

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



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 and obtain the state of the art among autoregressive models on class-conditional ImageNet. Code and pretrained models can be found at <https://git.io/JnyvK>.

1. Introduction

Transformers are on the rise—they are now the de-facto standard architecture for language tasks [74, 57, 58, 5]

and are increasingly adapted in other areas such as audio [12] and vision [8, 16]. 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 [16] 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 [51]*: 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 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 [74] is that it models interactions between its inputs solely through attention [2, 36, 52] 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 [41] 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

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

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. [55] and [76] restrict the receptive fields of the attention modules, which reduces the expressivity and, especially for high-resolution images, introduces assumptions on the independence of pixels. [12] and [26] 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 [53, 80, 68, 85, 84] suggest that this bias is often too restrictive.

Convolutional architectures have been used for autoregressive modeling of images [70, 71, 10] but, for low-resolution images, previous works [55, 12, 26] 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) [38, 62], and then again learning its distribution with a VAE. [18, 78] demonstrate similar gains when using an unconditional normalizing flow for the second stage, and [63, 64] when using a conditional normalizing flow. To improve training efficiency of Generative Adversarial Networks (GANs), [43] learns a GAN [20] on representations of an autoencoder and [21] on low-resolution wavelet coefficients which are then

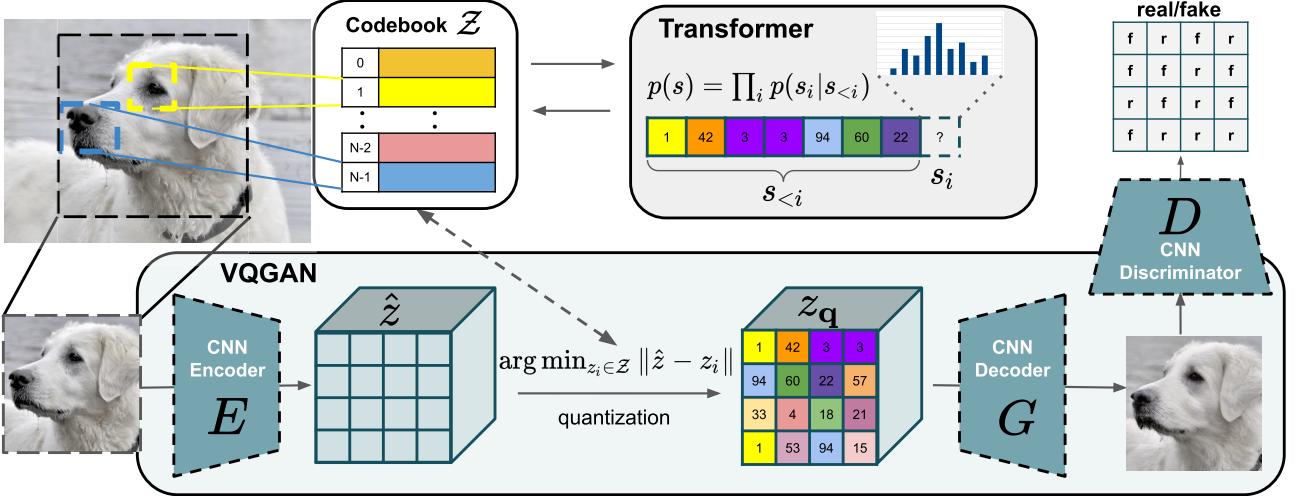


Figure 2. Our approach uses a convolutional *VQGAN* to learn a codebook of context-rich visual parts, whose composition is subsequently modeled with an autoregressive transformer architecture. A discrete codebook provides the interface between these architectures and a patch-based discriminator enables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

decoded to images with a learned generator.

[72] presents the Vector Quantised Variational Autoencoder (VQVAE), an approach to learn discrete representations of images, and models their distribution autoregressively with a convolutional architecture. [61] 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. [8] 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 [8], 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 [74] and introduce them to high-resolution image synthesis up to the megapixel range. Previous work [55, 8] 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 [72]. First, we learn a convolutional model consisting of an encoder E and a decoder G , such that taken together, they learn to represent images with codes from a learned, discrete codebook $Z = \{z_k\}_{k=1}^K \subset \mathbb{R}^{n_z}$ (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

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

Here, $\mathcal{L}_{\text{rec}} = \|x - \hat{x}\|^2$ is a reconstruction loss, $\text{sg}[\cdot]$ denotes the stop-gradient operation, and $\|\text{sg}[z_{\mathbf{q}}] - E(x)\|_2^2$ is the so-called “commitment loss” [72].

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 *VQ-GAN*, a variant of the original VQVAE, and use a discriminator and perceptual loss [40, 30, 39, 17, 47] to keep good perceptual quality at increased compression rate. Note that this is in contrast to previous works which applied pixel-based [71, 61] and transformer-based autoregressive models [8] on top of only a shallow quantization model. More specifically, we replace the L_2 loss used in [72] for \mathcal{L}_{rec} by a perceptual loss and introduce an adversarial training procedure with a patch-based discriminator D [28] 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

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

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 [81], $\nabla_{G_L} [\cdot]$ 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 . 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 [44].

Generating High-Resolution Images The attention mechanism of the transformer puts limits on the sequence

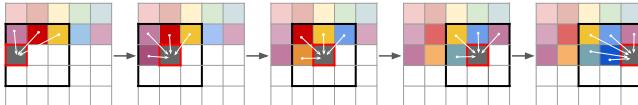


Figure 3. Sliding attention window.

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 [42].

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 $|\mathcal{Z}| = 1024$ 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) [58] on a GPU with 12GB VRAM. More details on architectures and hyperparameters can be found in the appendix (Tab. 7 and Tab. 8).

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 [55, 12, 26], 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 *VQGAN* with $m = 4$ downsampling blocks, and, if needed, another one for the

Data / # params	Negative Log-Likelihood (NLL)		
	Transformer P-SNAIL steps	Transformer P-SNAIL time	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.

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 [61]. 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) [14], *Restricted ImageNet* (RIN) [65], consisting of a subset of animal classes from ImageNet, *LSUN Churches and Towers* (LSUN-CT) [79], and for conditional image modeling of RIN conditioned on depth maps obtained with the approach of [60] (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 shows 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 (i.e. image size 256×256 , latent size 16×16), we perform various conditional image synthesis experiments:

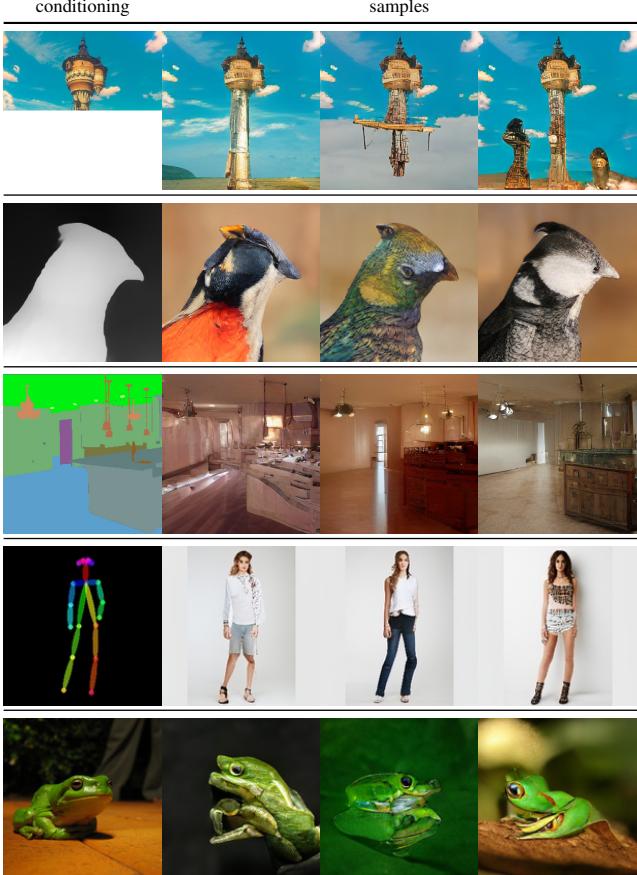


Figure 4. Transformers within our setting unify a wide range of image synthesis tasks. We show 256×256 synthesis results across different conditioning inputs and datasets, all obtained with the same approach to exploit inductive biases of effective CNN based *VQGAN* architectures in combination with 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 ADE20K. 4th row: Pose-guided person generation on DeepFashion. Bottom row: Class-conditional samples on RIN.

(i): **Semantic image synthesis**, where we condition on semantic segmentation masks of ADE20K [83], 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 [45] 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 for the RIN and IN dataset are demonstrated in Fig. 4 and Fig. 8, respectively.

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. D.

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. 29-39). 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 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 16×16 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* [31] and

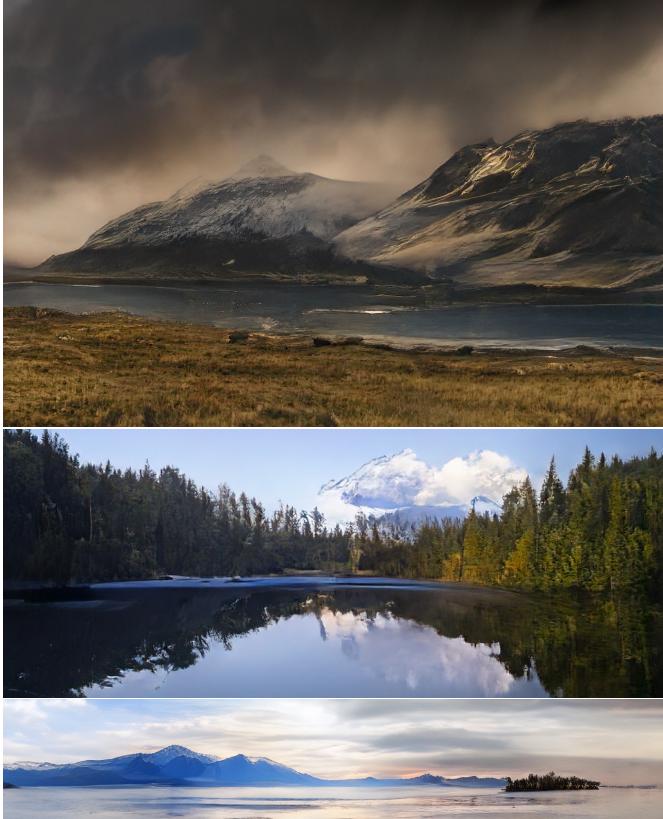


Figure 5. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832 , 1024×416 and 1280×240 pixels. Best viewed zoomed in. A larger visualization can be found in the appendix, see Fig 29.

FFHQ [33]. 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. 13 and Sec. C).

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 $14.08\times$ faster sampling of images.

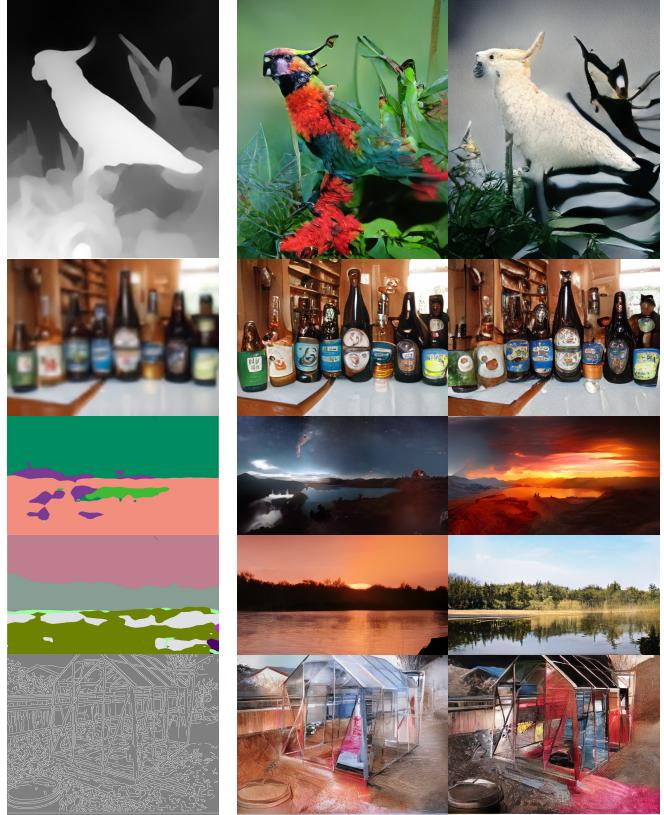


Figure 6. Applying the sliding attention window approach (Fig. 3) to various conditional image synthesis tasks. Top: Depth-to-image on RIN, 2nd row: Stochastic superresolution on IN, 3rd and 4th row: Semantic synthesis on S-FLCKR, bottom: Edge-guided synthesis on IN. The resulting images vary between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

Dataset	ours	SPADE [53]	Pix2PixHD (+aug) [75]	CRN [9]
COCO-Stuff	22.4	22.6/23.9(*)	111.5 (54.2)	70.4
ADE20K	35.5	33.9/35.7(*)	81.8 (41.5)	73.3

Table 2. FID score comparison for semantic image synthesis (256×256 pixels). (*): Recalculated with our evaluation protocol based on [50] on the validation splits of each dataset.

4.4. Benchmarking Image Synthesis Results

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). The results on *semantic synthesis* are shown in Tab. 2, where we compare to [53, 75, 35, 9], and the results on *unconditional face synthesis* are shown in Tab. 3. 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.

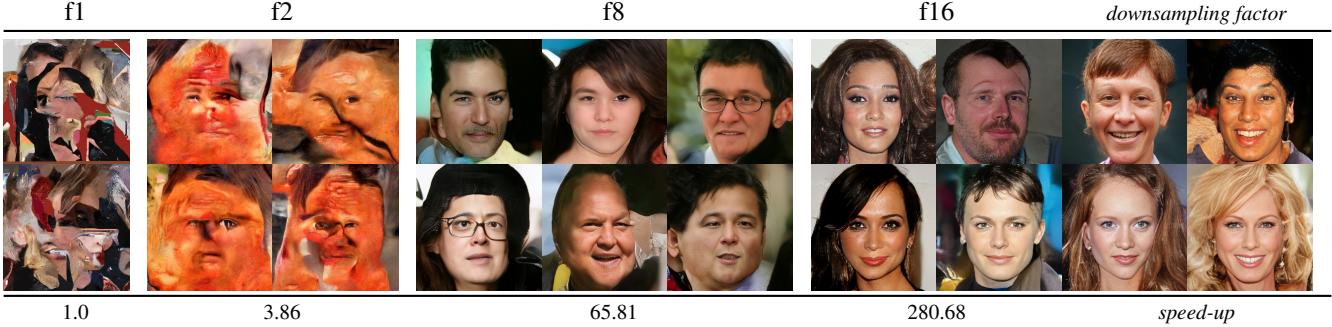


Figure 7. Evaluating the importance of effective codebook for HQ-Faces (CelebA-HQ and FFHQ) for a fixed sequence length $|s|=16 \cdot 16 = 256$. Globally consistent structures can only be modeled with a context-rich vocabulary (right). All samples are generated with temperature $t=1.0$ and top- k sampling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

CelebA-HQ 256 × 256		FFHQ 256 × 256	
Method	FID ↓	Method	FID ↓
GLOW [37]	69.0	VDVAE ($t=0.7$) [11]	38.8
NVAE [69]	40.3	VDVAE ($t=1.0$)	33.5
PIONEER (B.) [23]	39.2 (25.3)	VDVAE ($t=0.8$)	29.8
NCPVAE [1]	24.8	VDVAE ($t=0.9$)	28.5
VAEBM [77]	20.4	VQGAN+PSNAIL	21.9
Style ALAE [56]	19.2	BigGAN	12.4
DC-VAE [54]	15.8	ours ($k=300$)	9.6
ours ($k=400$)	10.2	U-Net GAN (+aug) [66]	10.9 (7.6)
PGGAN [31]	8.0	StyleGAN2 (+aug) [34]	3.8 (3.6)

Table 3. FID score comparison for face image synthesis. CelebA-HQ results reproduced from [1, 54, 77, 24], FFHQ from [66, 32].

Autoregressive models are typically sampled with a decoding strategy [27] such as beam-search, top-k or nucleus sampling. For most of our results, including those in Tab. 2, we use top-k sampling with $k=100$ unless stated otherwise. For the results on face synthesis in Tab. 3, we computed scores for $k \in \{100, 200, 300, 400, 500\}$ and report the best results, obtained with $k=400$ for CelebA-HQ and $k=300$ for FFHQ. Fig. 10 in the supplementary shows FID and Inception scores as a function of k .

Class-Conditional Synthesis on ImageNet To address a direct comparison with the previous state-of-the-art for autoregressive modeling of class-conditional image synthesis on ImageNet, VQVAE-2 [61], we train a class-conditional ImageNet transformer on 256×256 images, using a *VQGAN* with $\dim \mathcal{Z} = 16384$ and $f=16$, and additionally compare to BigGAN [4], IDDPMP [49], DCTransformer [48] and ADM [15] in Tab. 4. Note that our model uses $\simeq 10 \times$ less parameters than VQVAE-2, which has an estimated parameter count of 13.5B (estimate based on [67]).

Samples of this model for different ImageNet classes are shown in Fig. 8. We observe that the adversarial training of the corresponding *VQGAN* enables sampling of high-quality images with realistic textures, of comparable or higher quality than existing approaches such as BigGAN and VQVAE-2, see also Fig. 14–17 in the supplementary.

Model	acceptance rate	FID	IS
mixed $k, p=1.0$	1.0	17.04	70.6 ± 1.8
$k=973, p=1.0$	1.0	29.20	47.3 ± 1.3
$k=250, p=1.0$	1.0	15.98	78.6 ± 1.1
$k=973, p=0.88$	1.0	15.78	74.3 ± 1.8
$k=600, p=1.0$	0.05	5.20	280.3 ± 5.5
mixed $k, p=1.0$	0.5	10.26	125.5 ± 2.4
mixed $k, p=1.0$	0.25	7.35	188.6 ± 3.3
mixed $k, p=1.0$	0.05	5.88	304.8 ± 3.6
mixed $k, p=1.0$	0.005	6.59	402.7 ± 2.9
DCTransformer [48]	1.0	36.5	n/a
VQVAE-2 [61]	1.0	~31	~45
VQVAE-2	n/a	~10	~330
BigGAN [4]	1.0	7.53	168.6 ± 2.5
BigGAN-deep	1.0	6.84	203.6 ± 2.6
IDDPMP [49]	1.0	12.3	n/a
ADM-G, no guid. [15]	1.0	10.94	100.98
ADM-G, 1.0 guid.	1.0	4.59	186.7
ADM-G, 10.0 guid.	1.0	9.11	283.92
val. data	1.0	1.62	234.0 ± 3.9

Table 4. FID score comparison for class-conditional synthesis on 256×256 ImageNet, evaluated between 50k samples and the training split. Classifier-based rejection sampling as in VQVAE-2 uses a ResNet-101 [22] classifier. BigGAN(-deep) evaluated via <https://tfhub.dev/deepmind> truncated at 1.0. “Mixed” k refers to samples generated with different top-k values, here $k \in \{100, 200, 250, 300, 350, 400, 500, 600, 800, 973\}$.

Quantitative results are summarized in Tab. 4. We report FID and Inception Scores for the best k/p in top-k/top-p sampling. Following [61], we can further increase quality via classifier-rejection, which keeps only the best m -out-of- n samples in terms of the classifier’s score, i.e. with an acceptance rate of m/n . We use a ResNet-101 classifier [22].

