Taming Transformers for High-Resolution Image Synthesis

–

Supplementary Material

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

A. Implementation Details

The hyperparameters for all experiments presented in the main paper and supplementary material can be found in Tab. 7. 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.

VQGAN Architecture The architecture of our convolutional encoder and decoder models used in the *VQGAN* experiments is described in Tab. 6. 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 [51] and we vary its capacity mainly through varying the amount of layers (see Tab. 7). 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).

B. On Context-Rich Vocabularies

Sec. 4.3 investigated the effect of the downsampling factor f used for encoding images. As demonstrated in Fig. 7, large factors are crucial for our approach, since they enable the transformer to model long-range interactions efficiently. However, since larger f correspond to larger compression rates, the reconstruction quality of the *VQGAN* starts to decrease after a certain point, which is analyzed in Fig. 8. The left part shows the reconstruction error (measured by LPIPS [71]) 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

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

Table 6. High-level architecture of the encoder and decoder of our *VQGAN*. The design of the networks follows the architecture presented in [23] with no skip-connections. For the discriminator, we use a patch-based model as in [25]. Note that $h = \frac{H}{2^m}$, $w = \frac{W}{2^m}$ and $f = 2^m$.

Experiment	n_{layer}	# params $[M]$	n_z	$\lvert\mathcal{Z}\rvert$	dropout	length(s)	m
RIN	12	85	64	768	0.0	512	4
c-RIN	18	128	64	768	0.0	257	4
D-RINv1	14	180	256	1024	0.0	512	4
D-RINv2	24	307	256	1024	0.0	512	4
IN	24	307	256	1024	0.0	256	4
c-IN	24	307	256	1024	0.0	257	4
c-IN (big)	48	1400	256	16384	0.0	257	4
IN-Edges	24	307	256	1024	0.0	512	3
IN-SR	12	153	256	1024	0.0	512	3
S-FLCKR, $f = 4$	24	307	256	1024	0.0	512	2
S-FLCKR, $f = 16$	24	307	256	1024	0.0	512	4
S-FLCKR, $f = 32$	24	307	256	1024	0.0	512	5
(FacesHQ, $f = 1)^*$	24	307	–	512	0.0	512	–
FacesHQ, $f = 2$	24	307	256	1024	0.0	512	1
FacesHQ, $f = 4$	24	307	256	1024	0.0	512	2
FacesHQ, $f = 8$	24	307	256	1024	0.0	512	3
FacesHQ**, $f = 16$	24	307	256	1024	0.0	512	4
FFHQ**, $f = 16$	28	355	256	1024	0.0	256	4
CelebA-HQ**, $f = 16$	28	355	256	1024	0.0	256	4
COCO-Stuff	32	651	256	8192	0.0	512	4
ADE20K	28	405	256	4096	0.1	512	4
DeepFashion	18	129	256	1024	0.0	340	4
LSUN-CT	24	307	256	1024	0.0	256	4
CIFAR-10	24	307	256	1024	0.0	256	1

Table 7. Hyperparameters. For every experiment, we set the number of attention heads in the transformer to $n_h = 16$ and the embedding dimension to $n_e = 1024$ (except for c-RIN, D-RINv1 and DeepFashion, which use $n_e = 768$, c-IN (big) ($n_e = 1536$), and COCO-Stuff ($n_e = 1280$)). 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. The "commitment factor" β in Eq. (4) is always set to $\beta = 0.25$. A prefix "c" refers to a class-conditional model. The models marked with a '$**$' are trained on the same *VQGAN*. # params refers to the number of parameters of the transformer model; we round 307 M to 310 M in the main text.

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. 8 shows that we must learn perceptually rich encodings, *i.e.* encodings with a large f and perceptually faithful reconstructions. This is the goal of our *VQGAN* and Fig. 9 compares its reconstruction capabilities against a *VQVAE* [63]. Both approaches use the same architecture of Tab. 6 and a factor $f = 16$, which demonstrates how the *VQGAN* provides high-fidelity reconstructions at large factors, and thereby enables efficient high-resolution image synthesis with transformers.

To illustrate how the choice of f depends on the dataset, Fig. 10 presents results on S-FLCKR. In the left part, it shows, analogous to Fig. 7, how the quality of samples increases with increasing f. However, in the right part, it shows that reconstructions remain faithful perceptually faithful even for $f32$, which is in contrast to the corresponding results on faces in Fig. 8. These results might be explained by a higher perceptual sensitivity to facial features as compared to textures, and allow us to generate high-resolution landscapes even more efficiently with $f = 32$.

C. Additional Results

Qualitative Comparisons The qualitative comparison corresponding to Tab. 4 of the main paper can be found in Fig. 11, 12, 13 and 14. Since no models are available for VQVAE-2 and MSP, we extracted results directly from the supplementary[1] and from the provided samples[2], respectively. For BigGAN, we produced the samples via the provided model[3]. Similarly, the qualitative comparison with the best competitor model (SPADE) for semantic synthesis on standard

[1] https://drive.google.com/file/d/1H2nr_Cu7OK18tRemsWn_6o5DGMNYentM/view?usp=sharing
[2] https://bit.ly/2FJkvhJ
[3] https://tfhub.dev/deepmind/biggan-deep-256/1

benchmarks (see Tab. 2) can be found in Fig. 28 (ADE20K) and Fig. 29 (COCO-Stuff)[4].

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. 15 and Fig. 16, 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. 17, 18, 19 and Fig. 20 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. 21 and Fig. 22, edge-to-image on IN in Fig. 23, stochastic superresolution on IN in Fig. 24, more examples on semantically guided landscape synthesis on S-FLCKR in Fig. 25 with $f = 16$ and in Fig. 26 with $f = 32$, and unconditional image generation on LSUN-CT (see Sec. 4.1) in Fig. 27. Moreover, for images of size 256×256, we provide results for generation from semantic layout on (i) ADE20K in Fig. 28 and (ii) COCO-Stuff in Fig. 29, depth-to-image on IN in Fig. 30, pose-guided person generation in Fig. 31 and class-conditional synthesis on RIN and IN in Fig. 32 and Fig. 33, respectively.

D. On the Ordering of Image Representations

For the "classical" domain of transformer models, NLP, the order of tokens is defined by the language at hand. For images and their discrete representations, in contrast, it is not clear which linear ordering to use. In particular, our sliding-window approach depends on a row-major ordering and we thus investigate the performance of the following five different permutations of the input sequence of codebook indices: (i) **row major**, or *raster scan order*, where the image representation is unrolled from top left to bottom right. (ii) **spiral out**, which incorporates the prior assumption that most images show a *centered* object. (iii) **z-curve**, also known as *z-order* or *morton curve*, which introduces the prior of *preserved locality* when mapping a 2D image representation onto a 1D sequence. (iv) **subsample**, where prefixes correspond to subsampled representations, see also [42]. (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. 34. 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.34 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. 35 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 [62, 8] outperforms other orderings.

