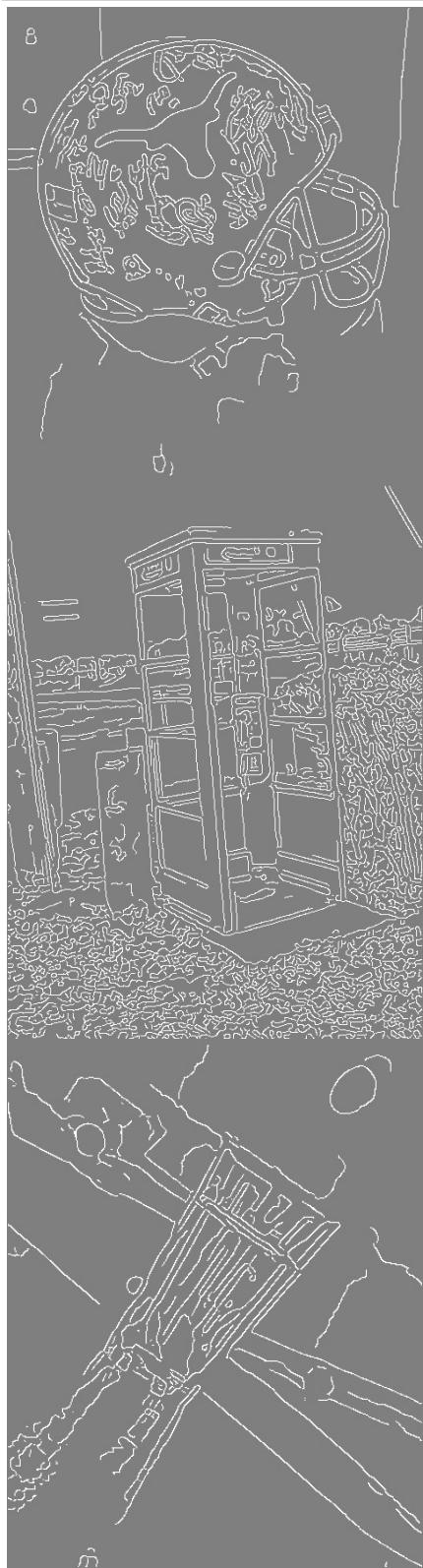


Figure 18. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.

conditioning



samples



Figure 19. Intentionally limiting the receptive field can lead to interesting creative applications like this one: Edge-to-Image synthesis on IN with $f = 8$, using the sliding attention window.



Figure 20. Additional results for stochastic superresolution with an $f = 16$ model on IN, using the sliding attention window.