We observe that our model outperforms other autoregressive approaches (VQVAE-2, DCTransformer) in terms of FID and IS, surpasses BigGAN and IDDPMP even for low rejection rates and yields scores close to the state of the art for higher rejection rates, see also Fig. 9.

How good is the *VQGAN*? Reconstruction FIDs obtained via the codebook provide an estimate on the achievable FID of the generative model trained on it. To quantify the per-

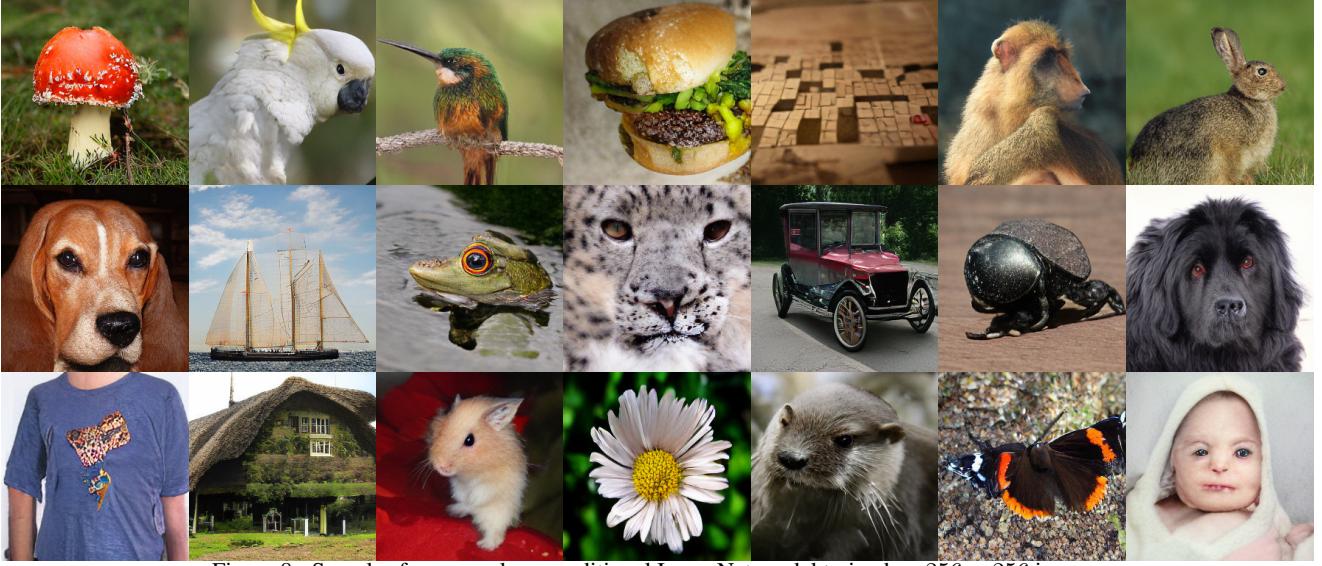


Figure 8. Samples from our class-conditional ImageNet model trained on 256×256 images.

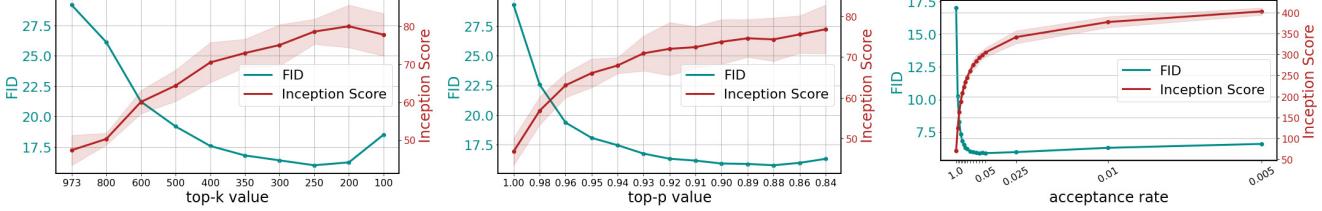


Figure 9. FID and Inception Score as a function of top-k, nucleus and rejection filtering.

Model	Codebook Size	dim \mathcal{Z}	FID/val	FID/train
VQVAE-2	64×64 & 32×32	512	n/a	~ 10
DALL-E [59]	32×32	8192	32.01	33.88
<i>VQGAN</i>	16×16	1024	7.94	10.54
<i>VQGAN</i>	16×16	16384	4.98	7.41
<i>VQGAN*</i>	32×32	8192	1.49	3.24
<i>VQGAN</i>	64×64 & 32×32	512	1.45	2.78

Table 5. FID on ImageNet between reconstructed validation split and original validation (FID/val) and training (FID/train) splits.

*trained with Gumbel-Softmax reparameterization as in [59, 29].

formance gains of our *VQGAN* over discrete VAEs trained without perceptual and adversarial losses (*e.g.* VQVAE-2, DALL-E [59]), we evaluate this metric on ImageNet and report results in Tab. 5. Our *VQGAN* outperforms non-adversarial models while providing significantly more compression (seq. length of 256 vs. $5120 = 32^2 + 64^2$ for VQVAE-2, 256 vs 1024 for DALL-E). 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. The qualitative comparison corresponding to the results in Tab. 5 can be found in Fig. 12.

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.

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, Sec. A summarizes changes to a previous version of this paper. In Sec. B, we present hyperparameters and architectures which were used to train our models. Next, extending the discussion of Sec. 4.3, Sec. C 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. D, and Sec. E shows nearest neighbors of samples. Finally, Sec. F contains results regarding the ordering of image representations.

A. Changelog

We summarize changes between this version ¹ of the paper and its previous version ².

In the previous version, Eq. (4) had a weighting term β on the commitment loss, and Tab. 8 reported a value of $\beta = 0.25$ for all models. However, due to a bug in the implementation, β was never used and all models have been trained with $\beta = 1.0$. Thus, we removed β in Eq. (4).

We updated class-conditional synthesis results on ImageNet in Sec. 4.4. The previous results, included here in Tab. 6 for completeness, were based on a slightly different implementation where the transformer did not predict the distribution of the first token but used a histogram for it. The new model has been trained for 2.4 million steps with a batch size of 16 accumulated over 8 batches, which took 45.8 days on a single A100 GPU. The previous model had been trained for 1.0 million steps. Furthermore, the FID values were based on 50k (18k) samples against 50k (18k) training examples (to compare with MSP). For better comparison with other works, the current version reports FIDs based on 50k samples against all training examples of ImageNet using `torch-fidelity` [50]. We updated all qualitative figures showing samples from this model and added visualizations of the effect of tuning top- k/p or rejection rate in Fig. 14-26.

To provide a better overview, we also include results from works that became available after the previous version of our work. Specifically, we include results on reconstruction quality of the VQVAE from [59] in Tab. 5 and Fig. 12 (which replaces the previous qualitative comparison), and results on class-conditional ImageNet sampling from [49, 48, 15] in Tab. 4. Note that with the exception of BigGAN and BigGAN-deep [4], no models or sampling results are available for the methods we compare to in Tab. 4. Thus, we can only report the numbers from the respective papers but cannot re-evaluate them with the same code. We follow the common evaluation protocol for class-conditional ImageNet synthesis from [4] and evaluate 50k samples from the model against the whole training split of ImageNet. However, it is not clear how different implementations resize the training images. In our code, we use the largest center-crop and resize it bilinearly with anti-aliasing to 256×256 using `Pillow` [73]. FID and Inception Scores are then computed with `torch-fidelity` [50].

We updated face-synthesis results in Tab. 3 based on a slightly different implementation as in the case of class-conditional ImageNet results and improve the previous results slightly. In addition, we evaluate the ability of our NLL-based training to detect overfitting. We train larger models (FFHQ (big) and CelebA-HQ (big) in Tab. 8) on the face datasets, and show nearest neighbors of samples obtained from checkpoints with the best NLL on the validation split and the training split in Sec. E. We also added Fig. 10, which visualizes the effect of tuning k in top- k sampling on FID and IS.

B. Implementation Details

The hyperparameters for all experiments presented in the main paper and supplementary material can be found in Tab. 8. Except for the *c-IN (big)*, *COCO-Stuff* and *ADE20K* models, 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.

¹<https://arxiv.org/abs/2012.09841v3>

²<https://arxiv.org/abs/2012.09841v2>

Dataset	ours-previous (+R)	BigGAN (-deep)	MSP	Dataset	ours-previous	ours-new
IN 256, 50K	19.8 (11.2)	7.1 (7.3)	n.a.	CelebA-HQ 256	10.7	10.2
IN 256, 18K	23.5	9.6 (9.7)	50.4	FFHQ 256	11.4	9.6

Table 6. Results from a previous version of this paper, see also Sec. A. Left: Previous results on class-conditional ImageNet synthesis with a slightly different implementation and evaluated against 50k and 18k training examples instead of the whole training split. See Tab. 4 for new, improved results evaluated against the whole training split. Right: Previous results on face-synthesis with a slightly different implementation compared to the new implementation. See also Tab. 3 for comparison with other methods.

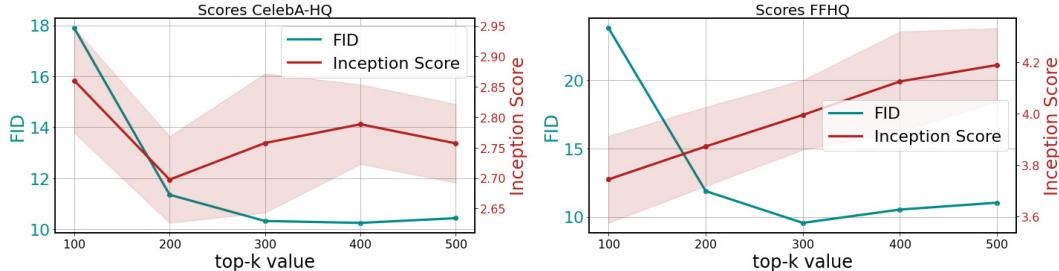


Figure 10. FID and Inception Score as a function of top-k for CelebA-HQ (left) and FFHQ (right).

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

Table 7. High-level architecture of the encoder and decoder of our *VQGAN*. The design of the networks follows the architecture presented in [25] with no skip-connections. For the discriminator, we use a patch-based model as in [28]. Note that $h = \frac{H}{2^m}$, $w = \frac{W}{2^m}$ and $f = 2^m$.

VQGAN Architecture The architecture of our convolutional encoder and decoder models used in the *VQGAN* experiments is described in Tab. 7. Note that we adopt the compression rate by tuning the number of downsampling steps m . Further note that λ in Eq. 5 is set to zero in an initial warm-up phase. Empirically, we found that longer warm-ups generally lead to better reconstructions. As a rule of thumb, we recommend setting $\lambda = 0$ for at least one epoch.

Transformer Architecture Our transformer model is identical to the GPT2 architecture [58] and we vary its capacity mainly through varying the amount of layers (see Tab. 8). Furthermore, we generally produce samples with a temperature $t = 1.0$ and a top-k cutoff at $k = 100$ (with higher top-k values for larger codebooks).

C. 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. 11. The left part shows the reconstruction error (measured by LPIPS [81]) 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.

Hence, Fig. 11 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. 12 compares its reconstruction capabilities against the VQVAE [72]

Experiment	n_{layer}	# params [M]	n_z	$ \mathcal{Z} $	dropout	length(s)	n_e	m
RIN	12	85	64	768	0.0	512	1024	4
c-RIN	18	128	64	768	0.0	257	768	4
D-RINV1	14	180	256	1024	0.0	512	768	4
D-RINV2	24	307	256	1024	0.0	512	1024	4
IN	24	307	256	1024	0.0	256	1024	4
c-IN	24	307	256	1024	0.0	257	1024	4
c-IN (big)	48	1400	256	16384	0.0	257	1536	4
IN-Edges	24	307	256	1024	0.0	512	1024	3
IN-SR	12	153	256	1024	0.0	512	1024	3
S-FLCKR, $f = 4$	24	307	256	1024	0.0	512	1024	2
S-FLCKR, $f = 16$	24	307	256	1024	0.0	512	1024	4
S-FLCKR, $f = 32$	24	307	256	1024	0.0	512	1024	5
(FacesHQ, $f = 1$) [*]	24	307	—	512	0.0	512	1024	—
FacesHQ, $f = 2$	24	307	256	1024	0.0	512	1024	1
FacesHQ, $f = 4$	24	307	256	1024	0.0	512	1024	2
FacesHQ, $f = 8$	24	307	256	1024	0.0	512	1024	3
FacesHQ ^{**} , $f = 16$	24	307	256	1024	0.0	512	1024	4
FFHQ ^{**} , $f = 16$	28	355	256	1024	0.0	256	1024	4
CelebA-HQ ^{**} , $f = 16$	28	355	256	1024	0.0	256	1024	4
FFHQ (big)	24	801	256	1024	0.0	256	1664	4
CelebA-HQ (big)	24	801	256	1024	0.0	256	1664	4
COCO-Stuff	32	651	256	8192	0.0	512	1280	4
ADE20K	28	405	256	4096	0.1	512	1024	4
DeepFashion	18	129	256	1024	0.0	340	768	4
LSUN-CT	24	307	256	1024	0.0	256	1024	4
CIFAR-10	24	307	256	1024	0.0	256	1024	1

Table 8. Hyperparameters. For every experiment, we set the number of attention heads in the transformer to $n_h = 16$. n_{layer} denotes the number of transformer blocks, # params the number of transformer parameters, n_z the dimensionality of codebook entries, $|\mathcal{Z}|$ the number of codebook entries, dropout the dropout rate for training the transformer, length(s) the total length of the sequence, n_e the embedding dimensionality and m the number of downsampling steps in the *VQGAN*. 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 [8] with $K = 512$ centroids. A prefix “c” refers to a class-conditional model. The models marked with a ‘**’ are trained on the same *VQGAN*.

used in DALL-E [59]. We observe that for $f = 8$ and 8192 codebook entries, both the VQVAE and *VQGAN* capture the global structure faithfully. However, the textures produced by the VQVAE are blurry, whereas those of the *VQGAN* are crisp and realistic looking (*e.g.* the stone texture and the fur and tail of the squirrel). When we increase the compression rate of the *VQGAN* further to $f = 16$, we see that some reconstructed parts are not perfectly aligned with the input anymore (*e.g.* the paw of the squirrel), but, especially with slightly larger codebooks, the reconstructions still look realistic. This 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. 13 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. 11. 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 $f = 32$.

D. Additional Results

Qualitative Comparisons The qualitative comparison corresponding to Tab. 4 and Tab. 6 can be found in Fig. 14, 15, 16 and 17. Since no models are available for VQVAE-2 and MSP, we extracted results directly from the supplementary³ and

³https://drive.google.com/file/d/1H2nr_Cu7OK18tRemsWn_6o5DGMNYentM/view?usp=sharing

from the provided samples⁴, respectively. For BigGAN, we produced the samples via the provided model⁵. Similarly, the qualitative comparison with the best competitor model (SPADE) for semantic synthesis on standard benchmarks (see Tab. 2) can be found in Fig. 40 (ADE20K) and Fig. 41 (COCO-Stuff)⁶.

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 [8]. 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. 27 and Fig. 28, 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 [8] are obtained from <https://openai.com/blog/image-gpt/>.

Additional High-Resolution Results Fig. 29, 30, 31 and Fig. 32 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. 33 and Fig. 34, edge-to-image on IN in Fig. 35, stochastic superresolution on IN in Fig. 36, more examples on semantically guided landscape synthesis on S-FLCKR in Fig. 37 with $f = 16$ and in Fig. 38 with $f = 32$, and unconditional image generation on LSUN-CT (see Sec. 4.1) in Fig. 39. Moreover, for images of size 256×256 , we provide results for generation from semantic layout on (i) ADE20K in Fig. 40 and (ii) COCO-Stuff in Fig. 41, depth-to-image on IN in Fig. 42, pose-guided person generation in Fig. 43 and class-conditional synthesis on RIN in Fig. 44.

E. Nearest Neighbors of Samples

One advantage of likelihood-based generative models over, *e.g.*, GANs is the ability to evaluate NLL on training data and validation data to detect overfitting. To test this, we trained large models for face synthesis, which can easily overfit them, and retained two checkpoints on each dataset: One for the best validation NLL (at the 10th and 13th epoch for FFHQ and CelebA-HQ, respectively), and another for the best training NLL (at epoch 1000). We then produced samples from both checkpoints and retrieved nearest neighbors from the training data based on the LPIPS similarity metric [81]. The results are shown in Fig. 45, where it can be observed that the checkpoints with best training NLL (best train NLL) reproduce the training examples, whereas samples from the checkpoints with best validation NLL (best val. NLL) depict new faces which are not found in the training data.

Based on these results, we can conclude that early-stopping based on validation NLL can prevent overfitting. Furthermore, the bottleneck for our approach on face synthesis is given by the dataset size since it has the capacity to almost perfectly fit the training data. Unfortunately, FID scores cannot detect such an overfitting. Indeed, the best train NLL checkpoints achieve FID scores of 3.86 on CelebA-HQ and 2.68 on FFHQ, compared to 10.2 and 9.6 for the best val. NLL checkpoints. While validation NLL provides a way to detect overfitting for likelihood-based models, it is not clear if early-stopping based on it is optimal if one is mainly interested in the quality of samples. To address this and the evaluation of GANs, new metrics will be required which can differentiate between models that produce new, high-quality samples and those that simply reproduce the training data.

Our class-conditional ImageNet model does not display overfitting according to validation NLL, and the nearest neighbors shown in Fig. 46 also provide evidence that the model produces new, high-quality samples.

F. 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

⁴<https://bit.ly/2FJkvhJ>

⁵<https://tfhub.dev/deepmind/biggan-deep-256/1>

⁶samples were reproduced with the authors’ official implementation available at <https://github.com/nvlabs/spade/>

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, see also [46]. (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 Fig. 47. 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 Fig.47 depicts 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 Fig. 48 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 [71, 8] outperforms other orderings.

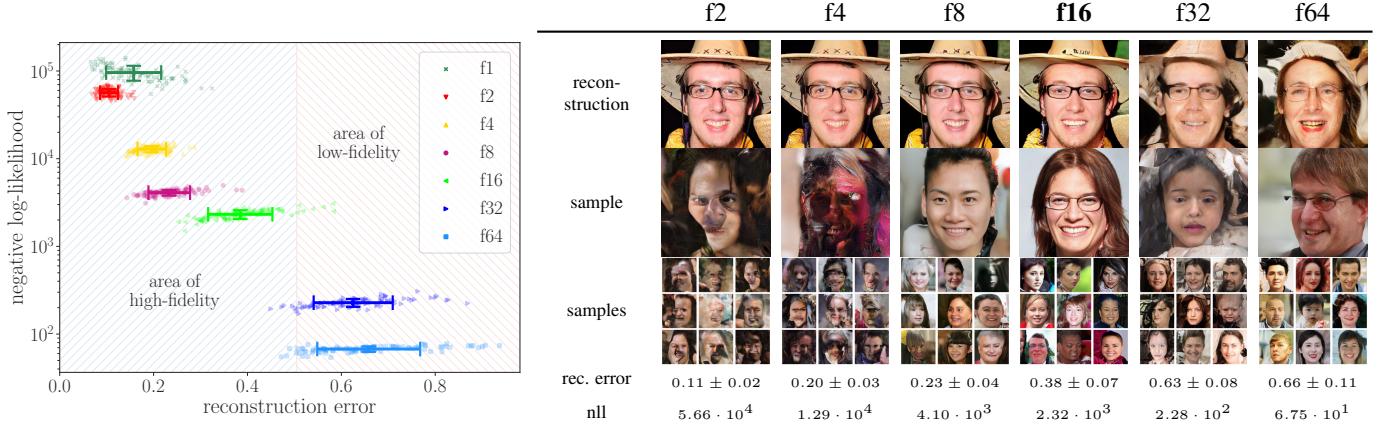


Figure 11. Trade-off between negative log-likelihood (nll) and reconstruction error. While context-rich encodings obtained with large factors f allow the transformer to effectively model long-range interactions, the reconstructions capabilities and hence quality of samples suffer after a critical value (here, $f = 16$). For more details, see Sec. C.