[4] samples were reproduced with the authors' official implementation available at https://github.com/nvlabs/spade/

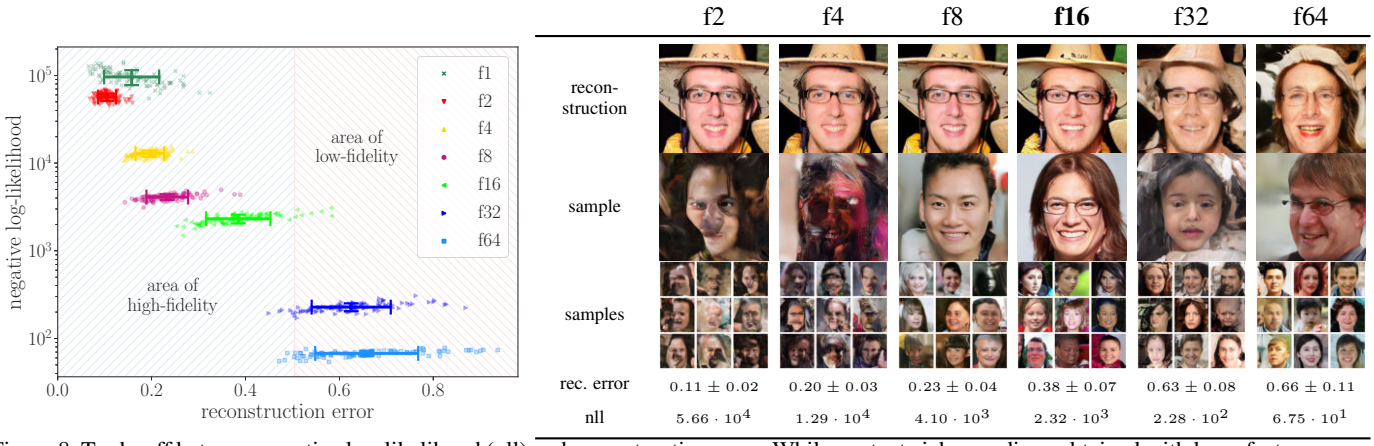

Figure 8. Trade-off between negative log-likelihood (nll) and reconstruction error. While context-rich encodings obtained with large factors f allow the transformer to effectively model long-range interactions, the reconstructions capabilities and hence quality of samples suffer after a critical value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

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

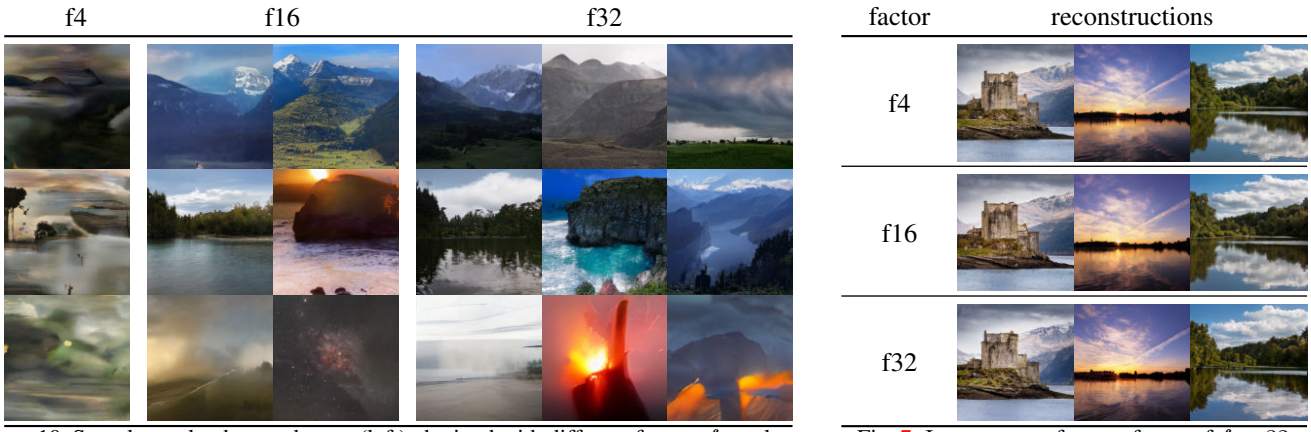

Figure 10. Samples on landscape dataset (left) obtained with different factors f, analogous to Fig. 7. In contrast to faces, a factor of $f = 32$ still allows for faithful reconstructions (right). See also Sec. B.

Figure 11. 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 [53].

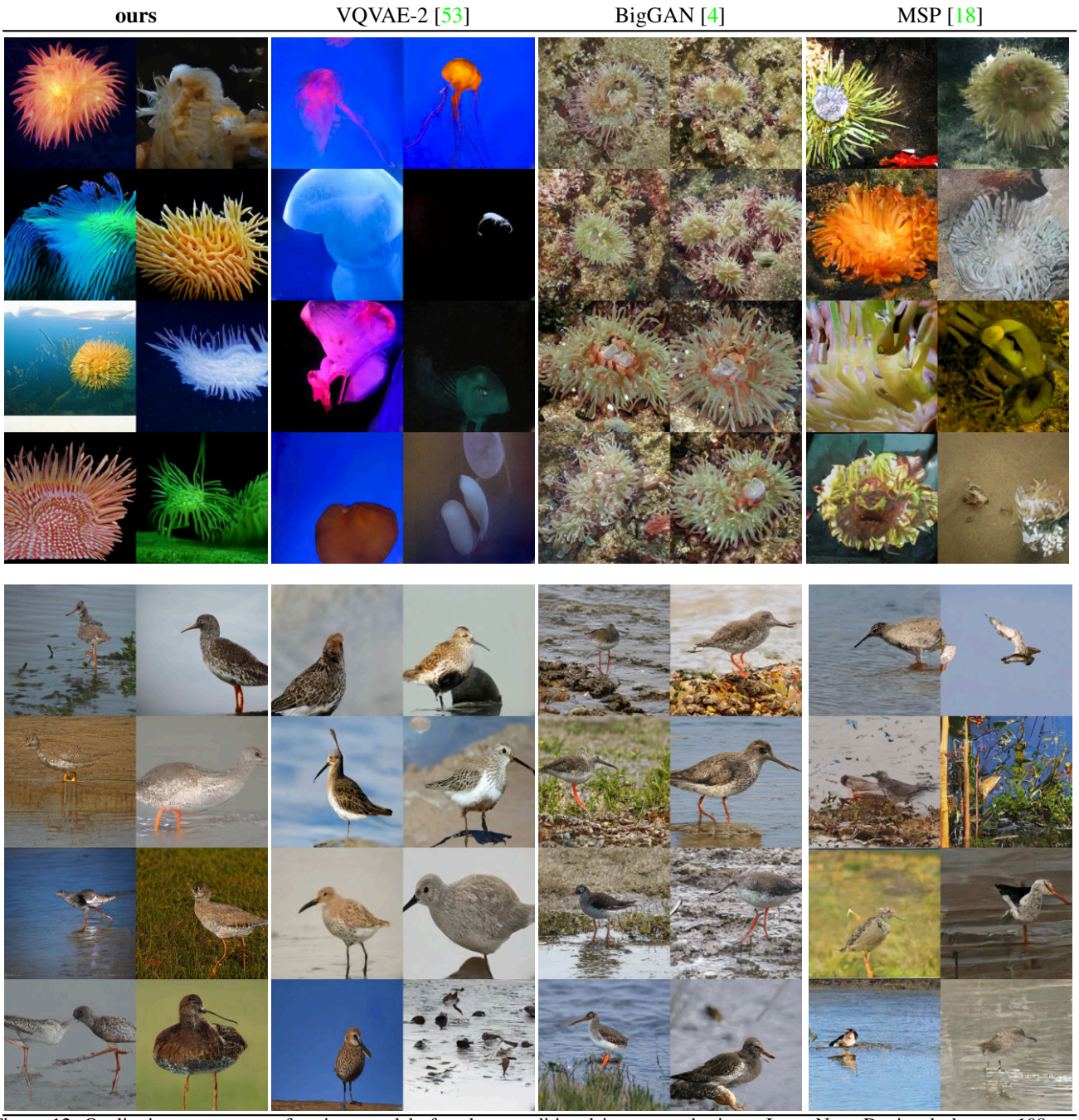

Figure 12. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: *108: sea anemone* (top) and *141: redshank* (bottom). We report class labels as in VQVAE-2 [53].