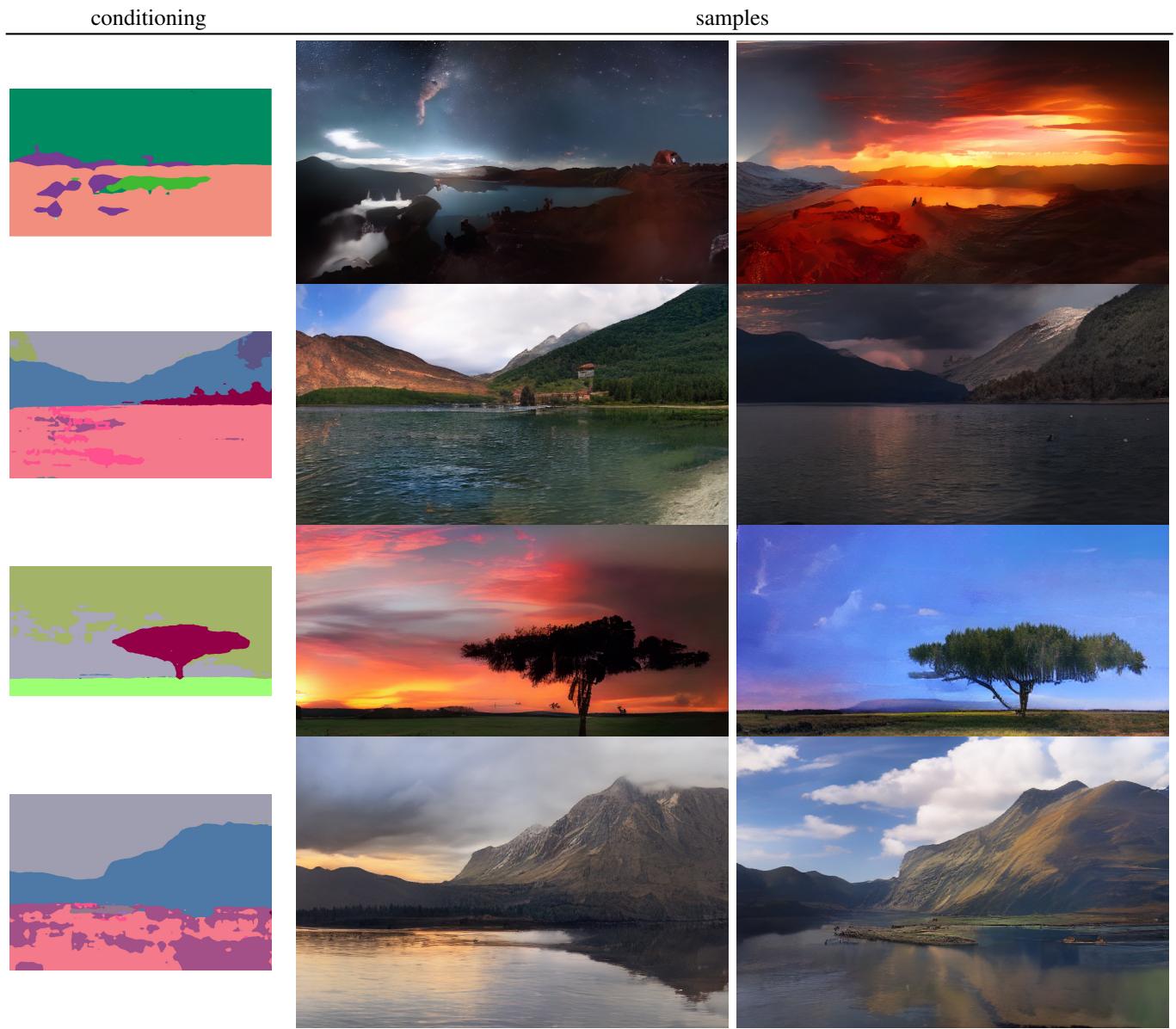


Figure 21. Samples generated from semantic layouts on S-FLCKR with $f = 16$, using the sliding attention window.