Figure 12. Comparing reconstruction capabilities between VQVAEs and VQGANs. Numbers in parentheses denote compression factor and codebook size. With the same compression factor and codebook size, VQGANs produce more realistic reconstructions compared to blurry reconstructions of VQVAEs. This enables increased compression rates for VQGAN while retaining realistic reconstructions. See Sec. C.

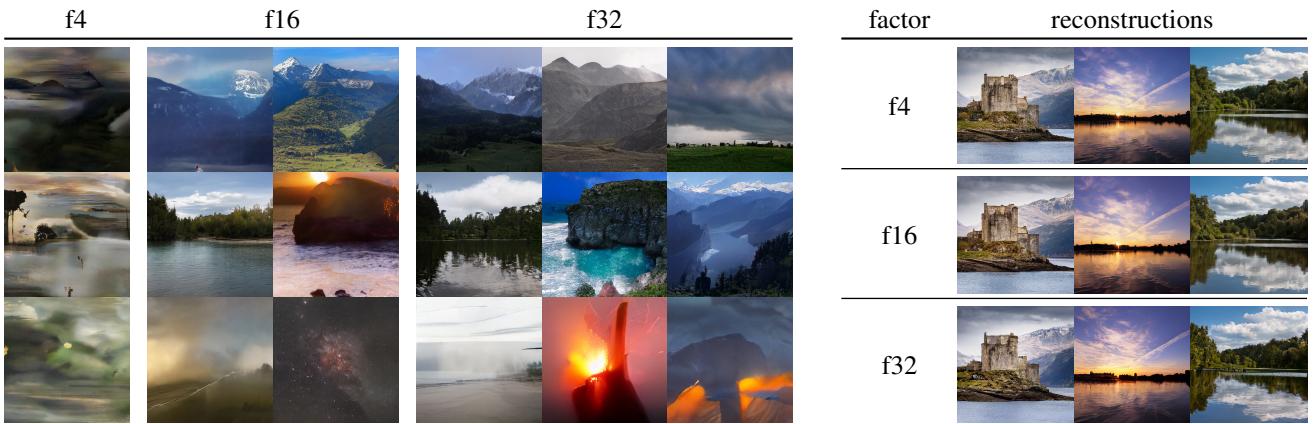


Figure 13. Samples on landscape dataset (left) obtained with different factors f , analogous to Fig. 7. In contrast to faces, a factor of $f = 32$ still allows for faithful reconstructions (right). See also Sec. C.



Figure 14. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: 28: *spotted salamander* (top) and 97: *drake* (bottom). We report class labels as in VQVAE-2 [61].

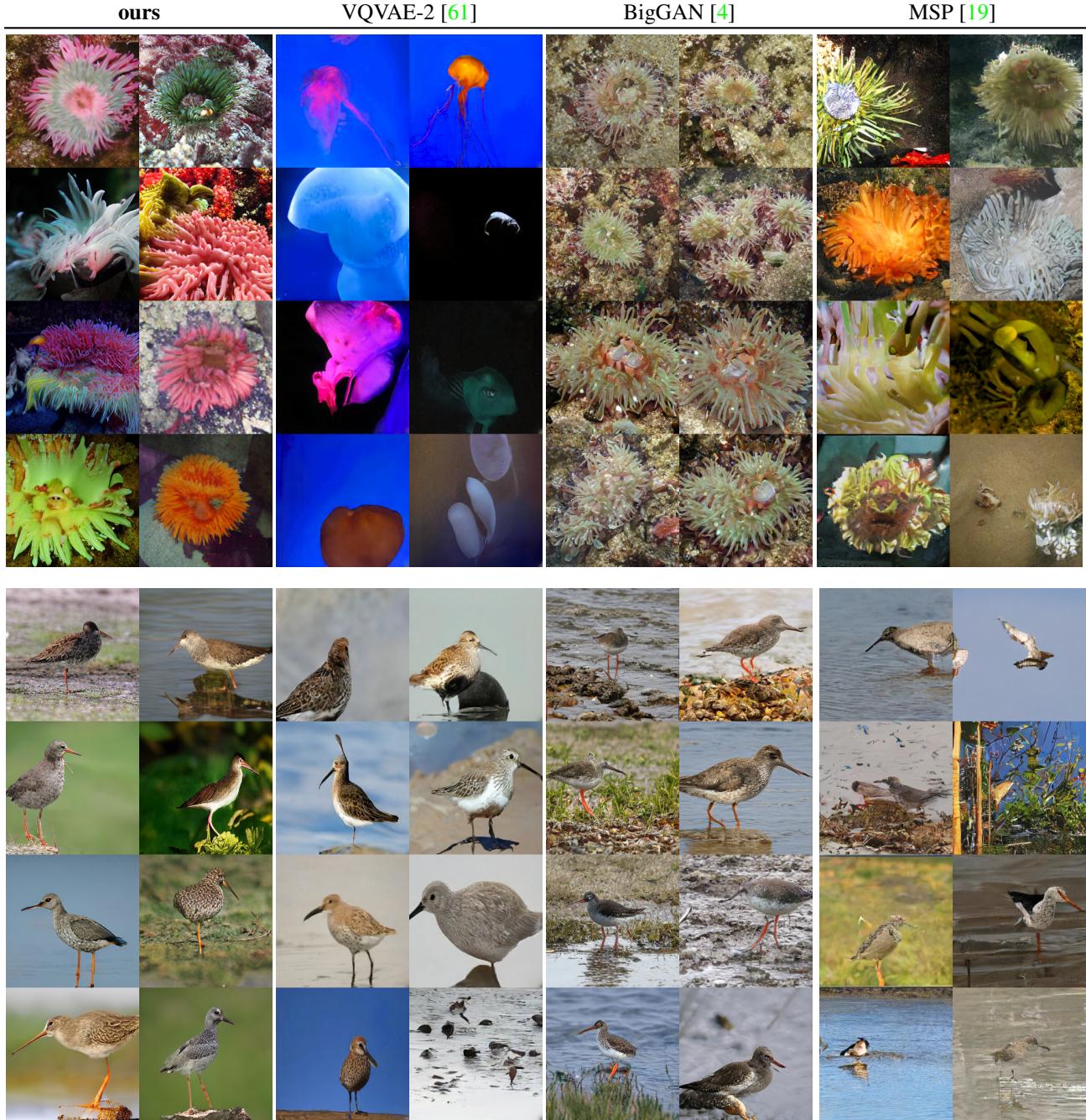


Figure 15. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: 108: sea anemone (top) and 141: redshank (bottom). We report class labels as in VQVAE-2 [61].

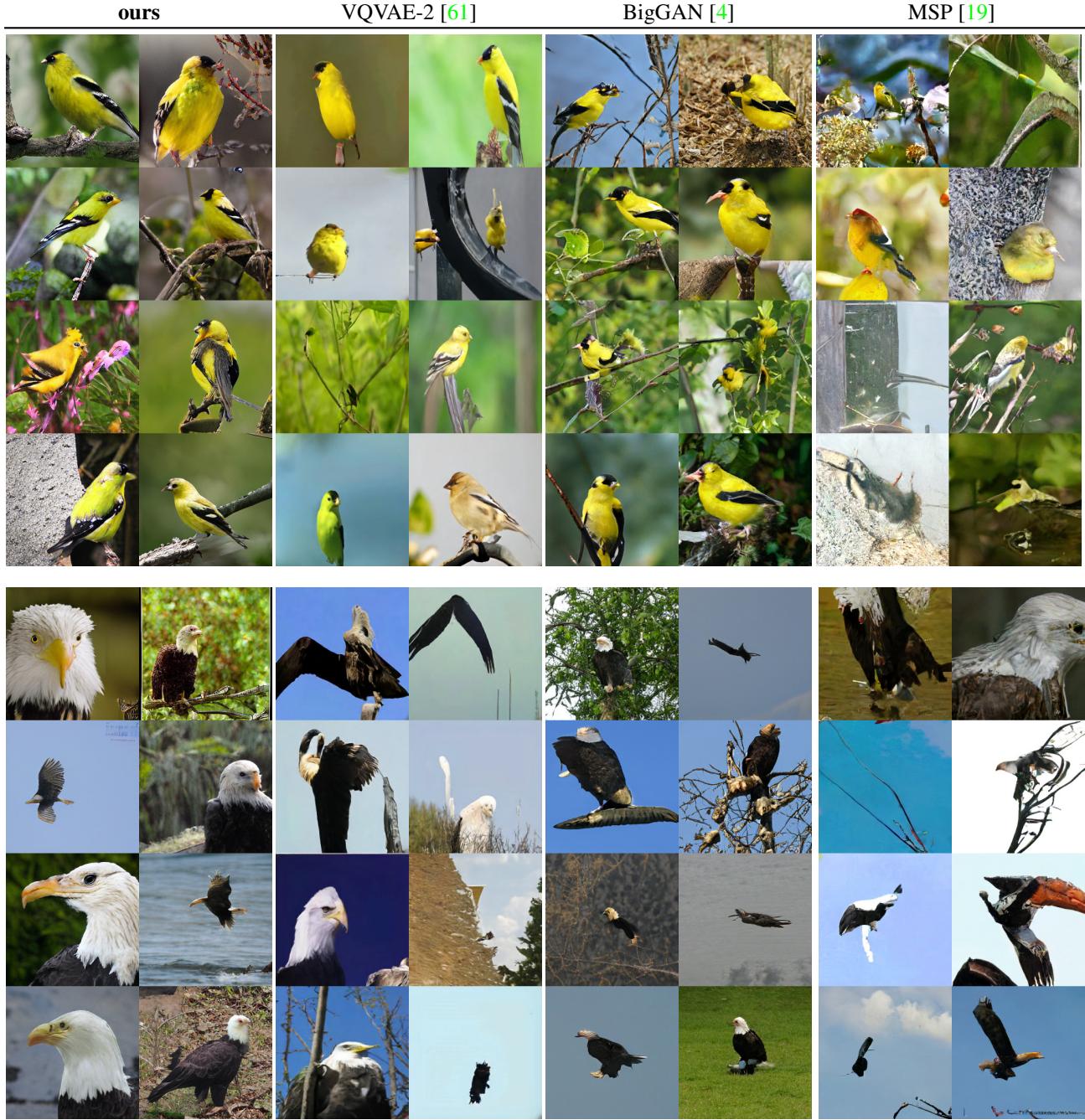


Figure 16. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: 11: goldfinch (top) and 22: bald eagle (bottom).

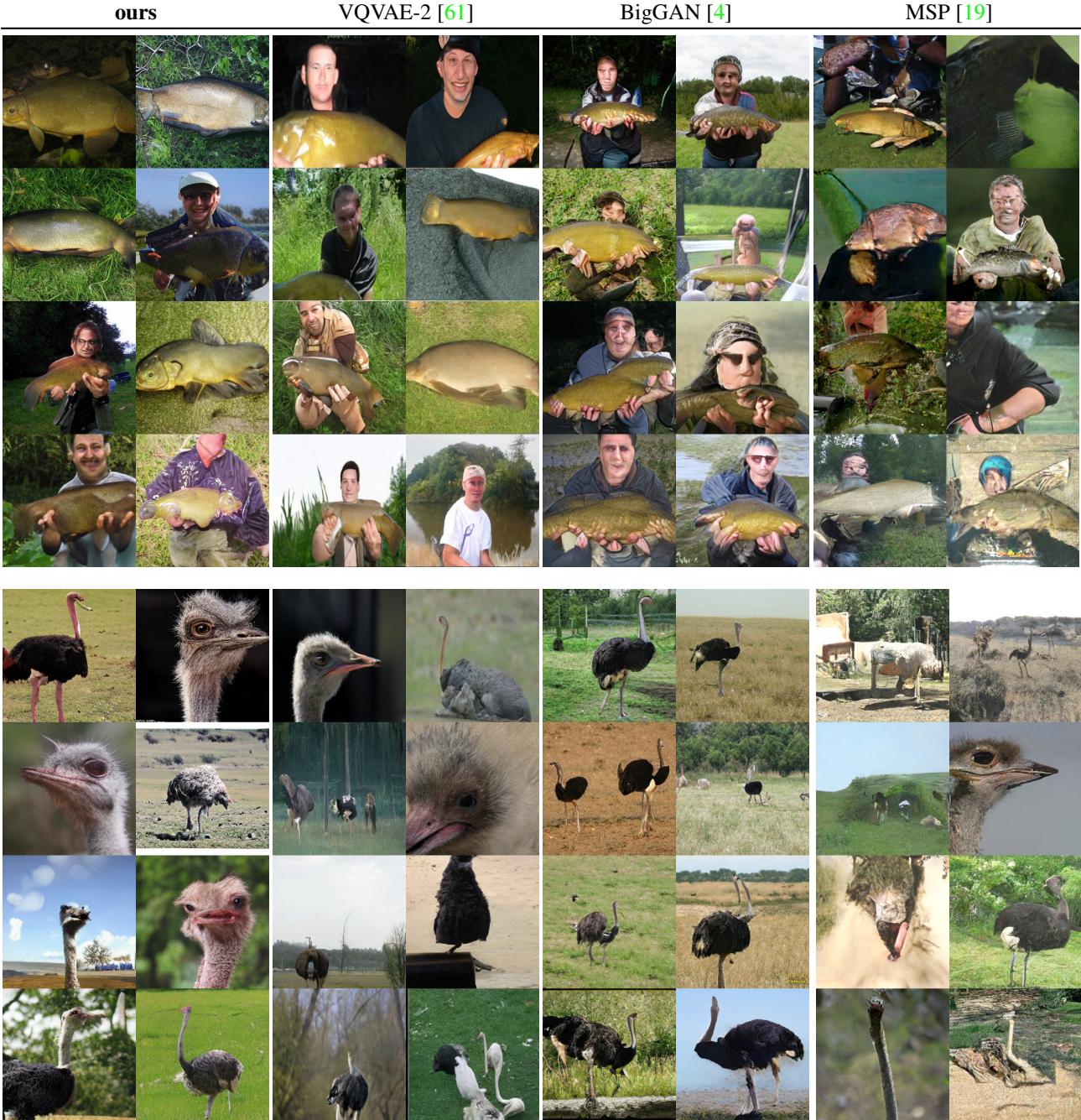


Figure 17. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: 0: tench (top) and 9: ostrich (bottom).

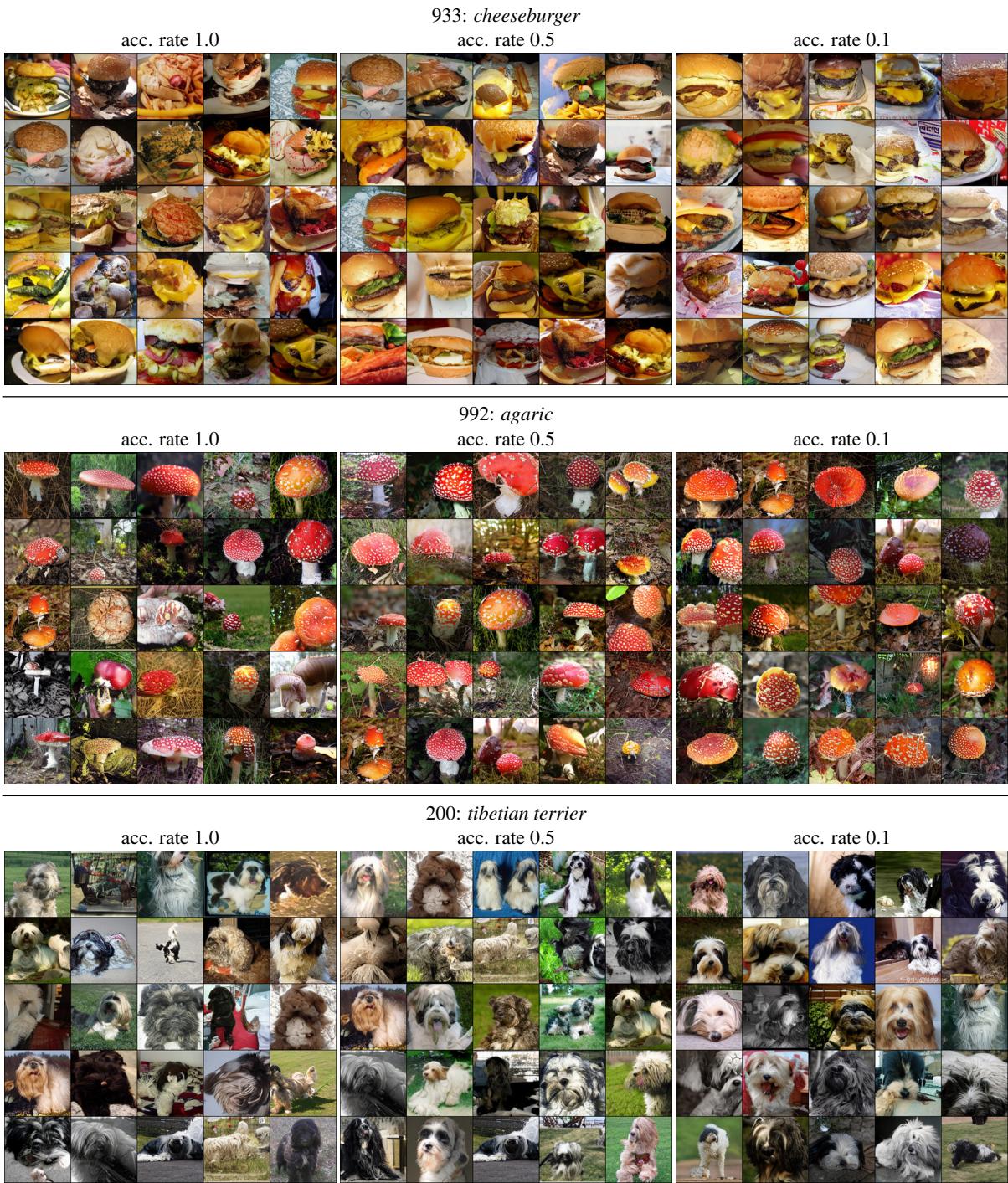


Figure 18. Visualizing the effect of increased rejection rate (*i.e.* lower acceptance rate) by using a ResNet-101 classifier trained on ImageNet and samples from our class-conditional ImageNet model. Higher rejection rates tend to produce images showing more central, recognizable objects compared to the unguided samples. Here, $k = 973$, $p = 1.0$ are fixed for all samples. Note that $k = 973$ is the effective size of the VQGAN’s codebook, *i.e.* it describes how many entries of the codebook with $\dim \mathcal{Z} = 16384$ are actually used.