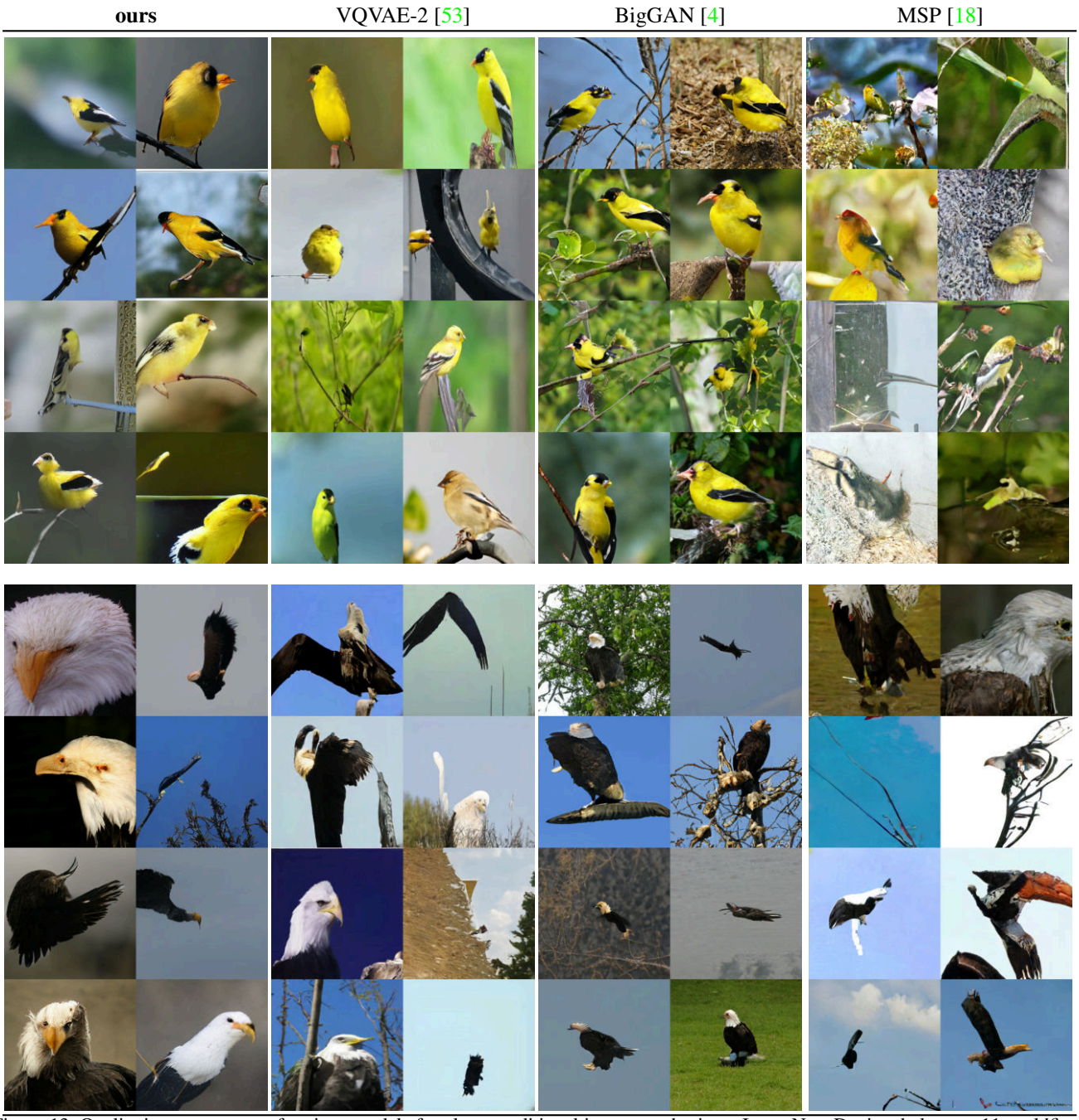

Figure 13. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: *11: goldfinch* (top) and *22: bald eagle* (bottom).

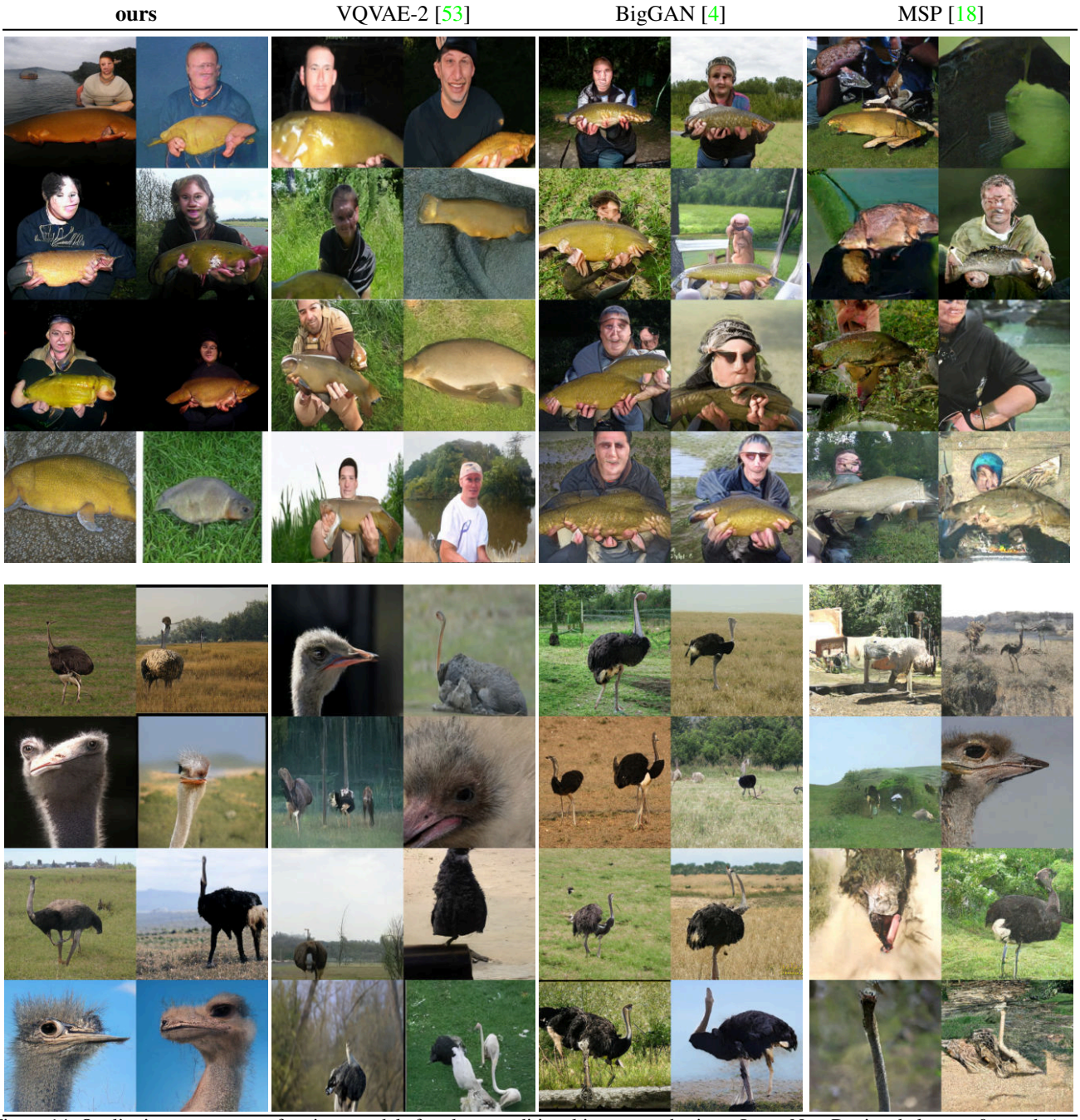

Figure 14. Qualitative assessment of various models for class-conditional image synthesis on ImageNet. Depicted classes: *0: tench* (top) and *9: ostrich* (bottom).