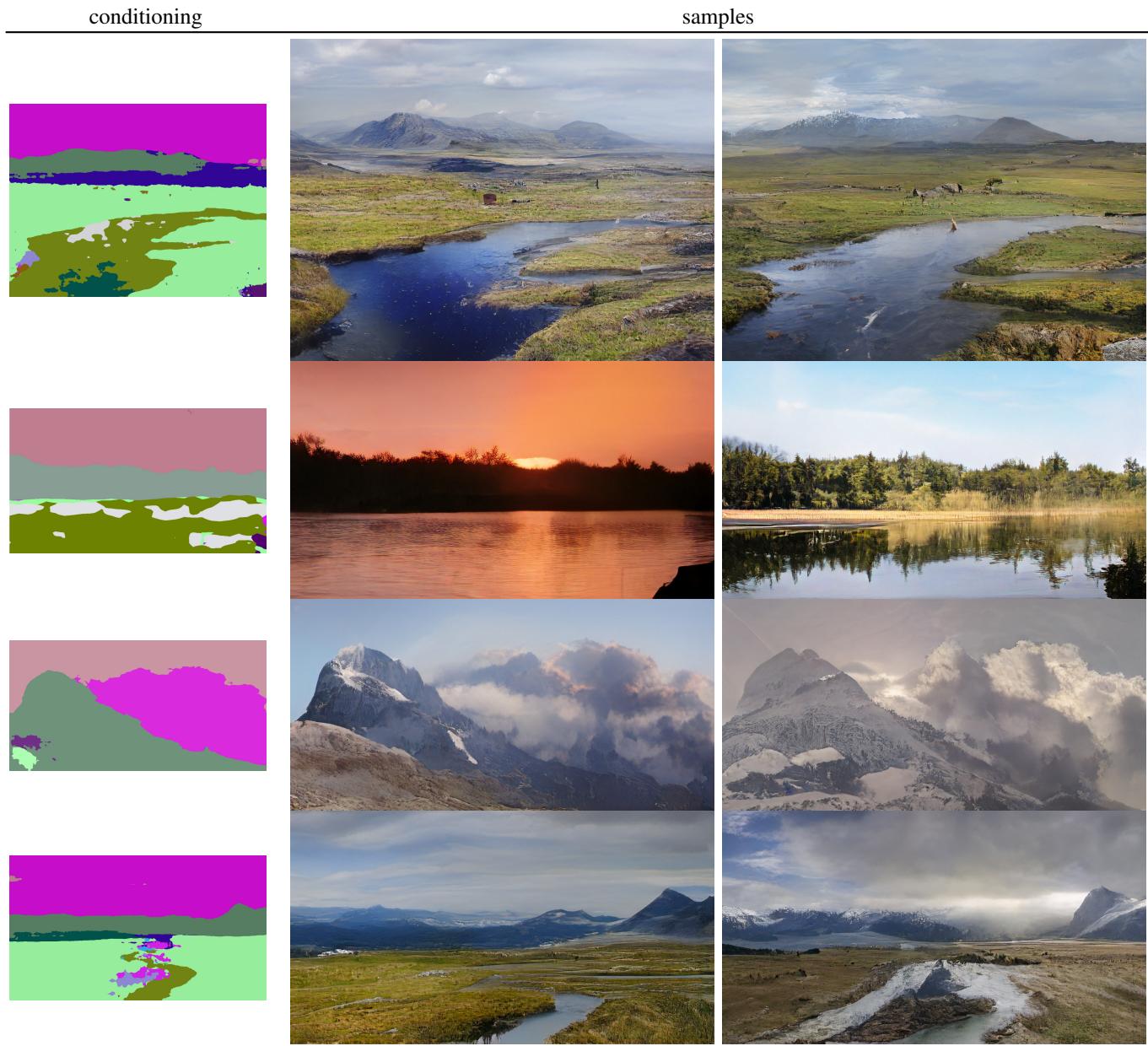


Figure 22. Samples generated from semantic layouts on S-FLCKR with $f = 32$, using the sliding attention window.



Figure 23. Unconditional samples from a model trained on LSUN Churches & Towers, using the sliding attention window.