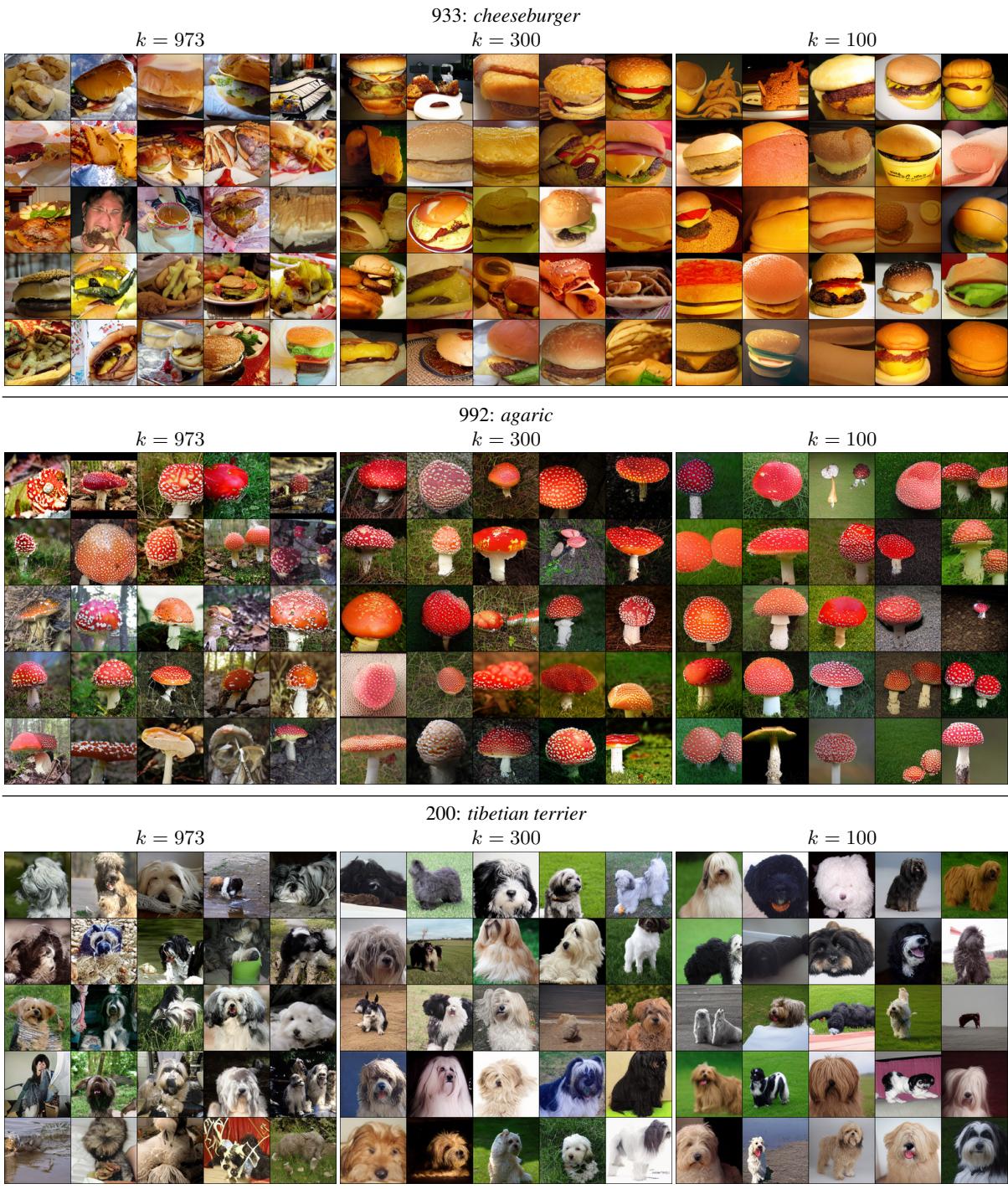


Figure 19. Visualizing the effect of varying k in top- k sampling (*i.e.* truncating the probability distribution per image token) by using a ResNet-101 classifier trained on ImageNet and samples from our class-conditional ImageNet model. Lower values of k produce more uniform, low-entropic images compared to samples obtained with full k . Here, an acceptance rate of 1.0 and $p = 1.0$ are fixed for all samples. Note that $k = 973$ is the *effective* size of the VQGAN’s codebook, *i.e.* it describes how many entries of the codebook with $\dim \mathcal{Z} = 16384$ are actually used.



Figure 20. Visualizing the effect of varying p in top- p sampling (or *nucleus sampling* [27]) by using a ResNet-101 classifier trained on ImageNet and samples from our class-conditional ImageNet model. Lowering p has similar effects as decreasing k , see Fig. 19. Here, an acceptance rate of 1.0 and $k = 973$ are fixed for all samples.

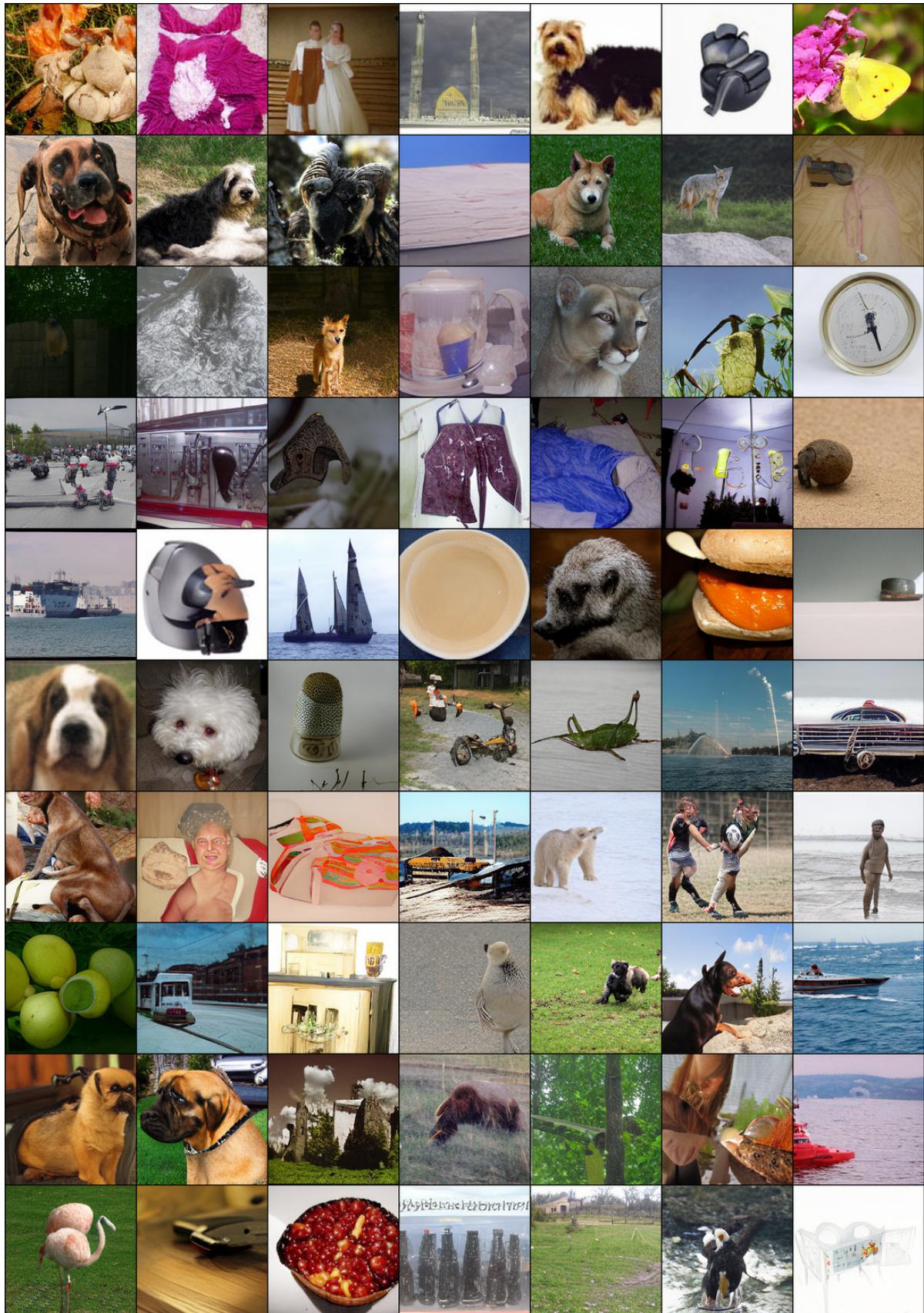


Figure 21. Random samples on 256×256 class-conditional ImageNet with $k \in [100, 200, 250, 300, 350, 400, 500, 600, 800, 973]$, $p = 1.0$, acceptance rate 1.0. FID: 17.04, IS: 70.6 ± 1.8 . Please see <https://git.io/JLlvY> for an uncompressed version.

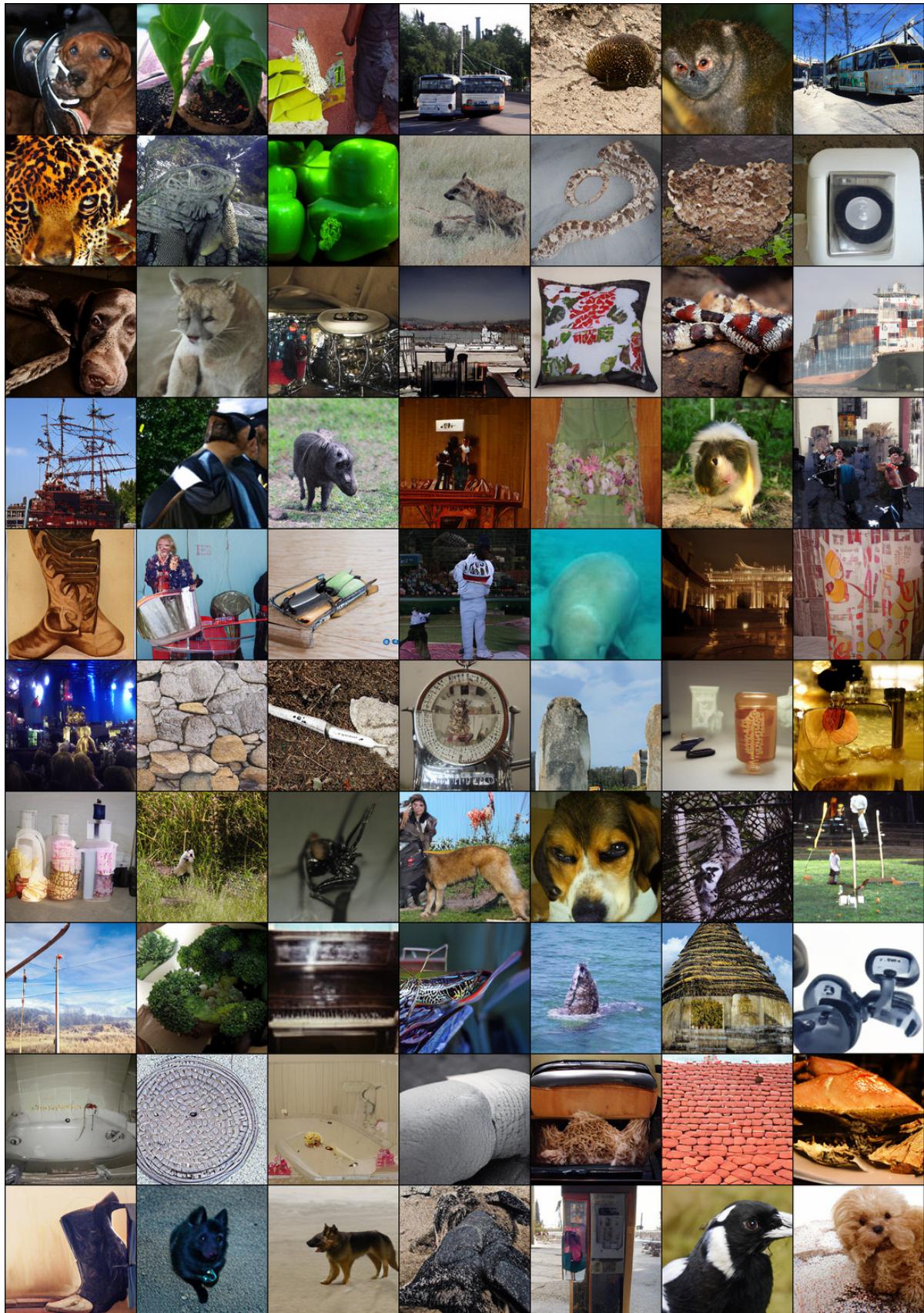


Figure 22. Random samples on 256×256 class-conditional ImageNet with $k = 600$, $p = 1.0$, acceptance rate 0.05. FID: 5.20, IS: 280.3 \pm 5.5. Please see <https://git.io/JLlvY> for an uncompressed version.

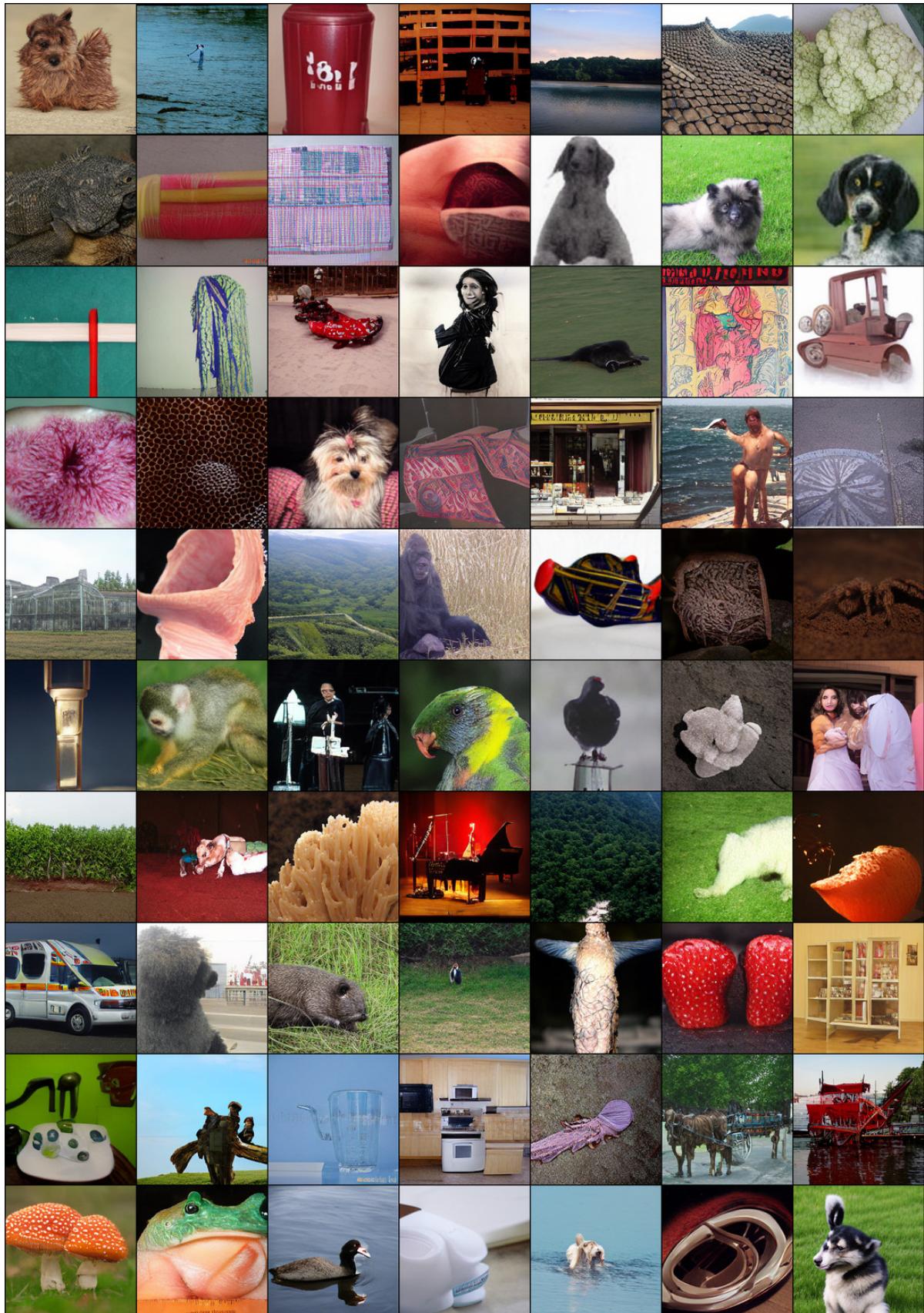


Figure 23. Random samples on 256×256 class-conditional ImageNet with $k = 250$, $p = 1.0$, acceptance rate 1.0. FID: 15.98, IS: 78.6 ± 1.1 . Please see <https://git.io/JLlvY> for an uncompressed version.

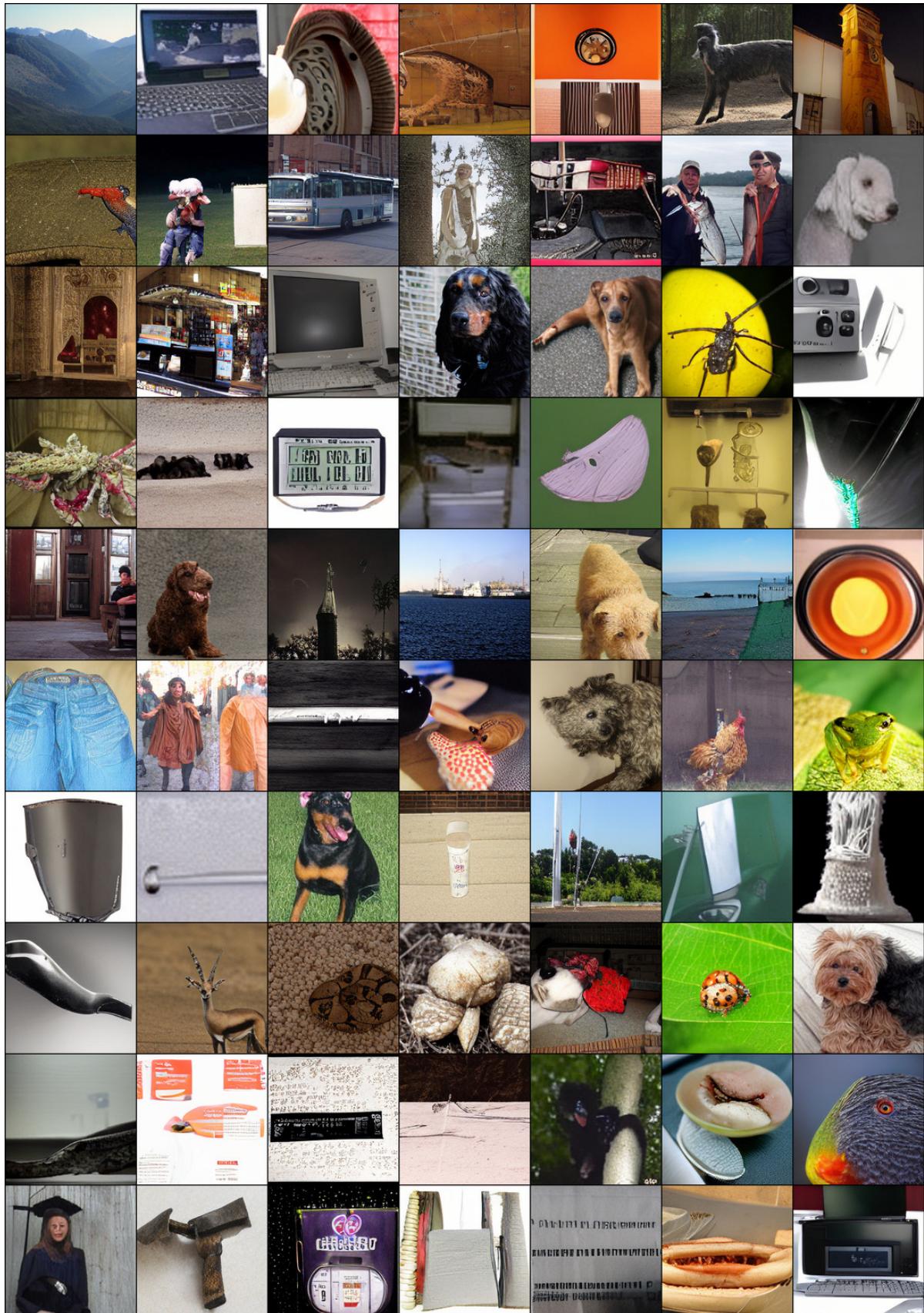


Figure 24. Random samples on 256×256 class-conditional ImageNet with $k = 973$, $p = 0.88$, acceptance rate 1.0. FID: 15.78, IS: 74.3 ± 1.8 . Please see <https://git.io/JLlvY> for an uncompressed version.