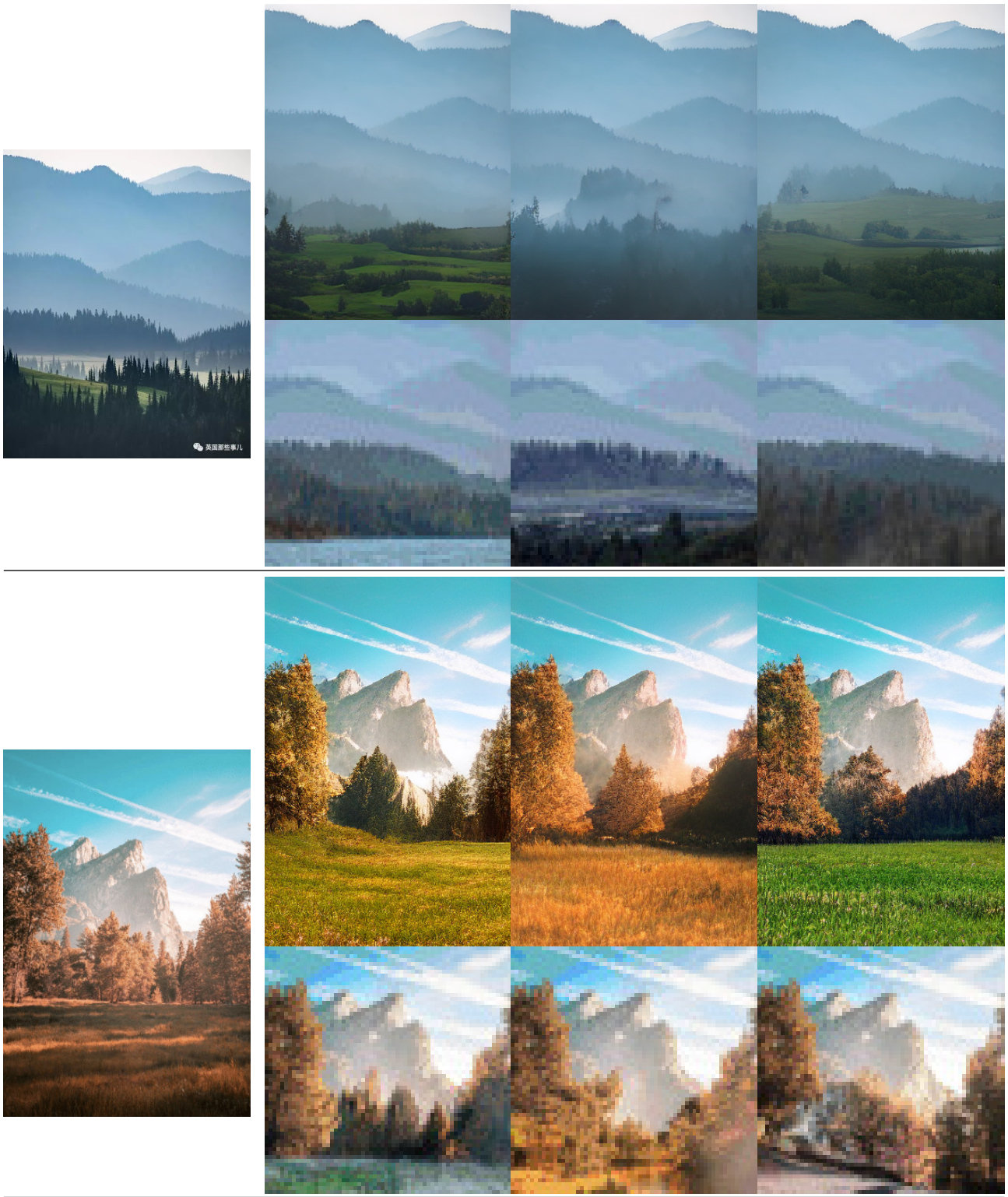

Figure 15. Comparing our approach with the pixel-based approach of [8]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [8] (bottom).

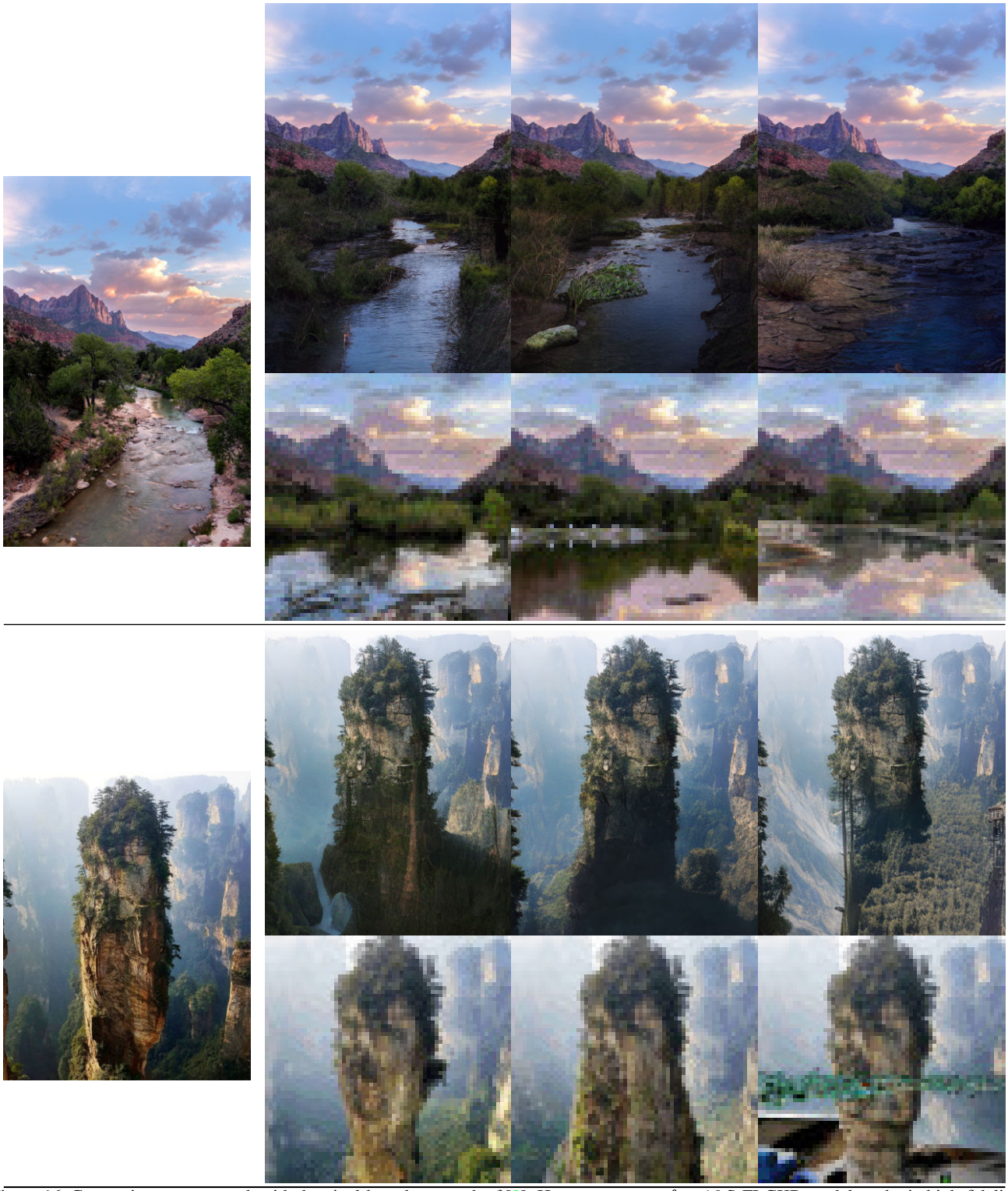

Figure 16. Comparing our approach with the pixel-based approach of [8]. Here, we use our $f = 16$ S-FLCKR model to obtain high-fidelity image completions of the inputs depicted on the left (half completions). For each conditioning, we show three of our samples (top) and three of [8] (bottom).