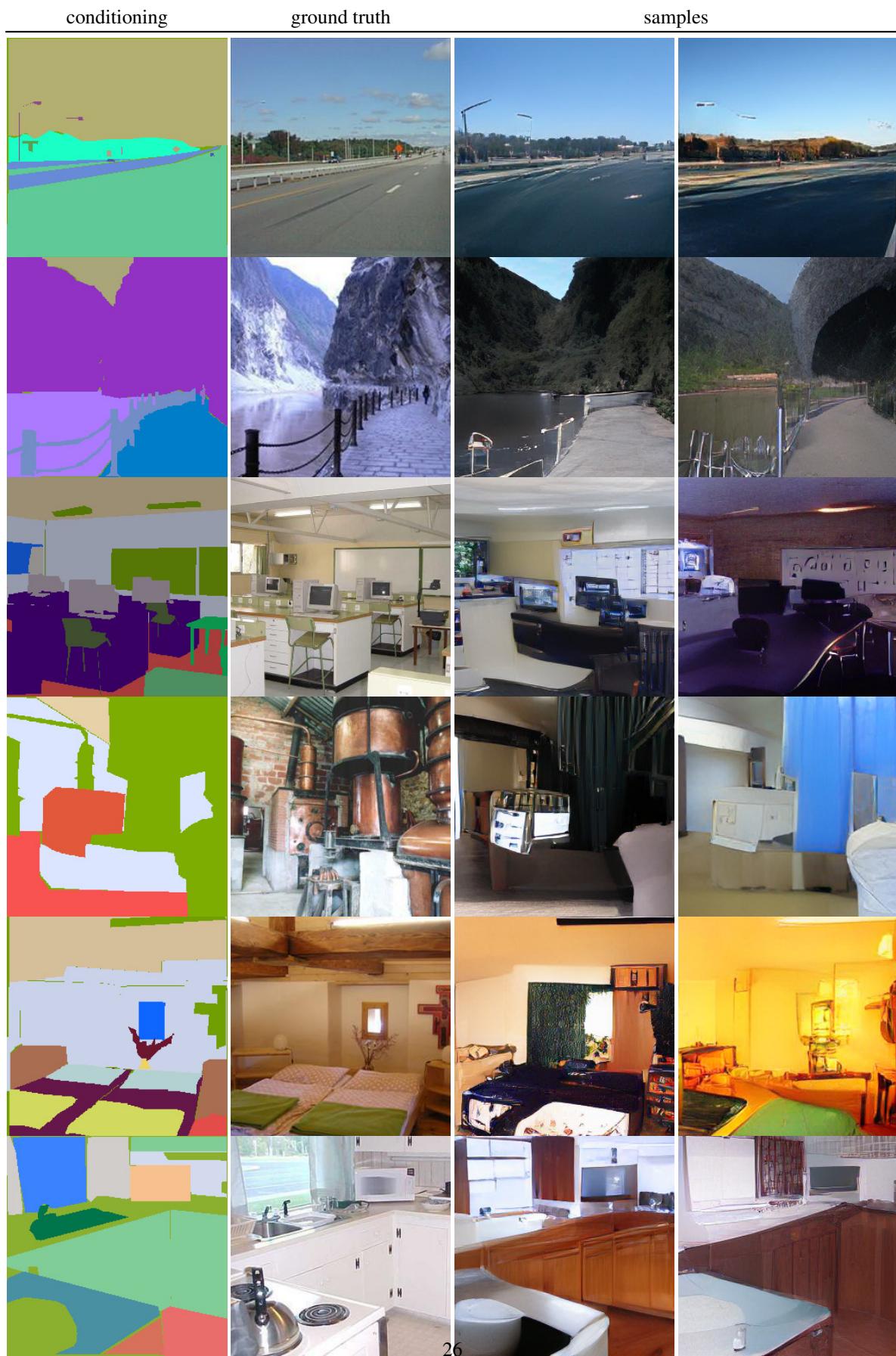


Figure 24. Additional 256×256 results on the ADE20K dataset.

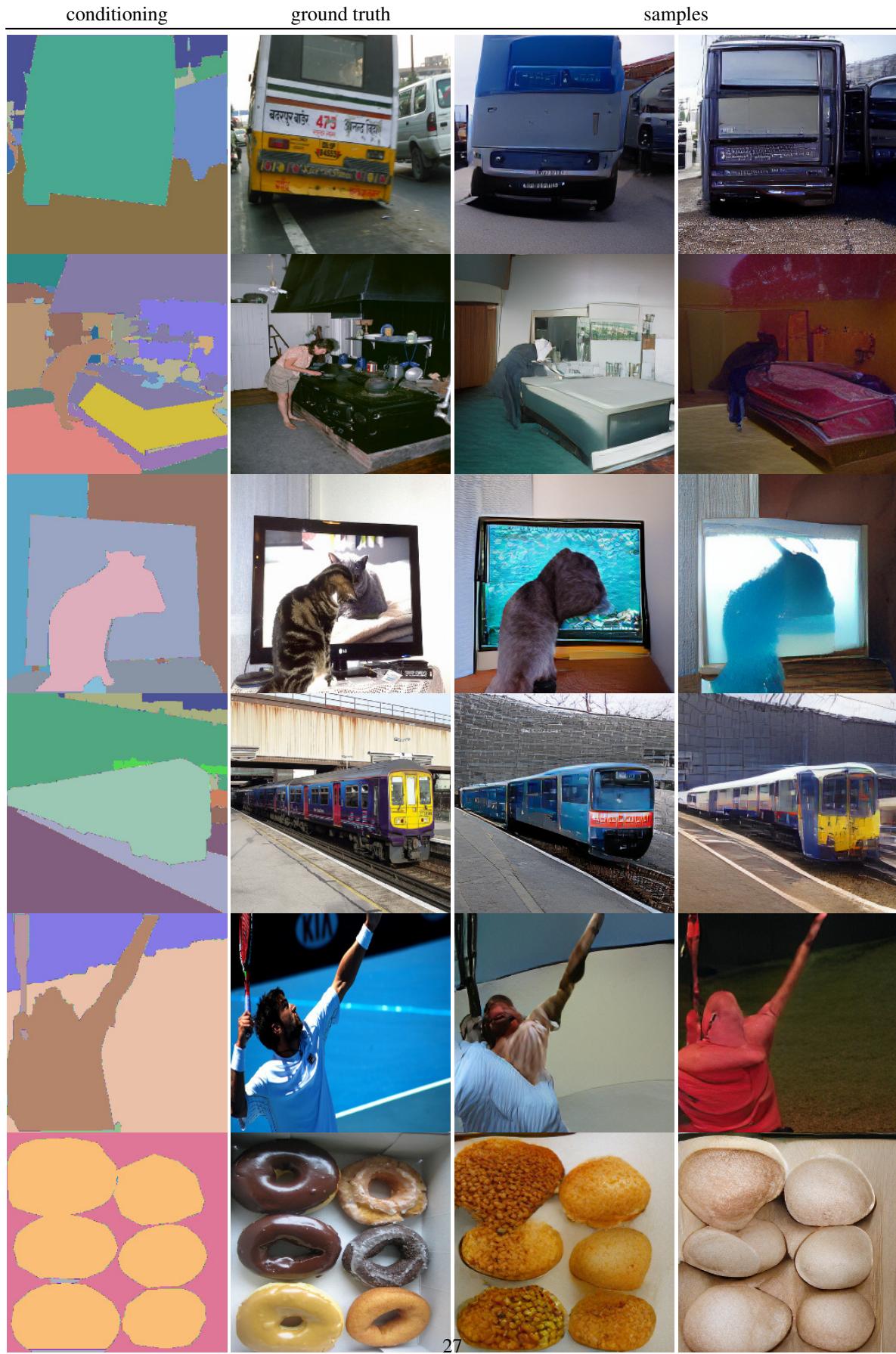


Figure 25. Additional 256 × 256 results on the COCO-Stuff dataset.