Figure 25. Random samples on 256×256 class-conditional ImageNet with $k \in [100, 200, 250, 300, 350, 400, 500, 600, 800, 973]$, $p = 1.0$, acceptance rate 0.005. FID: 6.59, IS: 402.7 ± 2.9 . Please see <https://git.io/JLlvY> for an uncompressed version.

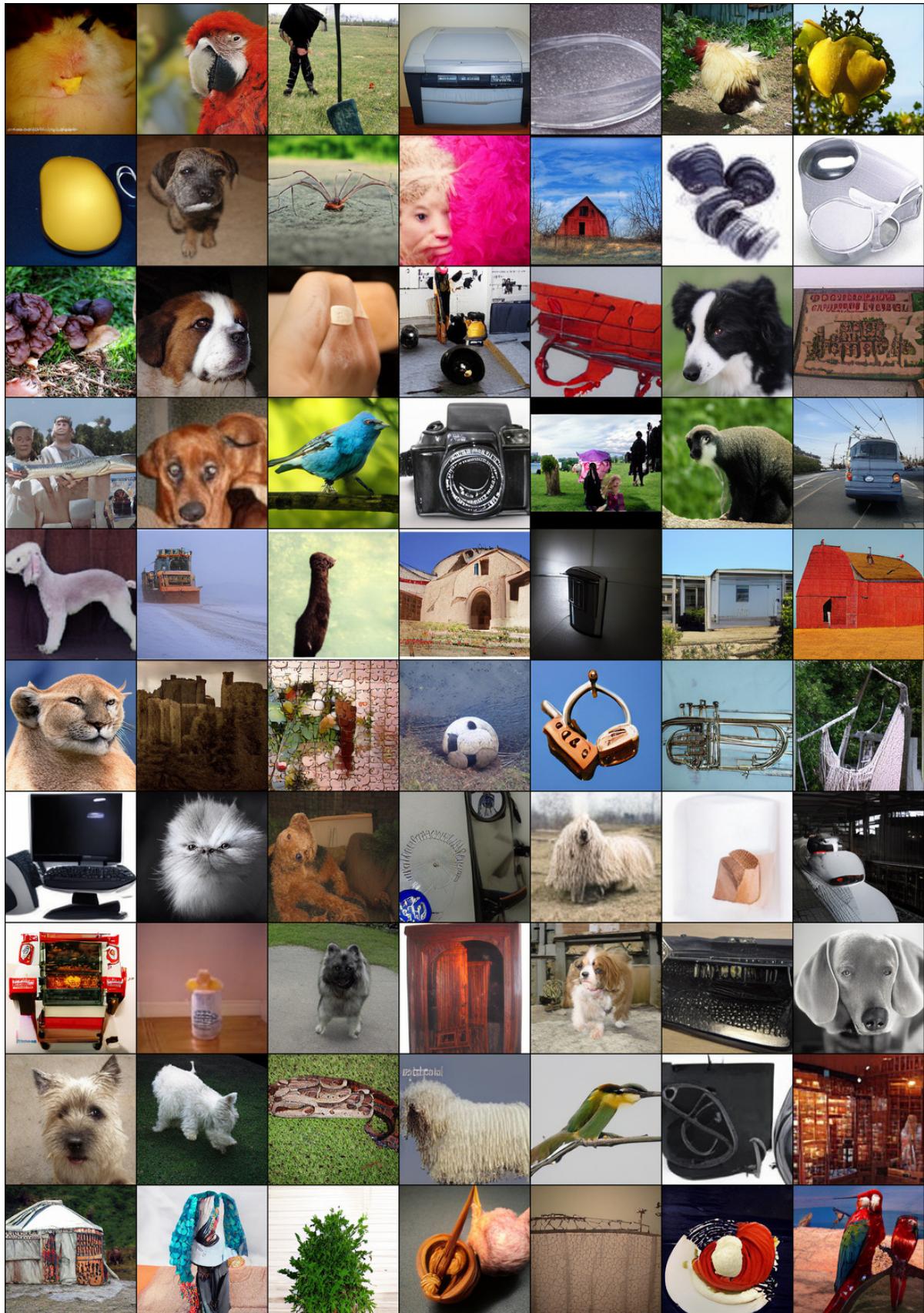


Figure 26. Random samples on 256×256 class-conditional ImageNet with $k \in [100, 200, 250, 300, 350, 400, 500, 600, 800, 973]$, $p = 1.0$, acceptance rate 0.05. FID: 5.88, IS: 304.8 ± 3.6 . Please see <https://git.io/J1lvY> for an uncompressed version.

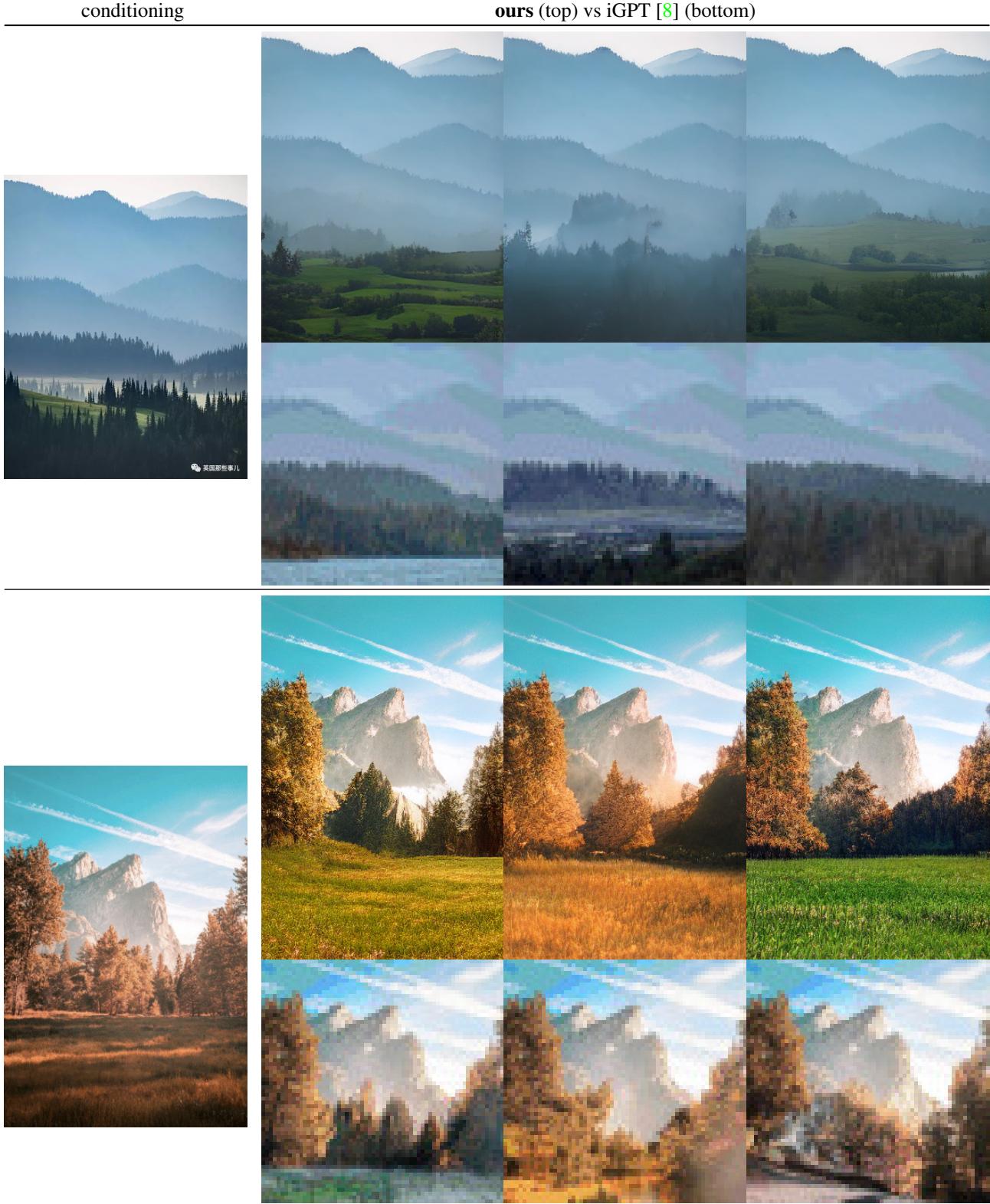


Figure 27. Comparing our approach with the pixel-based approach of [8]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [8] (bottom).

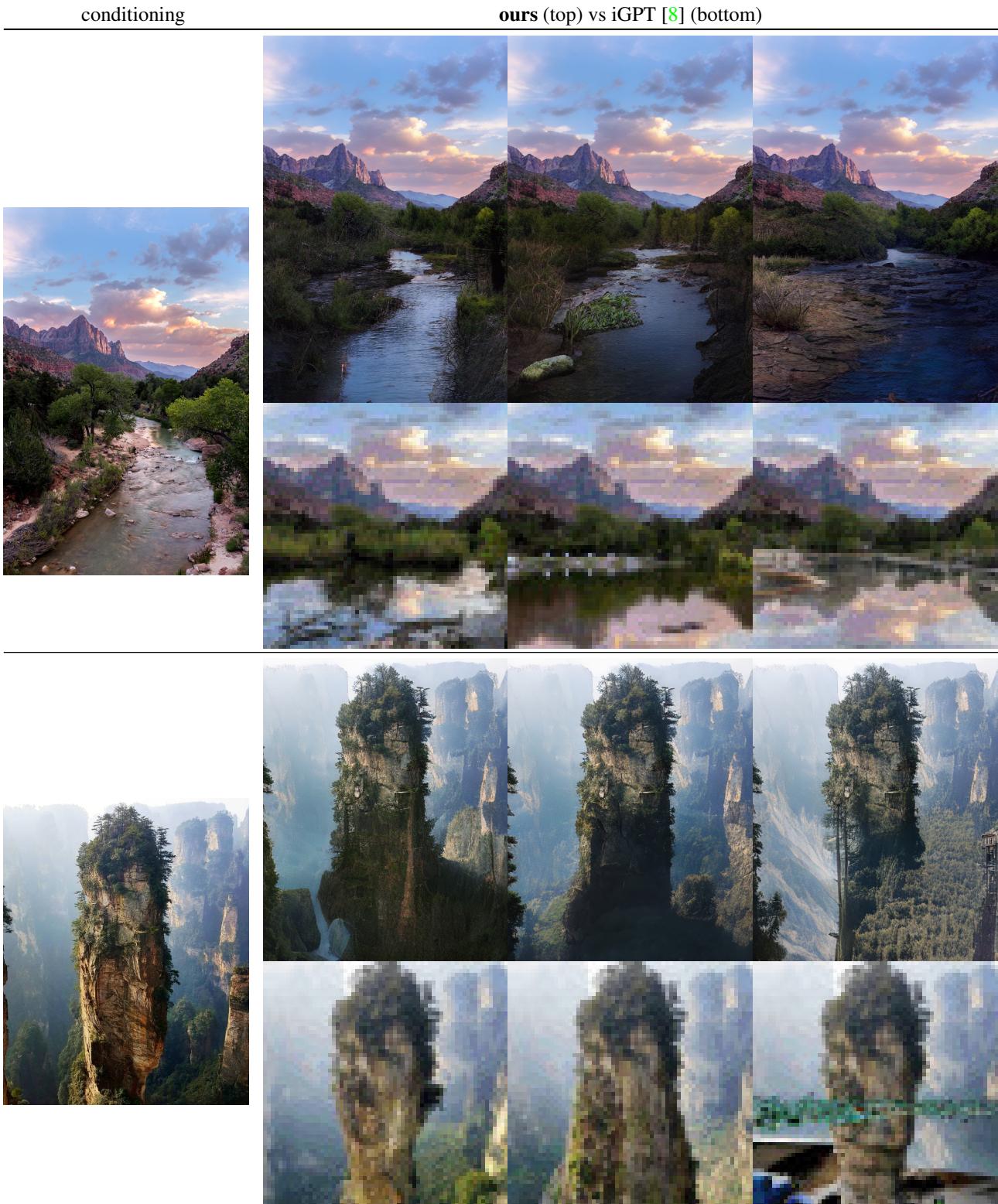


Figure 28. Comparing our approach with the pixel-based approach of [8]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [8] (bottom).

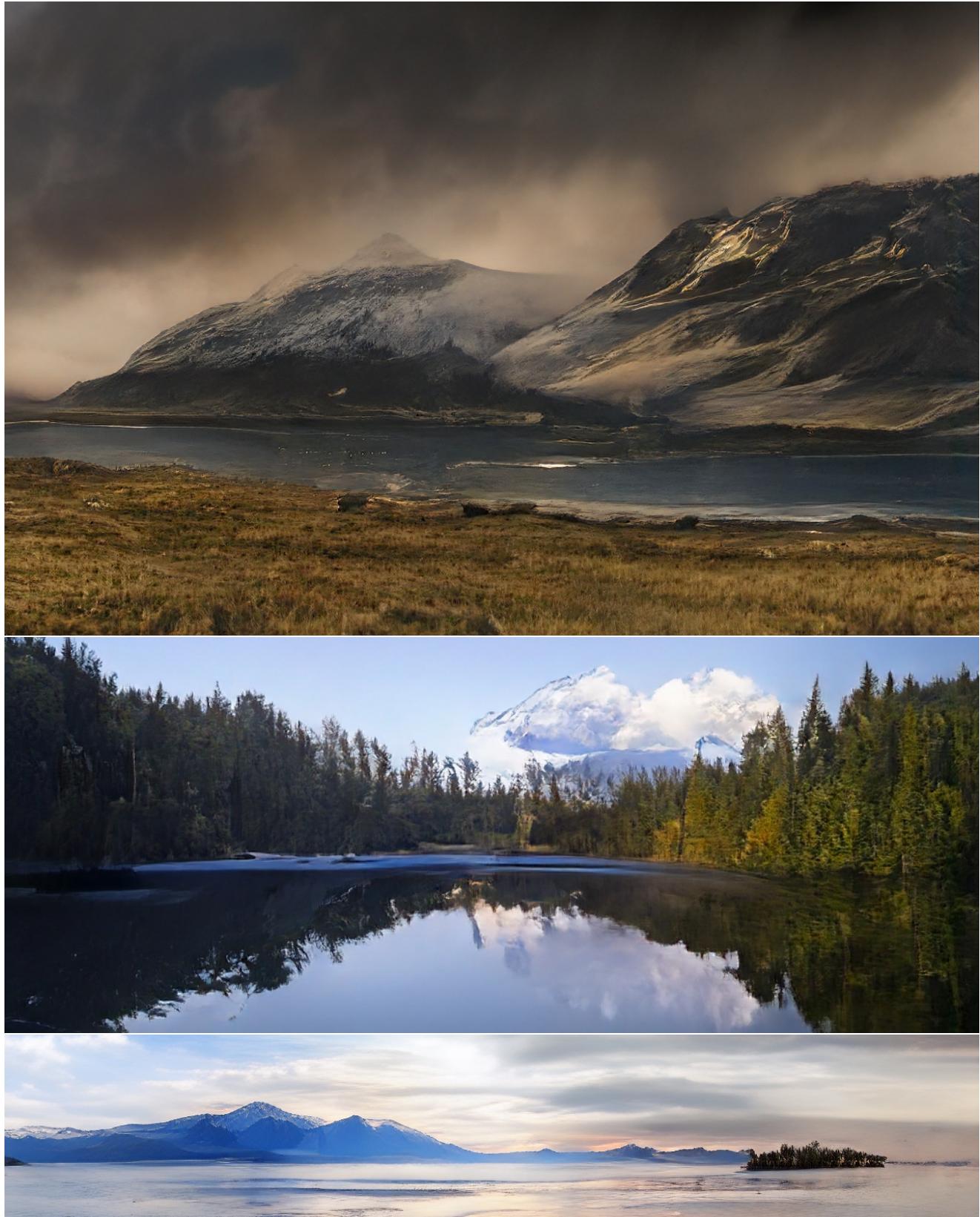


Figure 29. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832 , 1024×416 and 1280×240 pixels.

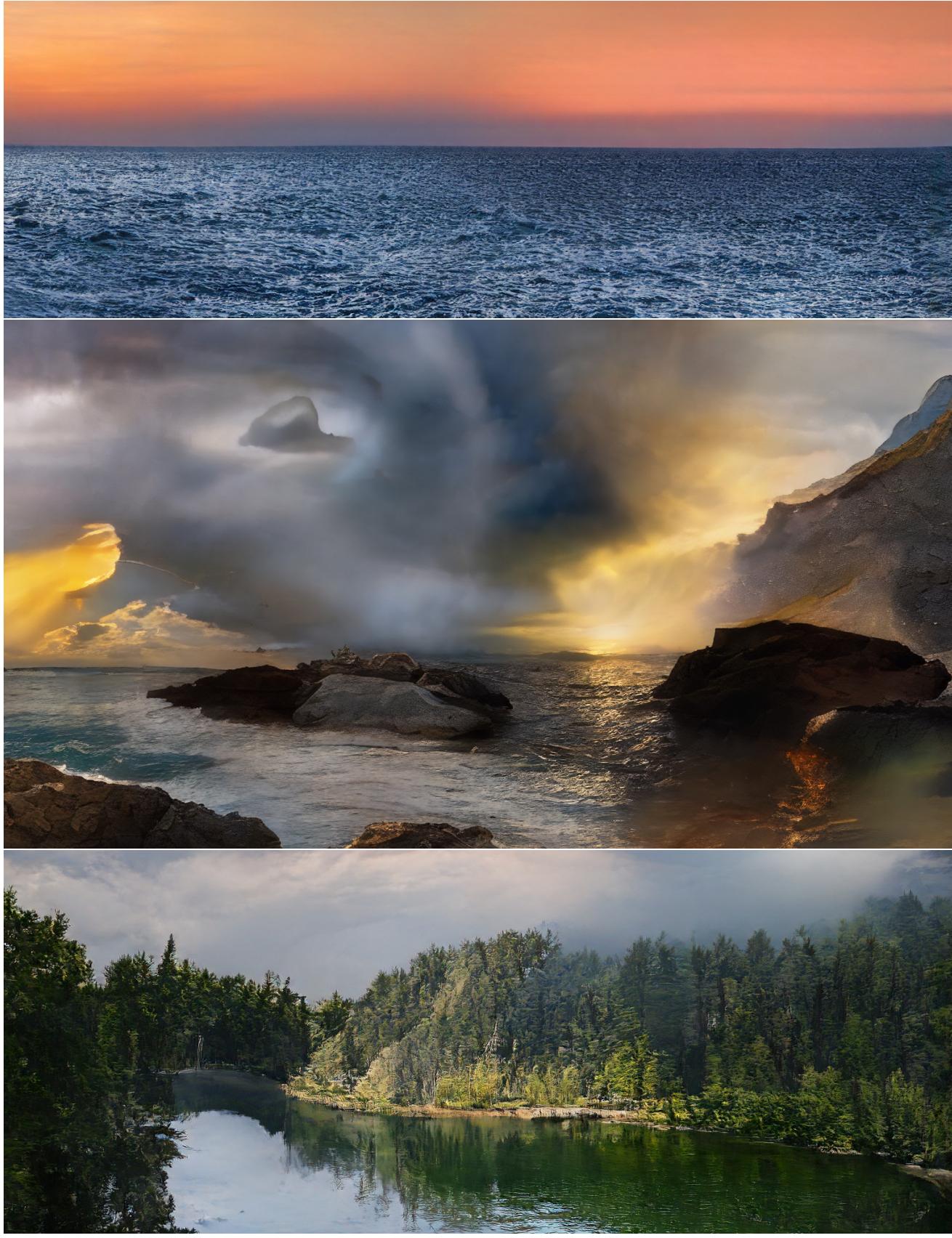


Figure 30. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1536×512 , 1840×1024 , and 1536×620 pixels.