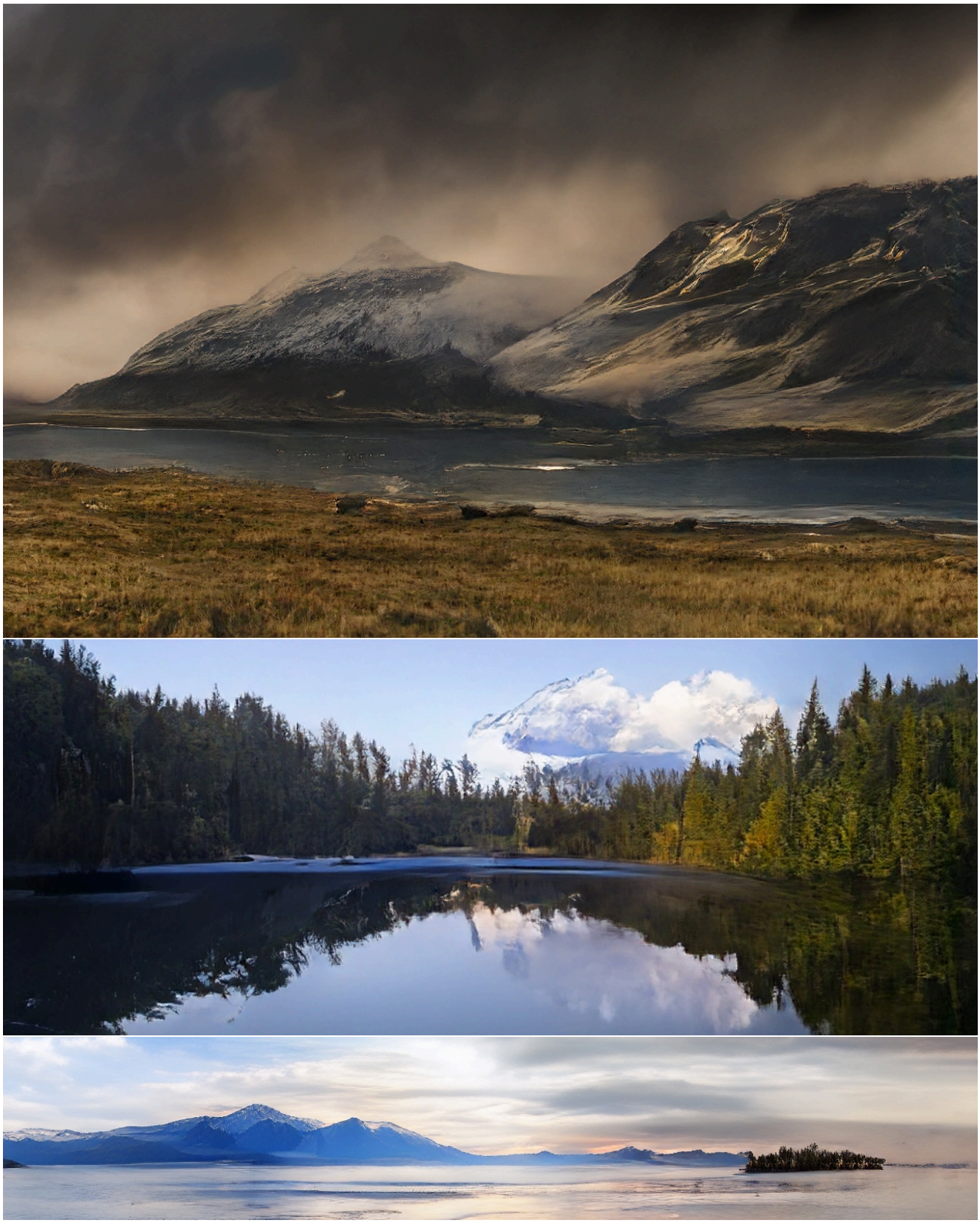

Figure 17. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832, 1024×416 and 1280×240 pixels.

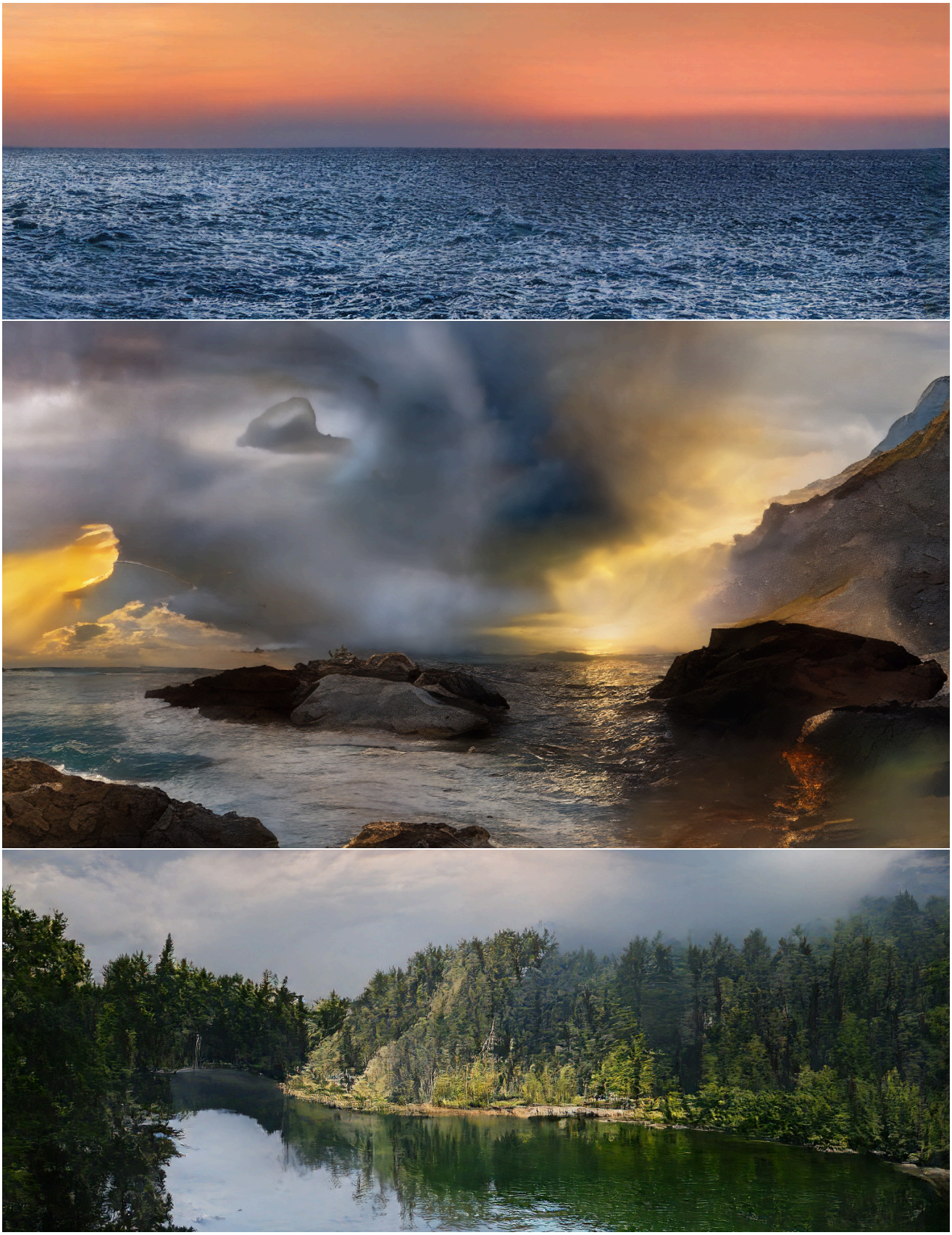

Figure 18. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1536×512, 1840×1024, and 1536×620 pixels.

Figure 19. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 2048×512, 1460×440, 2032×448 and 2016×672 pixels.