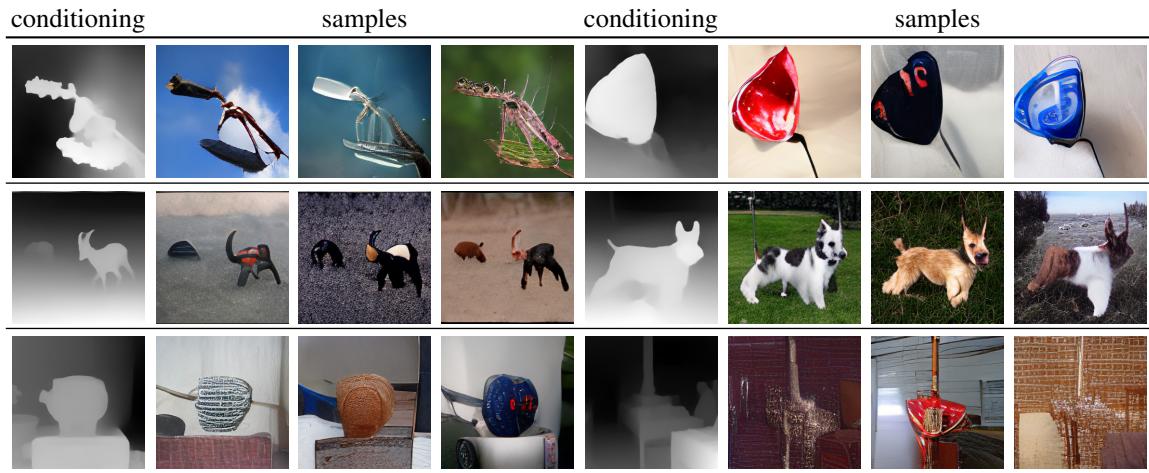


Figure 26. Conditional samples for the depth-to-image model on IN.

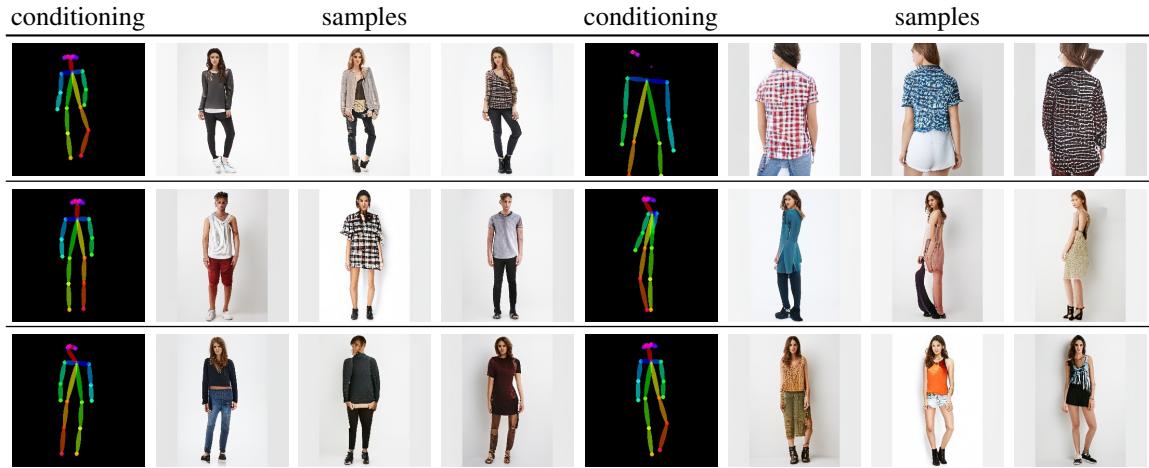


Figure 27. Conditional samples for the pose-guided synthesis model via keypoints on DeepFashion.