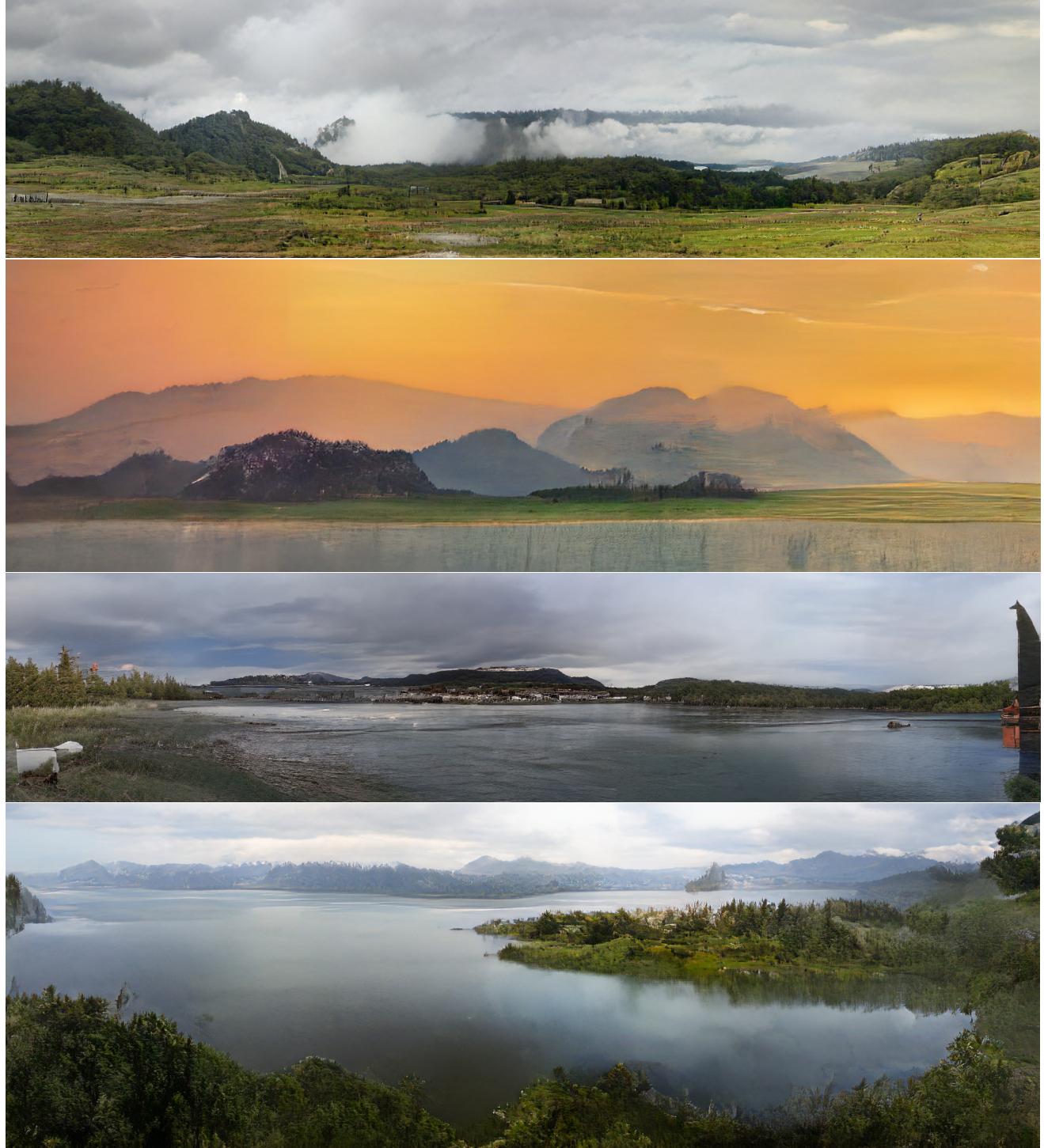


Figure 31. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 2048×512 , 1460×440 , 2032×448 and 2016×672 pixels.

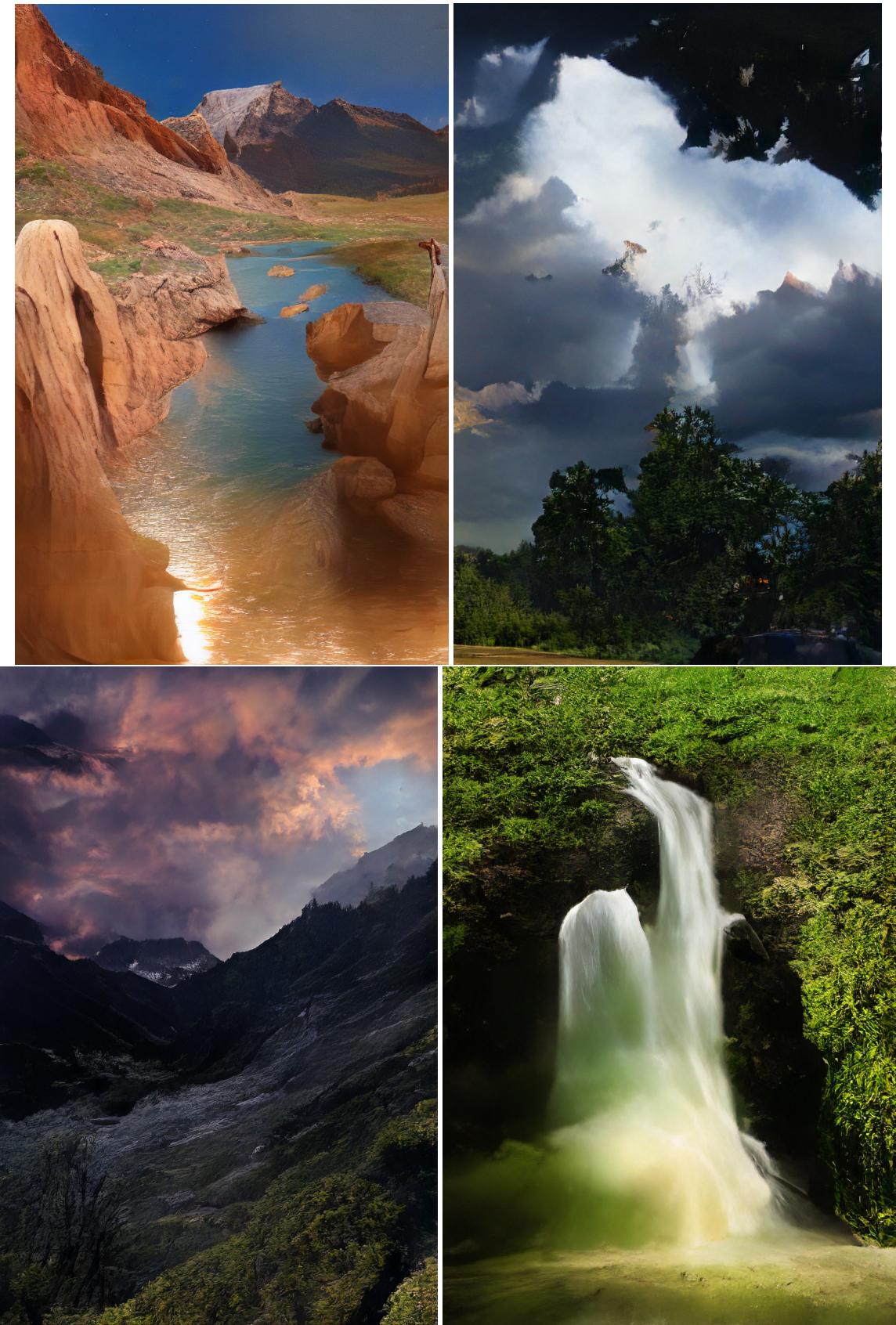


Figure 32. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832 , 1024×416 and 1280×240 pixels.

conditioning

samples

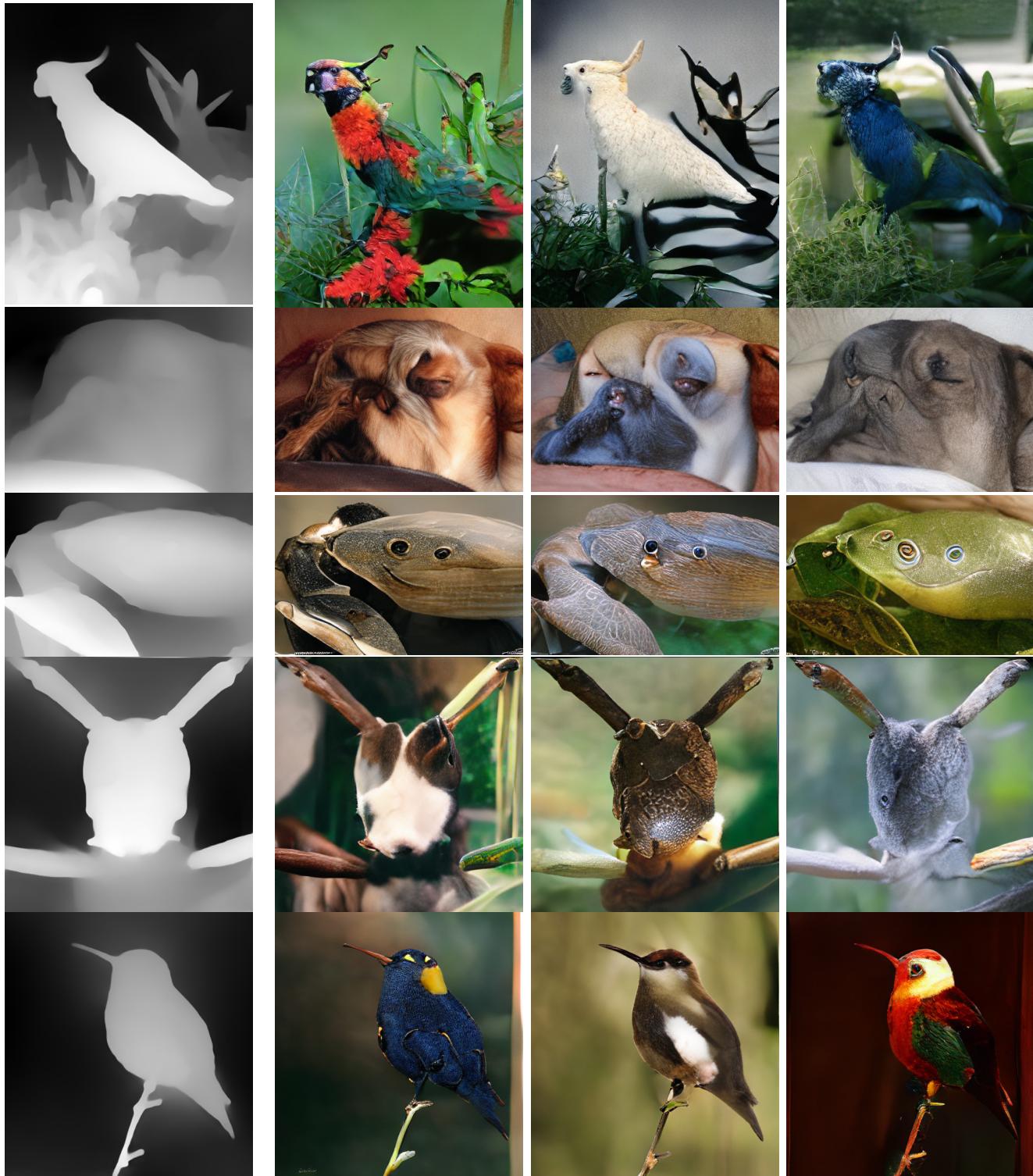
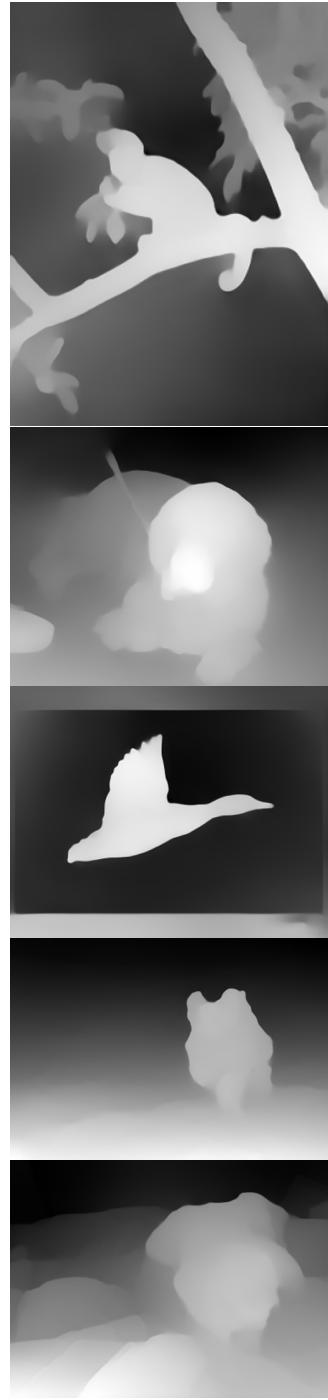


Figure 33. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.

conditioning



samples

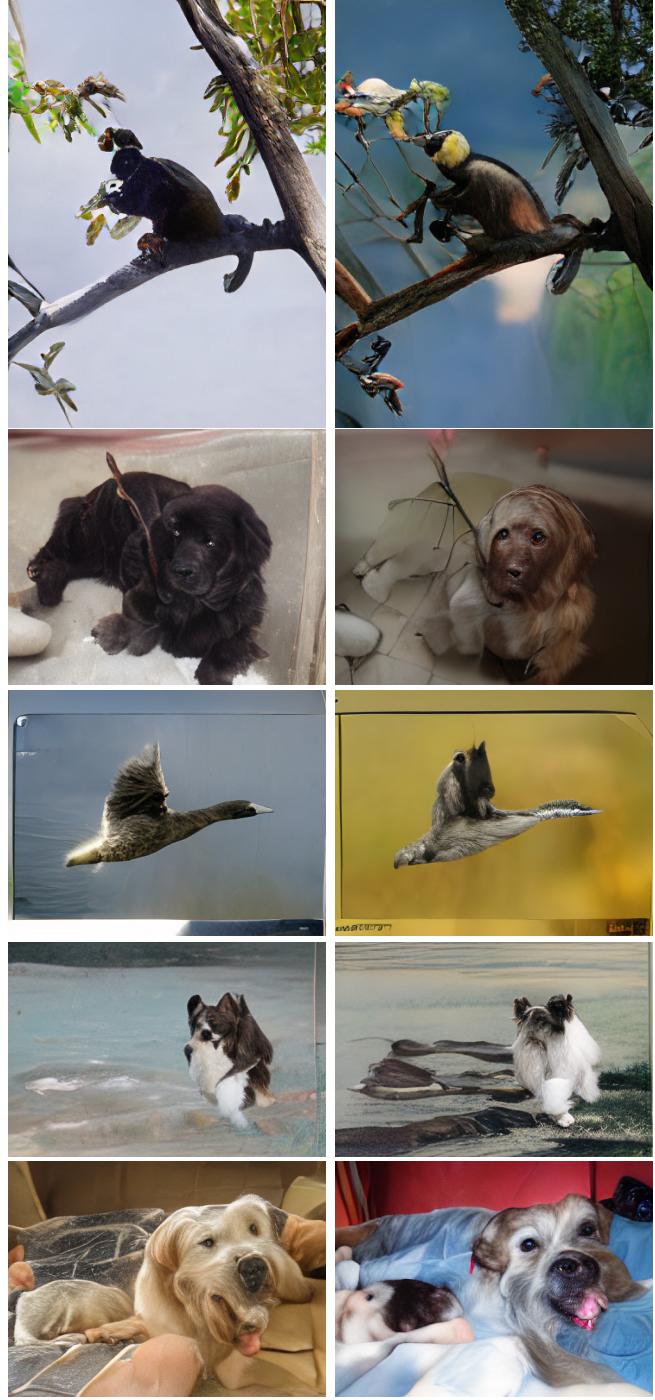


Figure 34. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.



Figure 35. 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.

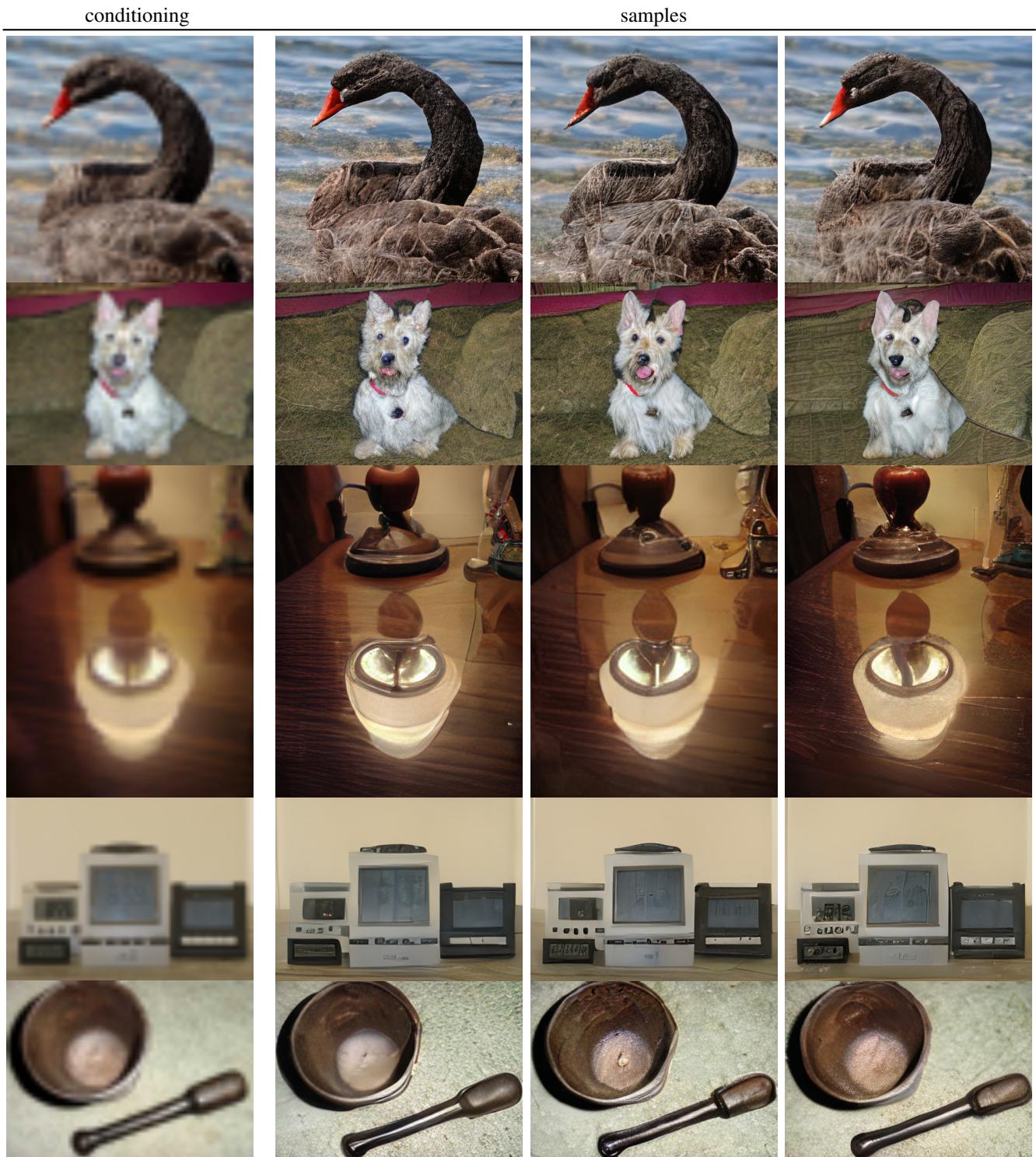


Figure 36. Additional results for stochastic superresolution with an $f = 16$ model on IN, using the sliding attention window.