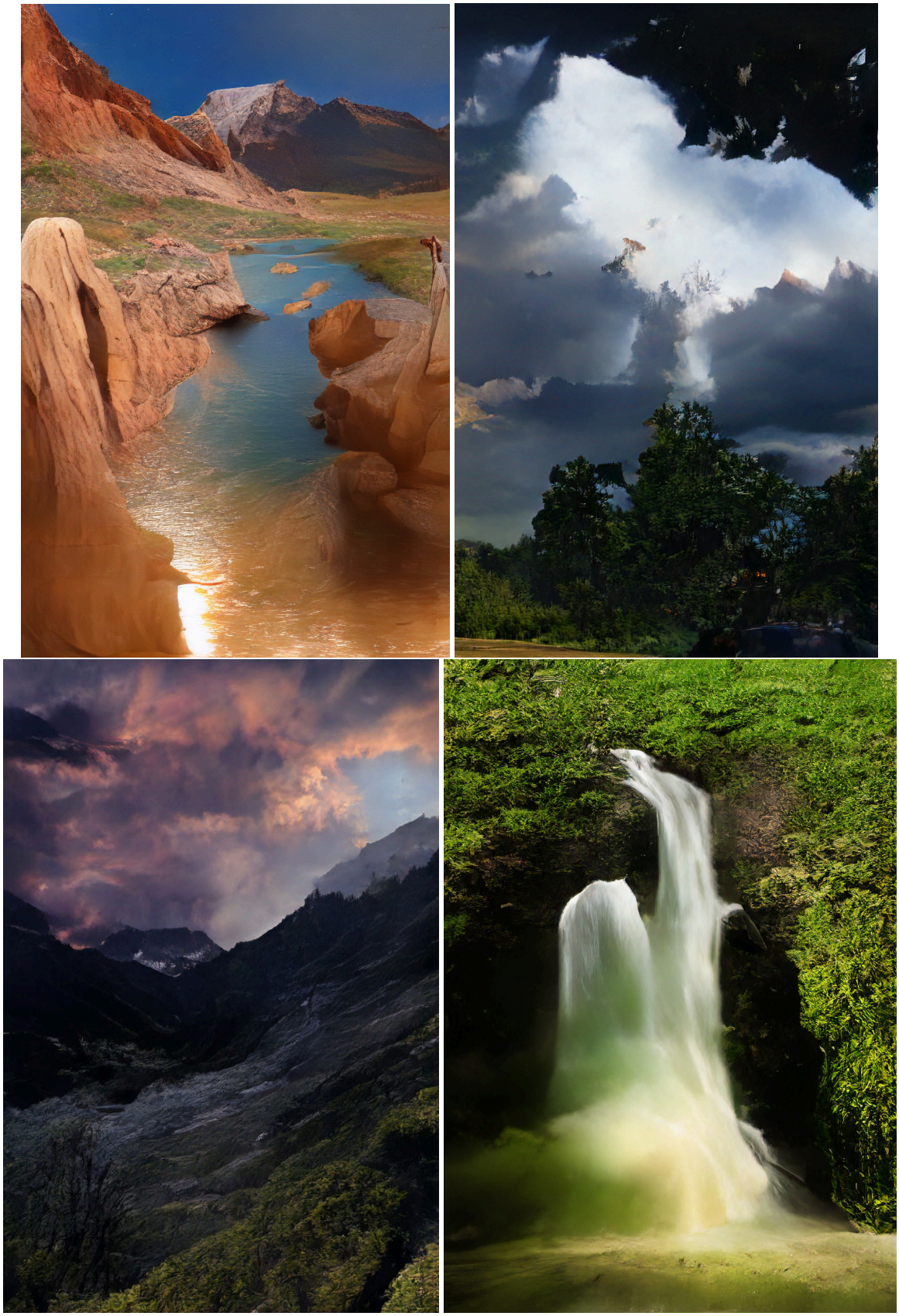

Figure 20. Samples generated from semantic layouts on S-FLCKR. Sizes from top-to-bottom: 1280×832, 1024×416 and 1280×240 pixels.

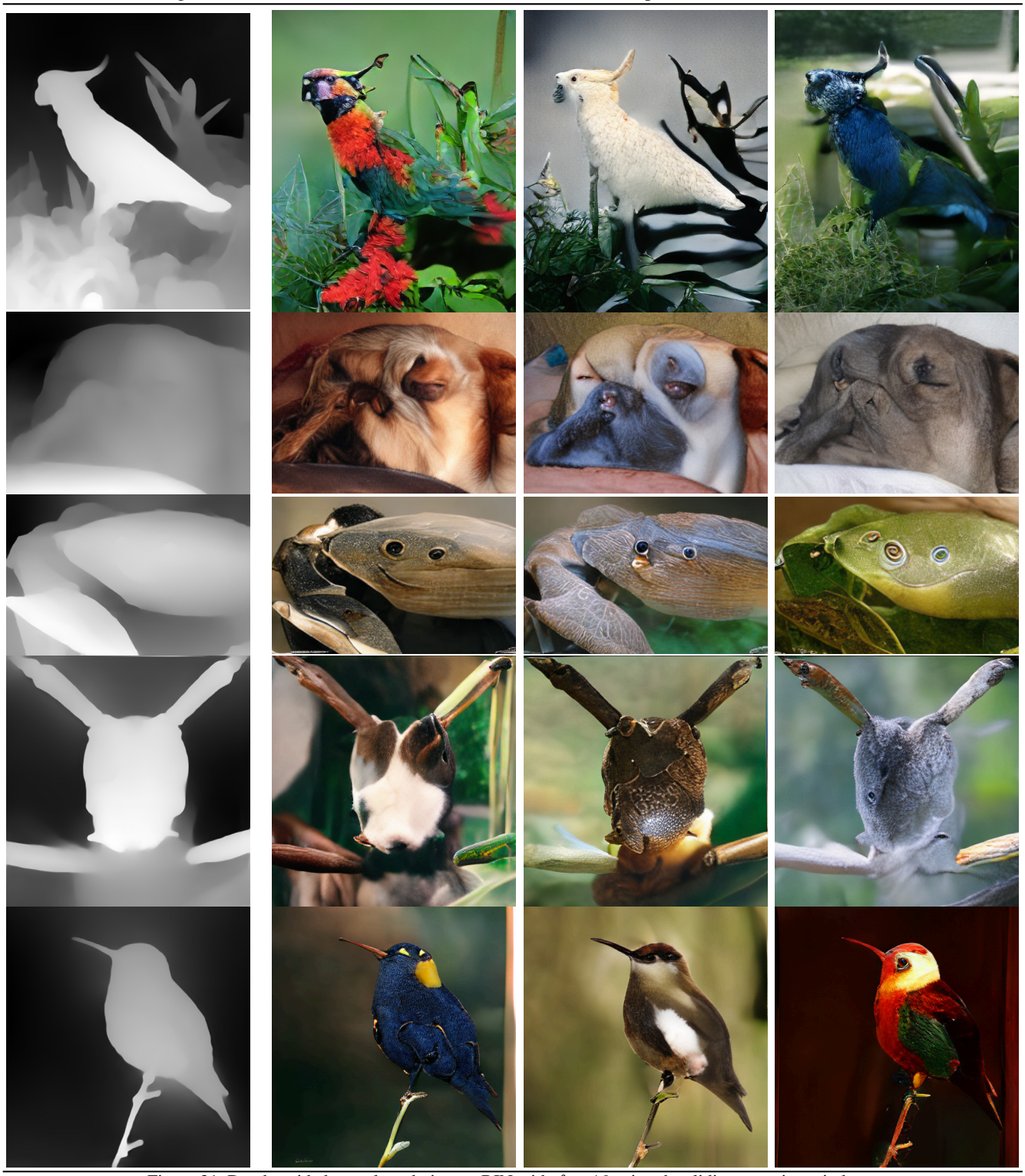

Figure 21. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.