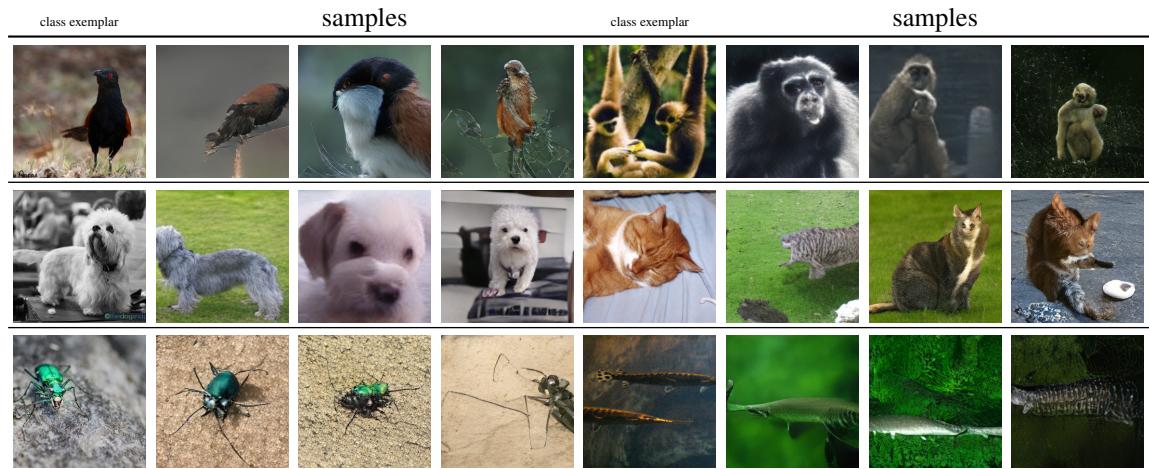


Figure 28. Samples produced by the class-conditional model trained on RIN.

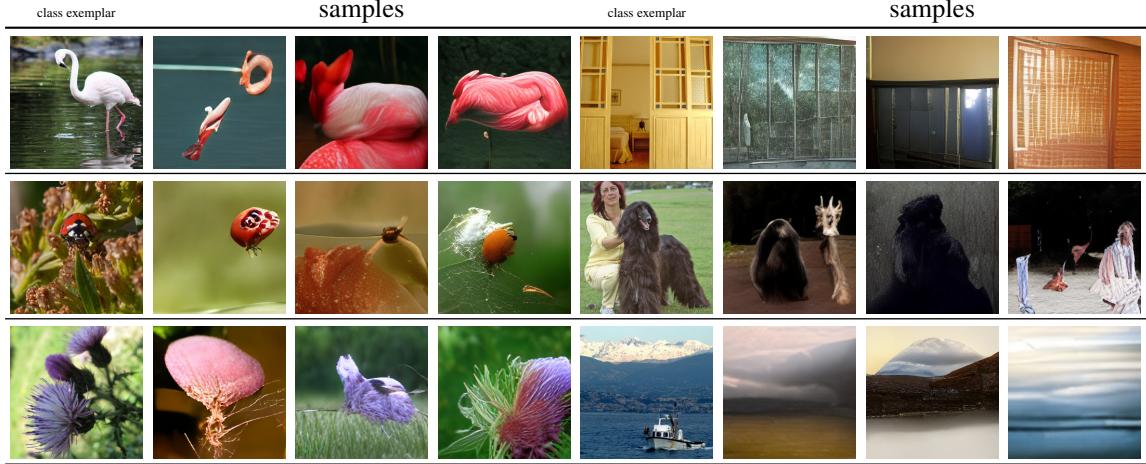


Figure 29. Samples synthesized by the class-conditional IN model.