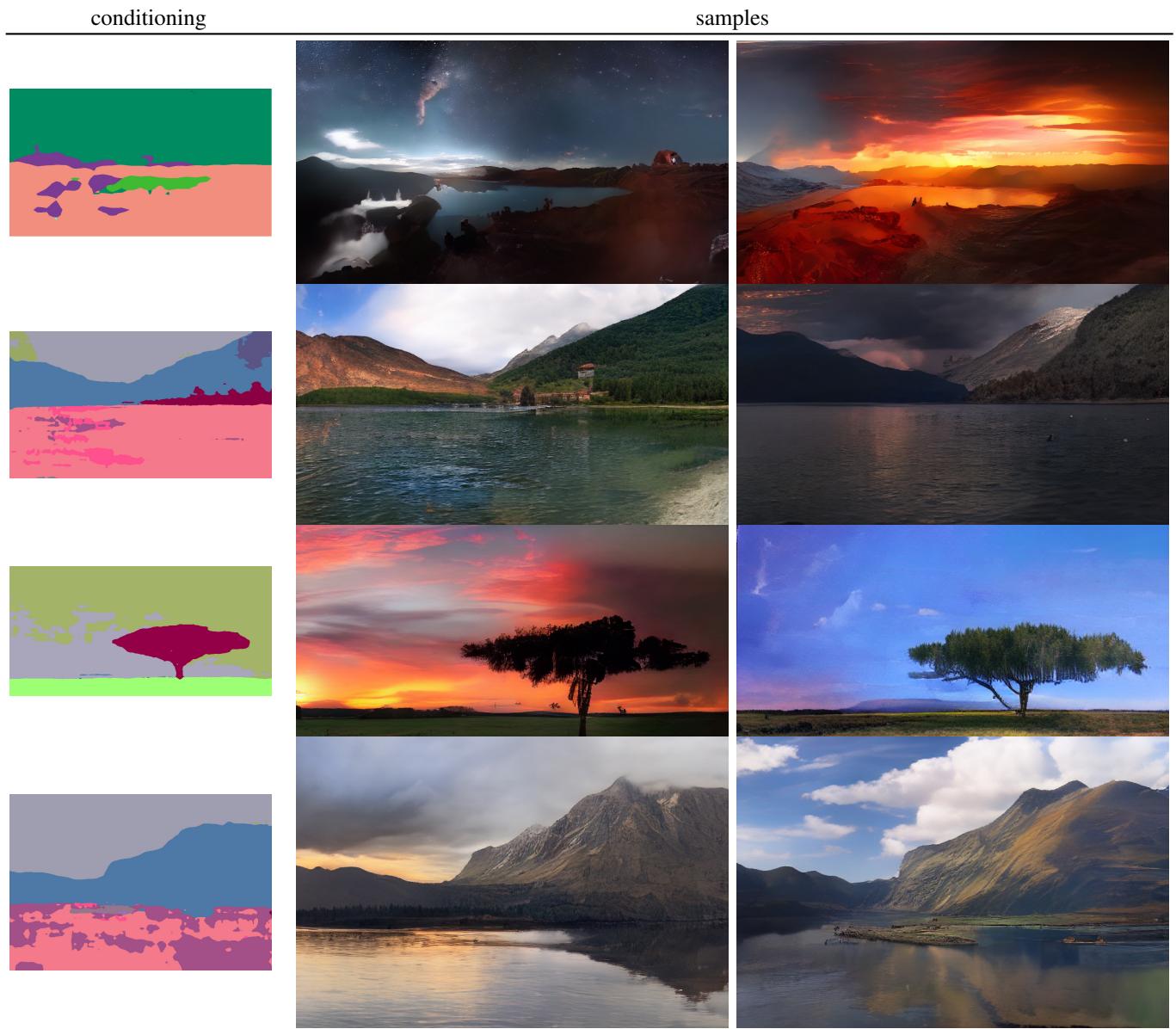


Figure 37. Samples generated from semantic layouts on S-FLCKR with $f = 16$, using the sliding attention window.

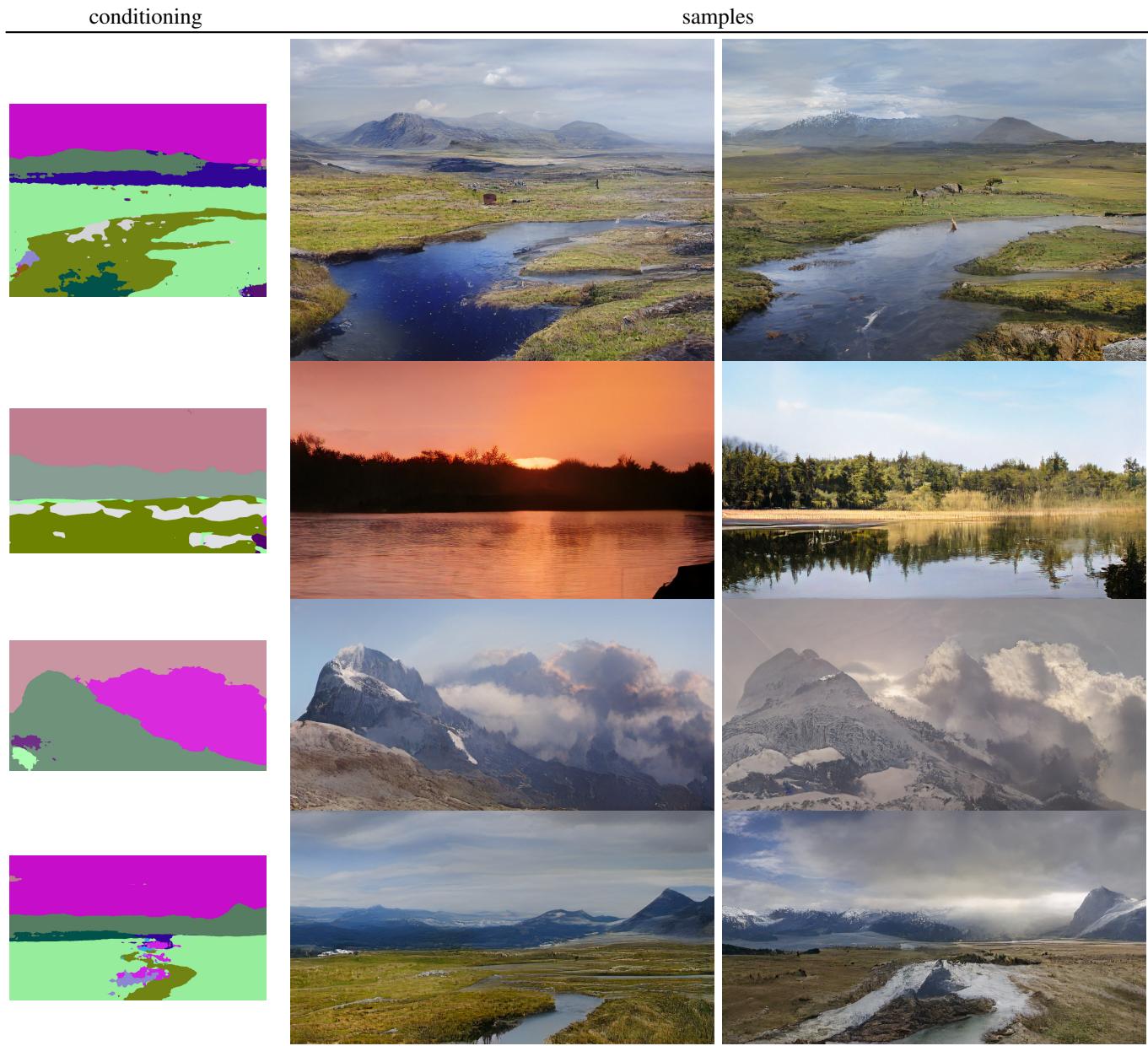


Figure 38. Samples generated from semantic layouts on S-FLCKR with $f = 32$, using the sliding attention window.



Figure 39. Unconditional samples from a model trained on LSUN Churches & Towers, using the sliding attention window.

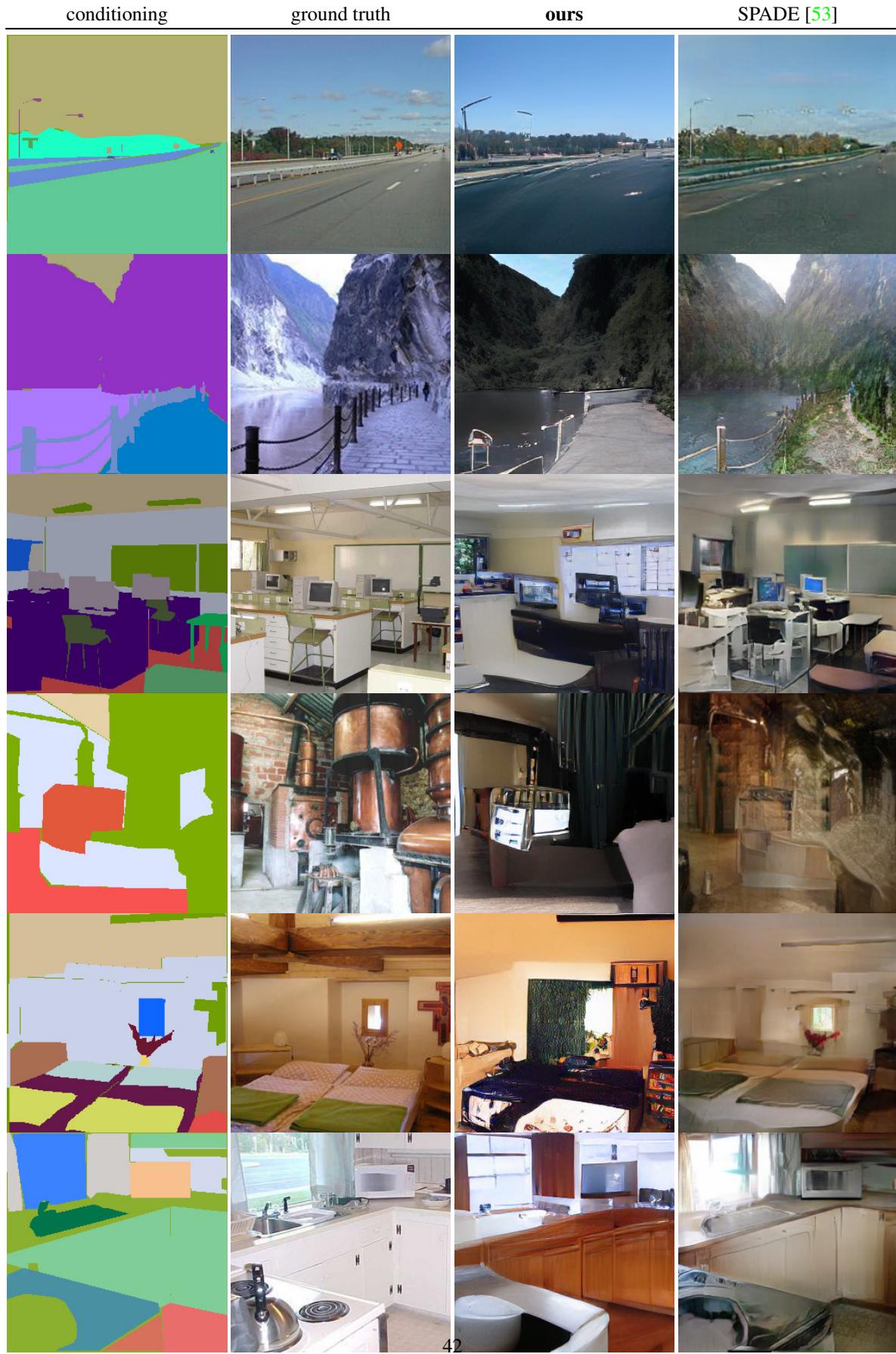


Figure 40. Qualitative comparison to [53] on 256×256 images from the ADE20K dataset.



Figure 41. Qualitative comparison to [53] on 256×256 images from the COCO-Stuff dataset.

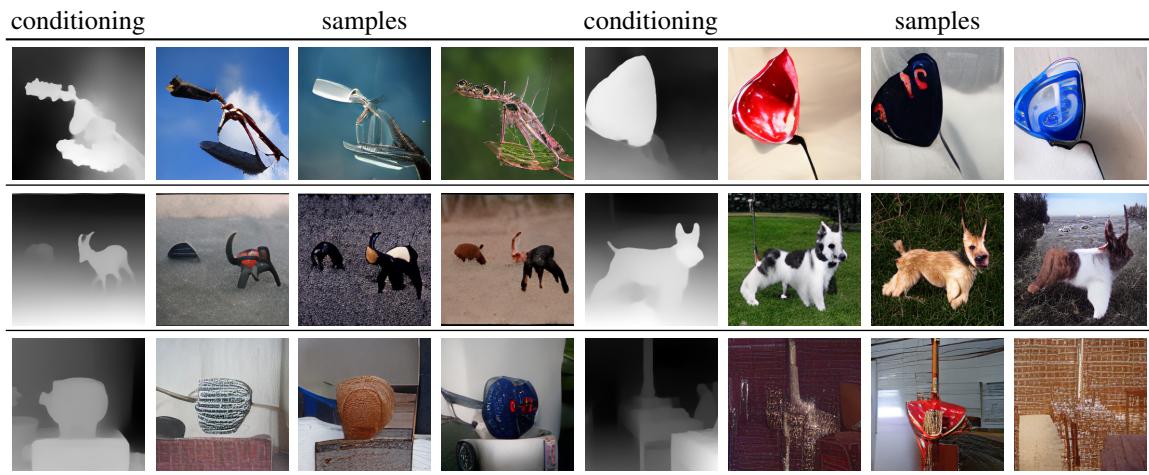


Figure 42. Conditional samples for the depth-to-image model on IN.

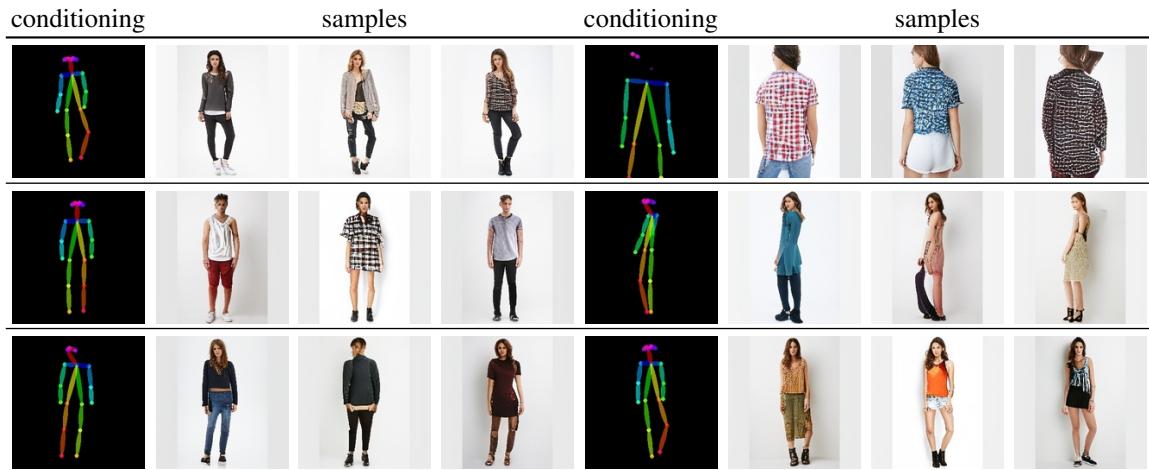


Figure 43. Conditional samples for the pose-guided synthesis model via keypoints on DeepFashion.

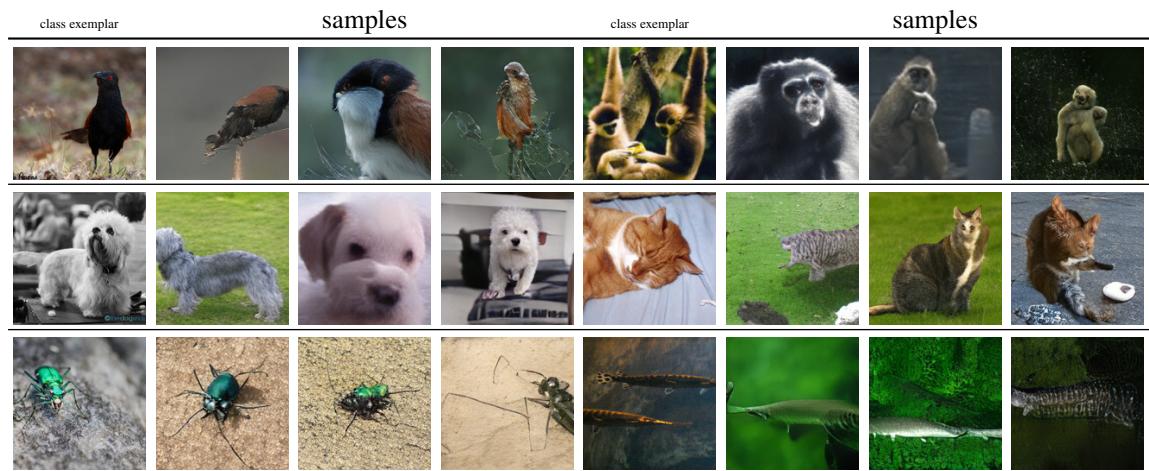


Figure 44. Samples produced by the class-conditional model trained on RIN.