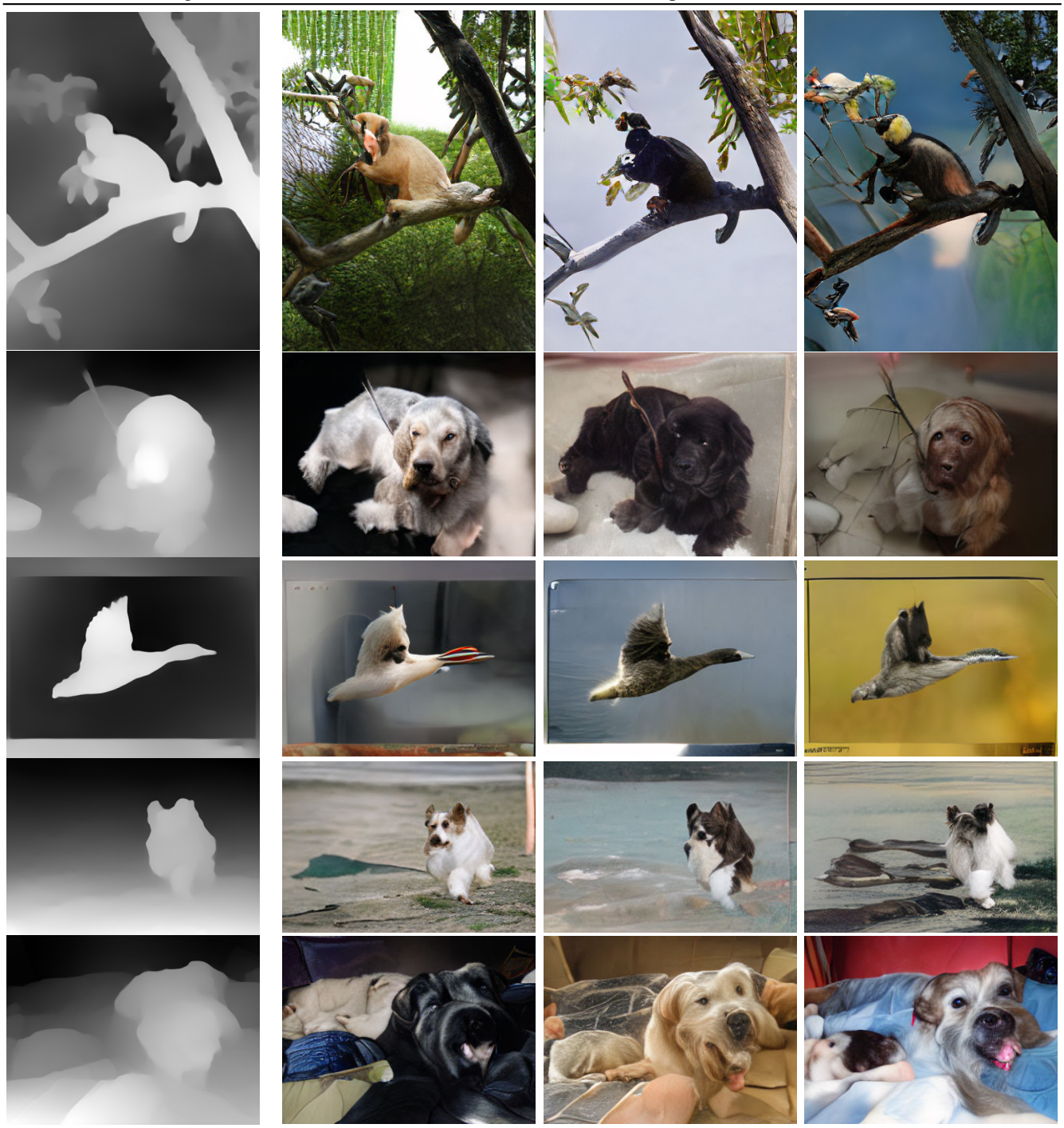

Figure 22. Depth-guided neural rendering on RIN with $f = 16$ using the sliding attention window.

Figure 23. 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.

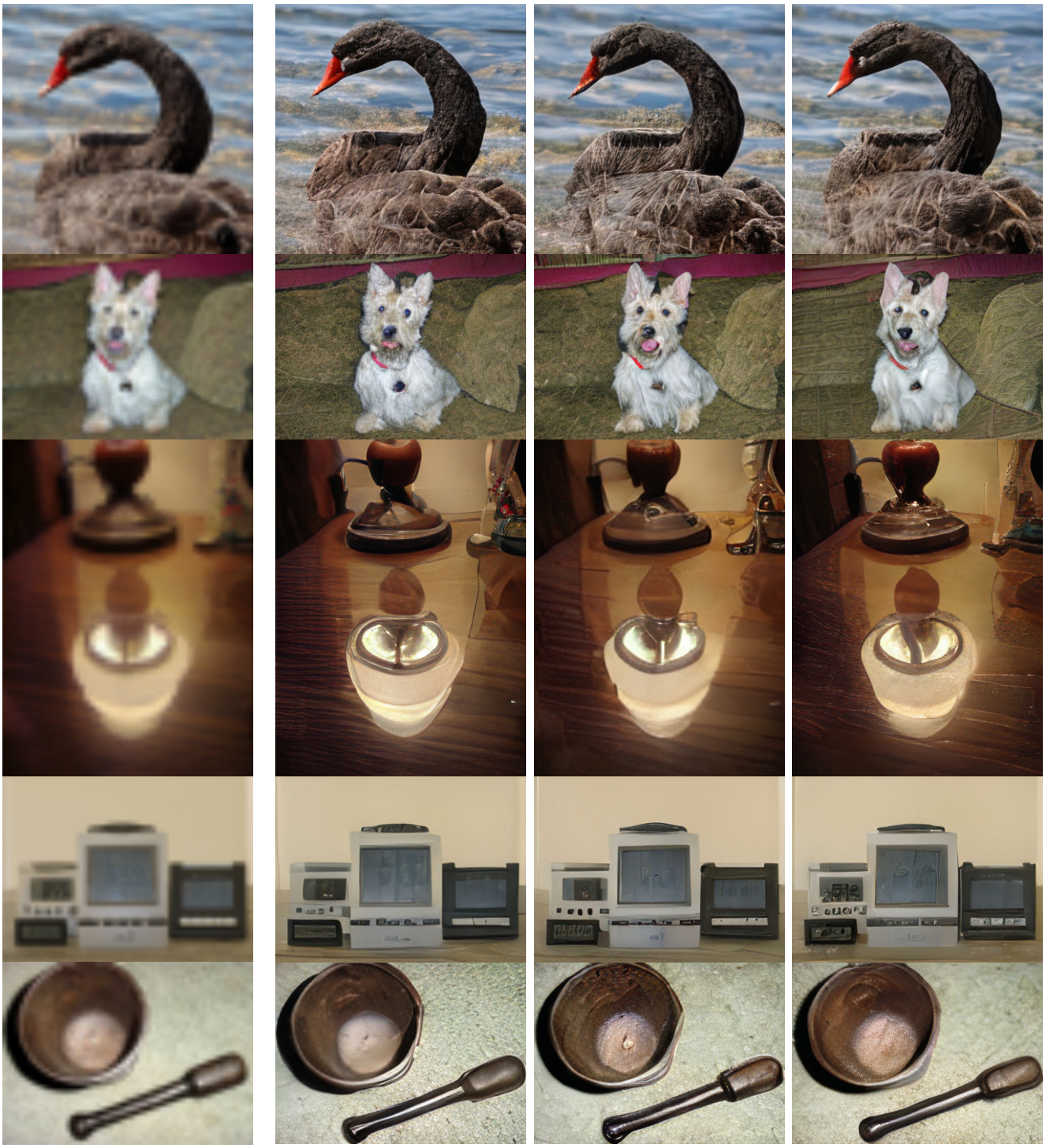

Figure 24. Additional results for stochastic superresolution with an $f = 16$ model on IN, using the sliding attention window.