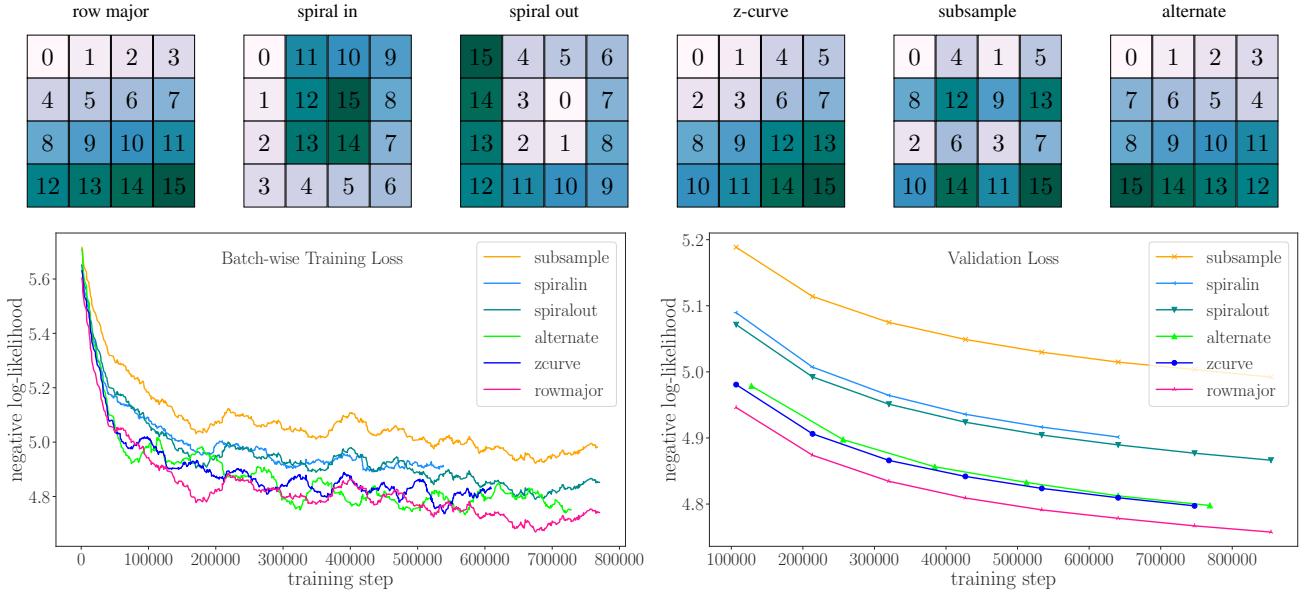


Figure 30. Top: All sequence permutations we investigate, illustrated on a 4×4 grid. Bottom: The transformer architecture is permutation invariant but next-token prediction is not: The average loss on the validation split of ImageNet, corresponding to the negative log-likelihood, differs significantly between different prediction orderings. Among our choices, the commonly used row-major order performs best.

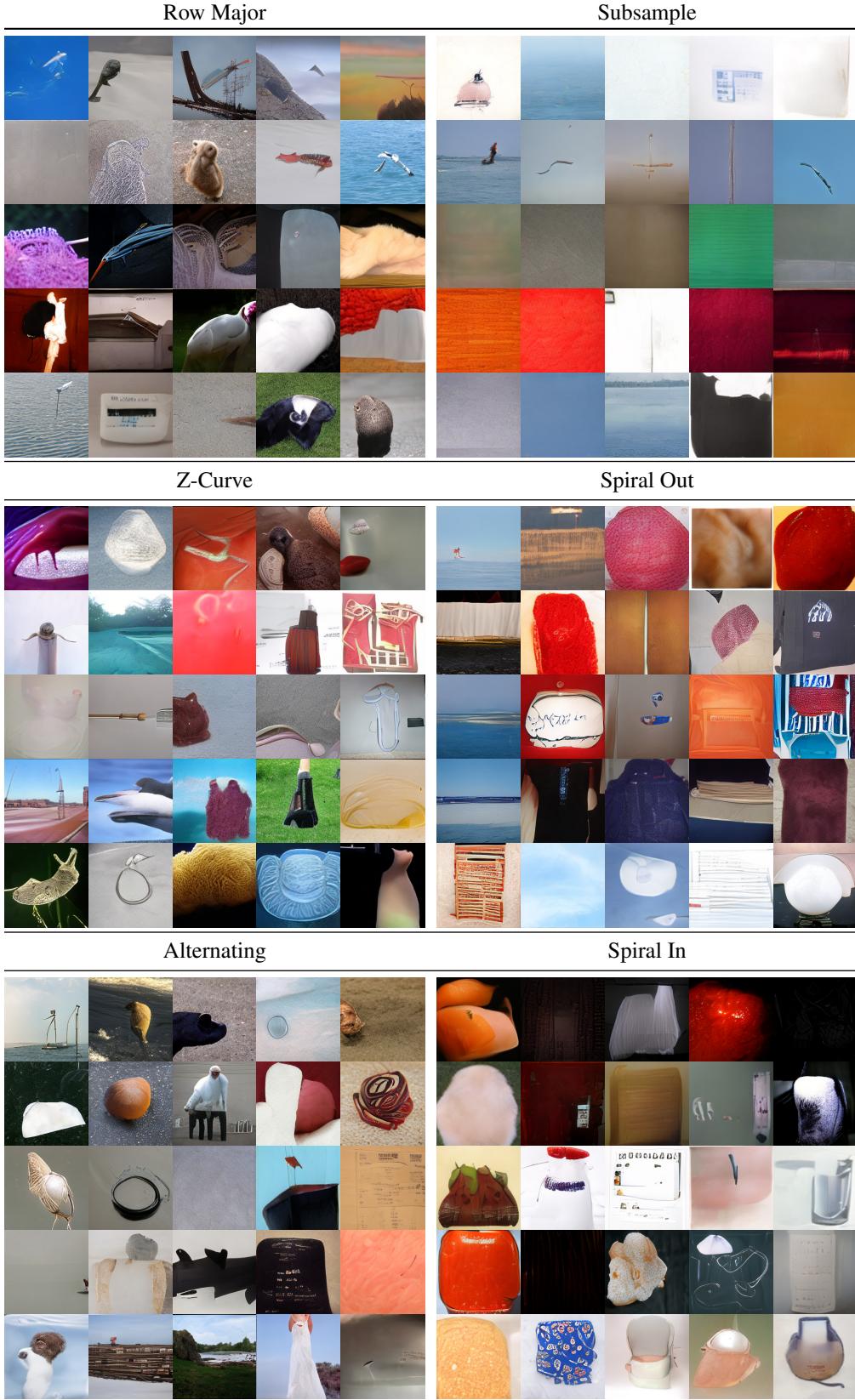


Figure 31. Random samples from transformer models trained with different orderings for autoregressive prediction as described in Sec. 4.4.