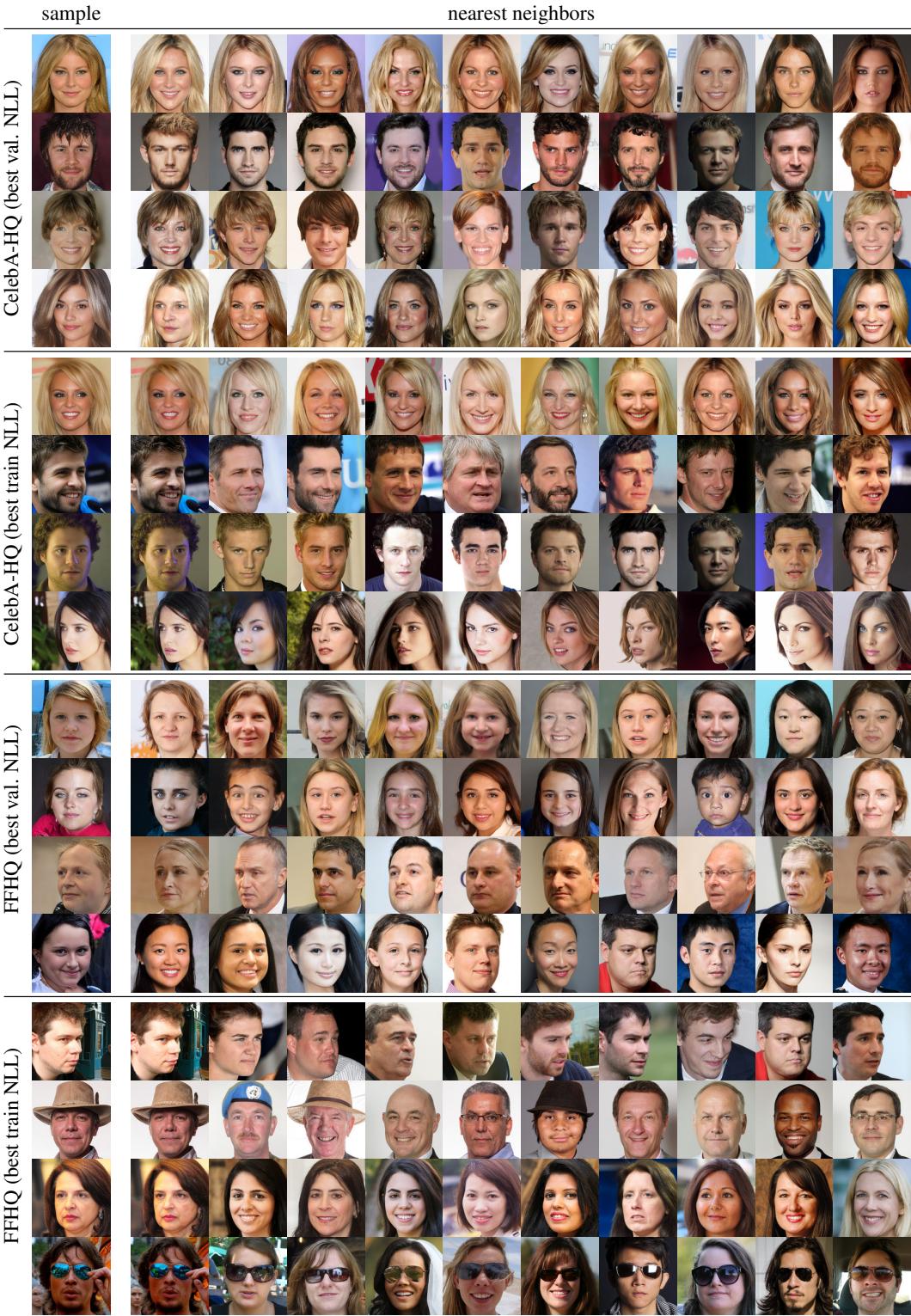


Figure 45. Nearest neighbors for our face-models trained on FFHQ and CelebA-HQ (256×256 pix), based on the LPIPS [82] distance. The left column shows a sample from our model, while the 10 examples to the right show the nearest neighbors from the corresponding class (increasing distance) in the training dataset. We evaluate two different model checkpoints for each dataset: Best val. NLL denotes the minimal NLL over the course of training, evaluated on unseen testdata. For this checkpoint, both models generate crisp, high-quality samples not present in the training data. However, when drastically overfitting the model, it reproduces samples from the training data (best train NLL). Although not an ideal measure of image quality, NLL thus provides a proxy on model selection, whereas FID does not. See also Sec. E.

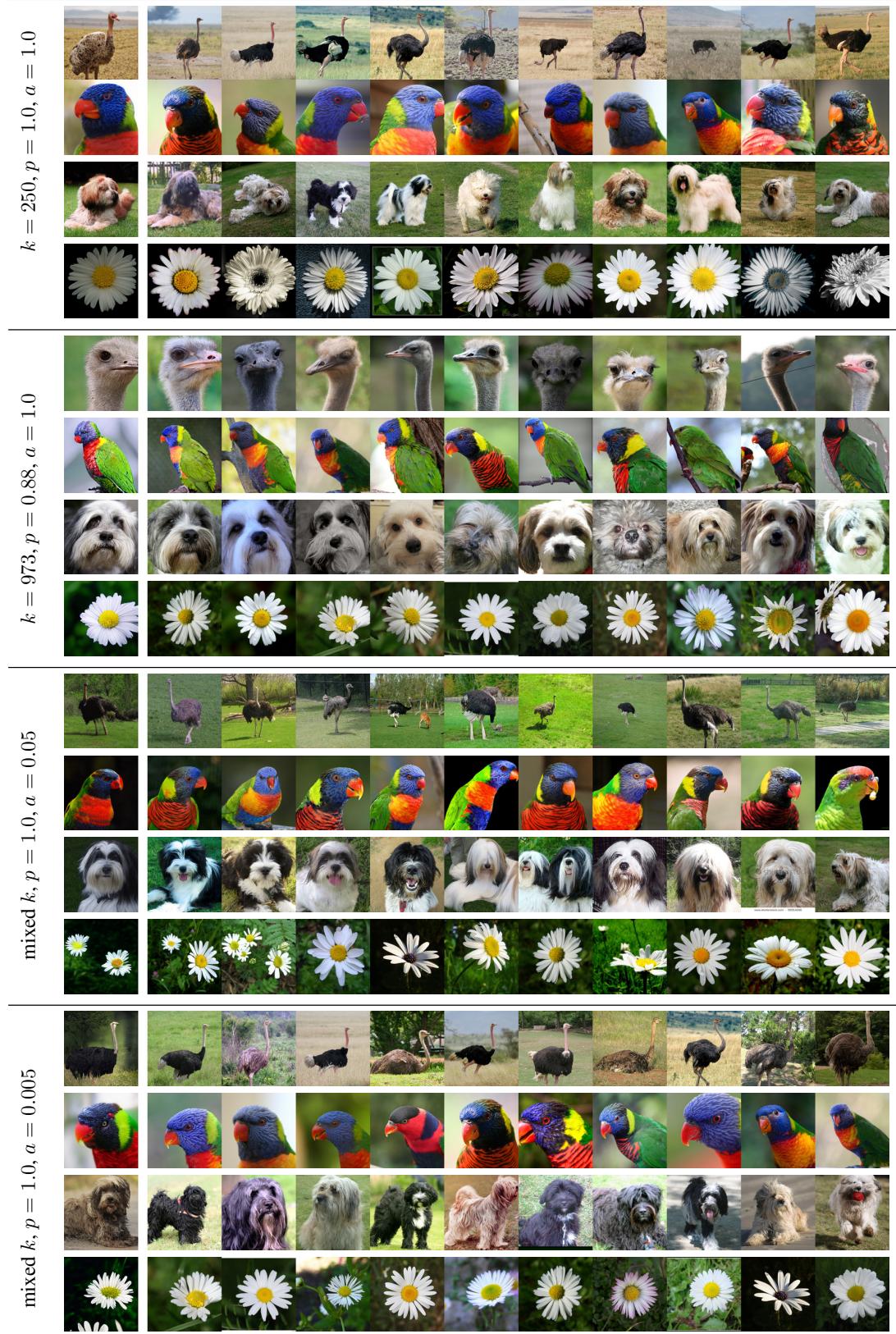


Figure 46. Nearest neighbors for our class-conditional ImageNet model (256×256 pix), based on the LPIPS [82] distance. The left column shows a sample from our model, while the 10 examples to the right show the nearest neighbors from the corresponding class (increasing distance) in the training dataset. Our model produces new, unseen high-quality images, not present in the training data.

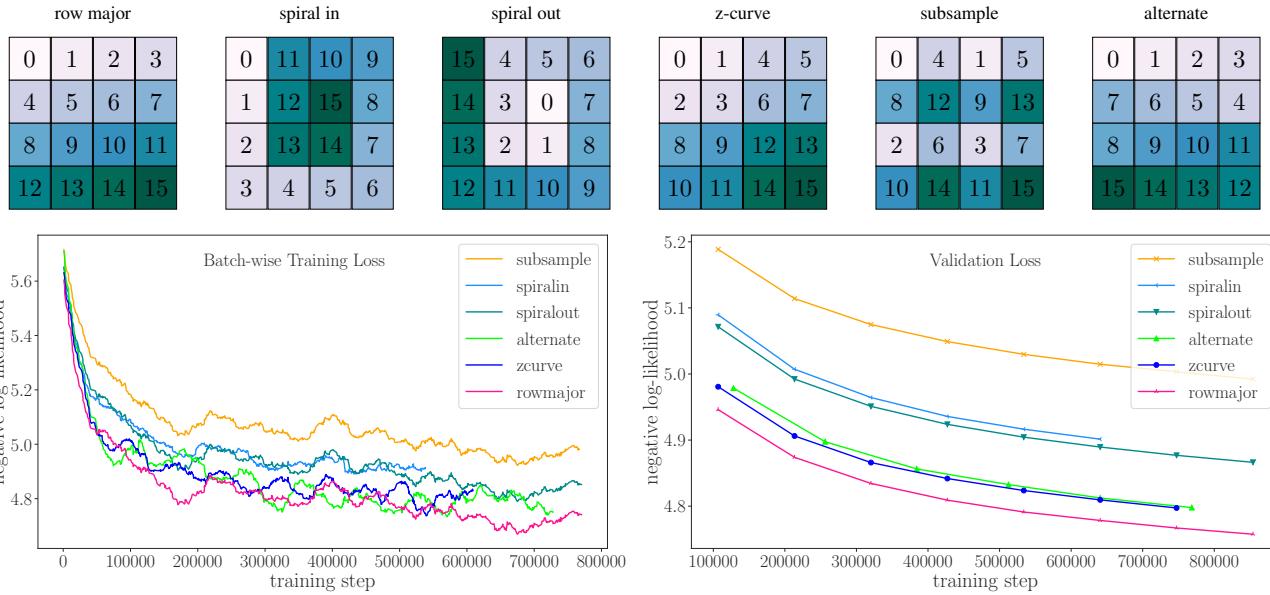


Figure 47. Top: All sequence permutations we investigate, illustrated on a 4×4 grid. Bottom: The transformer architecture is permutation invariant but next-token prediction is not: The average loss on the validation split of ImageNet, corresponding to the negative log-likelihood, differs significantly between different prediction orderings. Among our choices, the commonly used row-major order performs best.

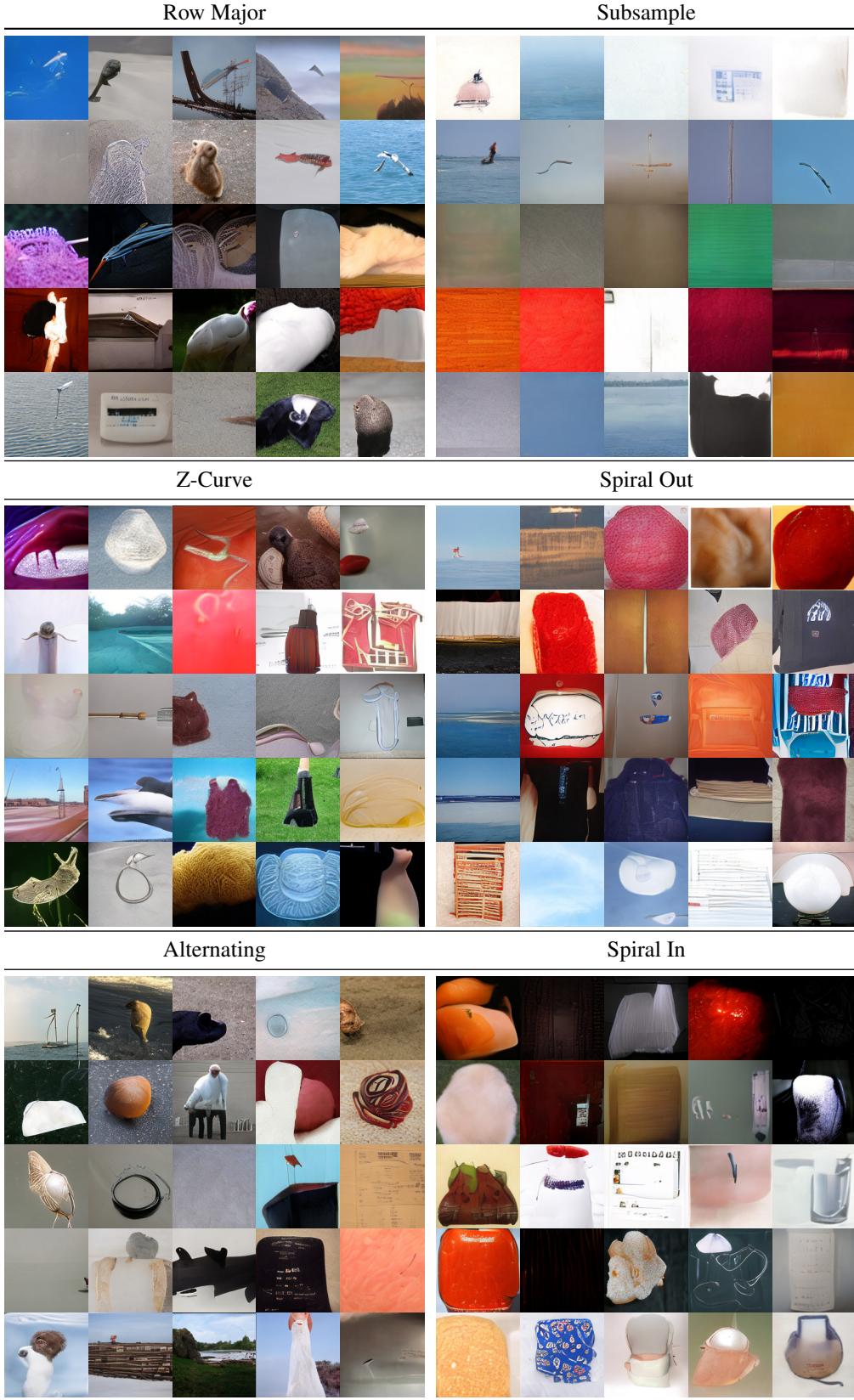


Figure 48. Random samples from transformer models trained with different orderings for autoregressive prediction as described in Sec. F.

References

- [1] Jyoti Aneja, Alexander G. Schwing, Jan Kautz, and Arash Vahdat. NCP-VAE: variational autoencoders with noise contrastive priors. *CoRR*, abs/2010.02917, 2020. 8
- [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] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR*, 2019. 8, 10, 16, 17, 18, 19
- [5] 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
- [6] 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
- [7] 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
- [8] 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, 6, 7, 12, 13, 14, 29, 30
- [9] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1520–1529. IEEE Computer Society, 2017. 7
- [10] 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
- [11] Rewon Child. Very deep vae’s generalize autoregressive models and can outperform them on images. *CoRR*, abs/2011.10650, 2020. 8
- [12] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. 1, 2, 5
- [13] Bin Dai and David P. Wipf. Diagnosing and enhancing VAE models. In *7th International Conference on Learning Representations, ICLR*, 2019. 2
- [14] 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
- [15] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 8, 10
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020. 1
- [17] 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
- [18] 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
- [19] Jeffrey De Fauw, Sander Dieleman, and Karen Simonyan. Hierarchical autoregressive image models with auxiliary decoders. *CoRR*, abs/1903.04933, 2019. 16, 17, 18, 19
- [20] 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. 2
- [21] 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. 2
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 8
- [23] Ari Heljakka, Arno Solin, and Juho Kannala. Pioneer networks: Progressively growing generative autoencoder. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision - ACCV 2018 - 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part I*, 2018. 8
- [24] Ari Heljakka, Arno Solin, and Juho Kannala. Towards photographic image manipulation with balanced growing of generative autoencoders. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 3109–3118. IEEE, 2020. 8

- [25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 11
- [26] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *CoRR*, abs/1912.12180, 2019. 2, 5
- [27] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*. OpenReview.net, 2020. 8, 22
- [28] 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, 11
- [29] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 9
- [30] 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
- [31] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 6, 8
- [32] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 8
- [33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019. 7
- [34] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. IEEE, 2020. 8
- [35] Prateek Katiyar and Anna Khoreva. Improving augmentation and evaluation schemes for semantic image synthesis, 2021. 7
- [36] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks, 2017. 2
- [37] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS, 2018*. 8
- [38] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR, 2014*. 2
- [39] Alex Lamb, Vincent Dumoulin, and Aaron C. Courville. Discriminative regularization for generative models. *CoRR*, abs/1602.03220, 2016. 4
- [40] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2015. 4
- [41] 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
- [42] 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
- [43] Jinlin Liu, Yuan Yao, and Jianqiang Ren. An acceleration framework for high resolution image synthesis. *CoRR*, abs/1909.03611, 2019. 2
- [44] 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
- [45] 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
- [46] Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 14
- [47] Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression, 2020. 4
- [48] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations, 2021. 8, 10
- [49] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. 8, 10
- [50] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. 7, 10
- [51] 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
- [52] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference, 2016. 2

- [53] 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, 7, 42, 43
- [54] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder, 2020. 8
- [55] 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
- [56] Stanislav Pidhorskyi, Donald A. Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14092–14101. IEEE, 2020. 8
- [57] A. Radford. Improving language understanding by generative pre-training. 2018. 1
- [58] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1, 5, 11
- [59] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 9, 10, 12
- [60] 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
- [61] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 3, 4, 5, 8, 16, 17, 18, 19
- [62] 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
- [63] Robin Rombach, Patrick Esser, and Björn Ommer. Making sense of cnns: Interpreting deep representations and their invariances with inns. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII*, volume 12362 of *Lecture Notes in Computer Science*, pages 647–664. Springer, 2020. 2
- [64] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2784–2797. Curran Associates, Inc., 2020. 2
- [65] 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
- [66] Edgar Schönfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8204–8213. IEEE, 2020. 8
- [67] Kim Seonghyeon. Implementation of generating diverse high-fidelity images with vq-vae-2 in pytorch, 2020. 8
- [68] 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
- [69] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 8
- [70] 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
- [71] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016. 2, 4, 14
- [72] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 3, 4, 11
- [73] Hugo van Kemenade, wiredfool, Andrew Murray, Alex Clark, Alexander Karpinsky, Ondrej Baranovič, Christoph Gohlke, Jon Dufresne, Brian Crowell, David Schmidt, Konstantin Kopachev, Alastair Houghton, Sandro Mani, Steve Landey, vashek, Josh Ware, Jason Douglas, David Caro, Uriel Martinez, Steve Kossouho, Riley Lahd, Stanislau T., Antony Lee, Eric W. Brown, Oliver Tonnhofer, Mickael Bonfill, Peter Rowlands, Fahad Al-Saidi, German Novikov, and Michał Górný. python-pillow/pillow: 8.2.0, Apr. 2021. 10
- [74] 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
- [75] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 7
- [76] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*. OpenReview.net, 2020. 2
- [77] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models, 2021. 8

- [78] 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](#)
- [79] 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](#)
- [80] 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](#)
- [81] 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](#), [11](#), [13](#)
- [82] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018. [45](#), [46](#)
- [83] 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](#)
- [84] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow, 2017. [2](#)
- [85] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization, 2019. [2](#)