References

- [1] Anonymous. An image is worth 16x16 words: Transformers for image recognition at scale. In *Submitted to International Conference on Learning Representations*, 2021. under review. 1
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 2
- [3] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. 4
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*, 2020. 1
- [5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018. 6
- [6] Liang-Chieh Chen, G. Papandreou, I. Kokkinos, Kevin Murphy, and A. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 5
- [7] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. 2020. 1, 2, 3, 4, 5, 7, 8, 10, 13, 14
- [8] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 863–871. PMLR, 2018. 2, 5
- [9] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. 1, 2, 5
- [10] Bin Dai and David P. Wipf. Diagnosing and enhancing VAE models. In *7th International Conference on Learning Representations, ICLR*, 2019. 2
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, 2009. 5
- [12] Alexey Dosovitskiy and Thomas Brox. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2016. 4
- [13] Patrick Esser, Robin Rombach, and Björn Ommer. A Disentangling Invertible Interpretation Network for Explaining Latent Representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020. 2
- [14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2014. 3
- [15] Seungwook Han, Akash Srivastava, Cole L. Hurwitz, Prasanna Sattigeri, and David D. Cox. not-so-biggan: Generating high-fidelity images on a small compute budget. *CoRR*, abs/2009.04433, 2020. 3
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 9
- [17] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *CoRR*, abs/1912.12180, 2019. 2, 5
- [18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017. 4, 9
- [19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV (2)*, volume 9906 of *Lecture Notes in Computer Science*, pages 694–711. Springer, 2016. 4
- [20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2018. 7
- [21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 7
- [22] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks, 2017. 2
- [23] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014. 2
- [24] Alex Lamb, Vincent Dumoulin, and Aaron C. Courville. Discriminative regularization for generative models. *CoRR*, abs/1602.03220, 2016. 4
- [25] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2015. 4
- [26] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *AAAI*, pages 6706–6713. AAAI Press, 2019. 2
- [27] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. COCO-GAN: generation by parts via conditional coordinating. In *ICCV*, pages 4511–4520. IEEE, 2019. 5

- [28] Jinlin Liu, Yuan Yao, and Jianqiang Ren. An acceleration framework for high resolution image synthesis. *CoRR*, abs/1909.03611, 2019. [2](#)
- [29] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *ICLR (Poster)*. OpenReview.net, 2018. [4](#)
- [30] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [6](#)
- [31] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual objects. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [2](#)
- [32] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference, 2016. [2](#)
- [33] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019. [2](#)
- [34] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 4052–4061. PMLR, 2018. [2, 3, 5](#)
- [35] A. Radford. Improving language understanding by generative pre-training. 2018. [1](#)
- [36] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. [1, 5, 9](#)
- [37] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. [5](#)
- [38] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. [3, 4, 5](#)
- [39] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning, ICML*, 2014. [2](#)
- [40] Robin Rombach, Patrick Esser, and Björn Ommer. Making sense of cnns: Interpreting deep representations & their invariances with inns. *CoRR*, abs/2008.01777, 2020. [2](#)
- [41] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Computer vision with a single (robust) classifier. In *ArXiv preprint arXiv:1906.09453*, 2019. [5](#)
- [42] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019. [2](#)
- [43] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1747–1756. JMLR.org, 2016. [2](#)
- [44] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016. [2, 4, 8](#)
- [45] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. [3, 4, 10, 12](#)
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2017. [1, 2, 3](#)
- [47] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*. OpenReview.net, 2020. [2](#)
- [48] Zhisheng Xiao, Qing Yan, Yi-an Chen, and Yali Amit. Generative latent flow: A framework for non-adversarial image generation. *CoRR*, abs/1905.10485, 2019. [2](#)
- [49] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. [5](#)
- [50] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-Domain Correspondence Learning for Exemplar-Based Image Translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020. [2](#)
- [51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. [4, 9](#)
- [52] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016. [6](#)
- [53] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow, 2017. [2](#)
- [54] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization, 2019. [2](#)