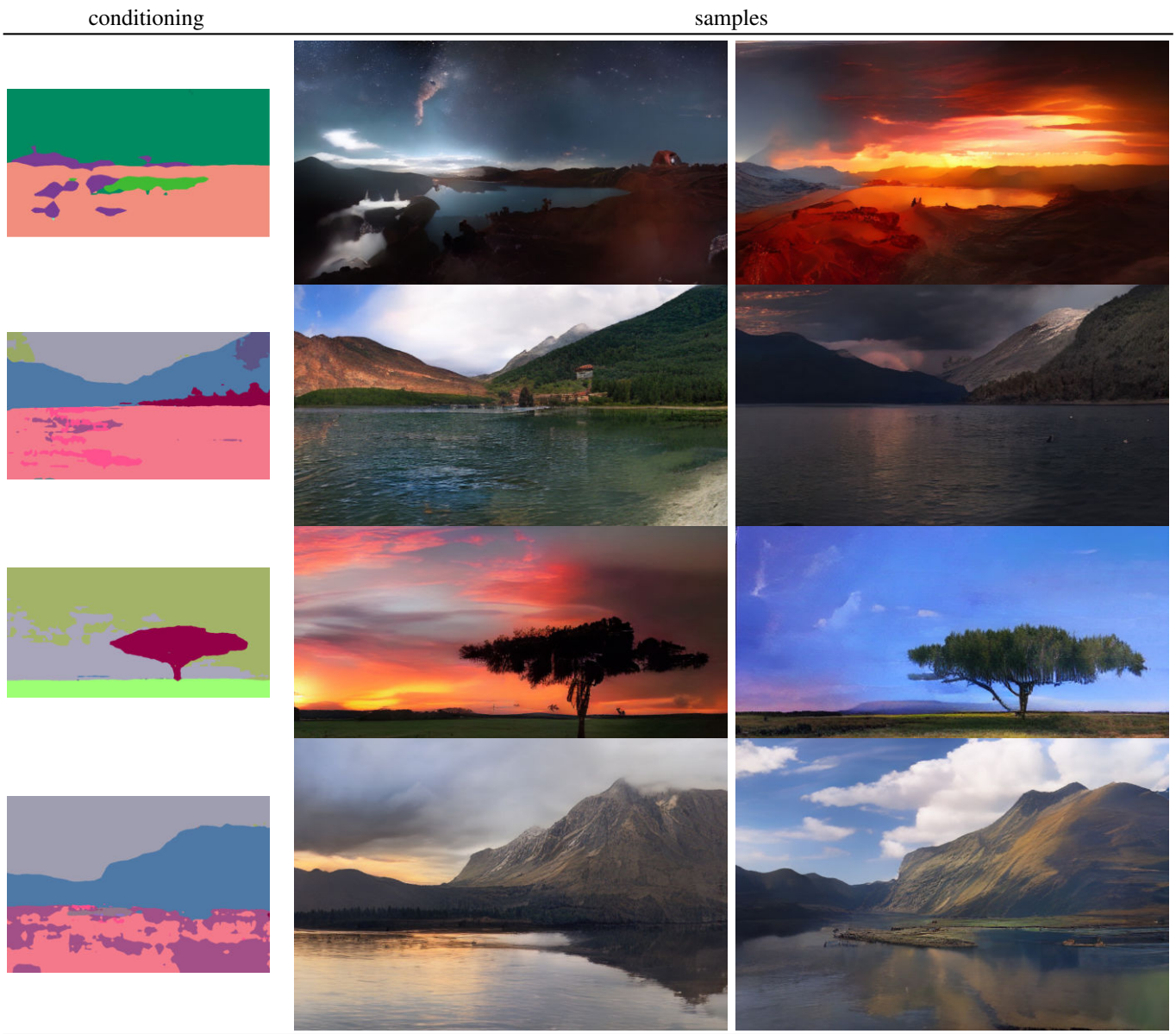

Figure 25. Samples generated from semantic layouts on S-FLCKR with $f = 16$, using the sliding attention window.

Figure 26. Samples generated from semantic layouts on S-FLCKR with $f = 32$, using the sliding attention window.

Figure 27. Unconditional samples from a model trained on LSUN Churches & Towers, using the sliding attention window.

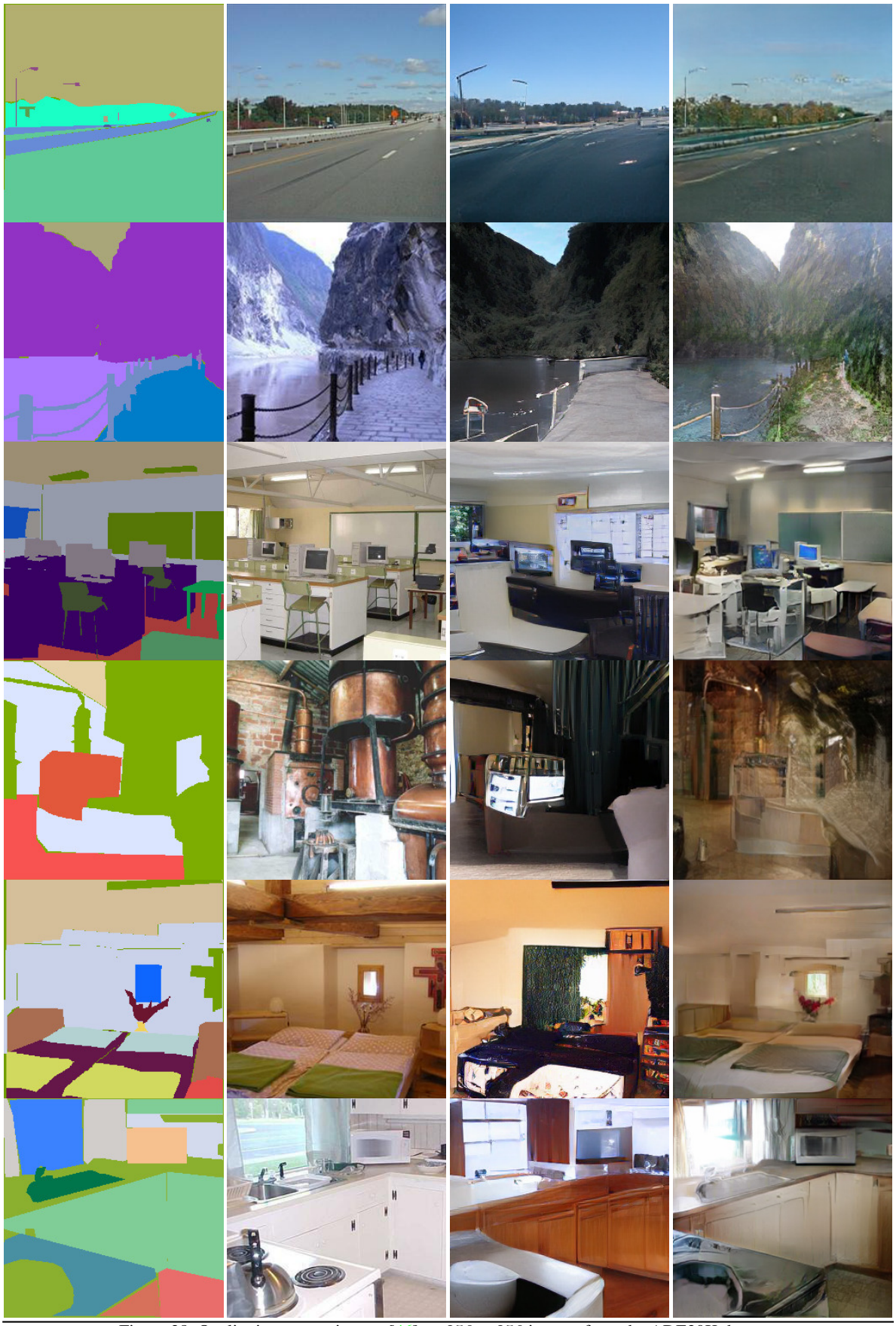

Figure 28. Qualitative comparison to [46] on 256 × 256 images from the ADE20K dataset.

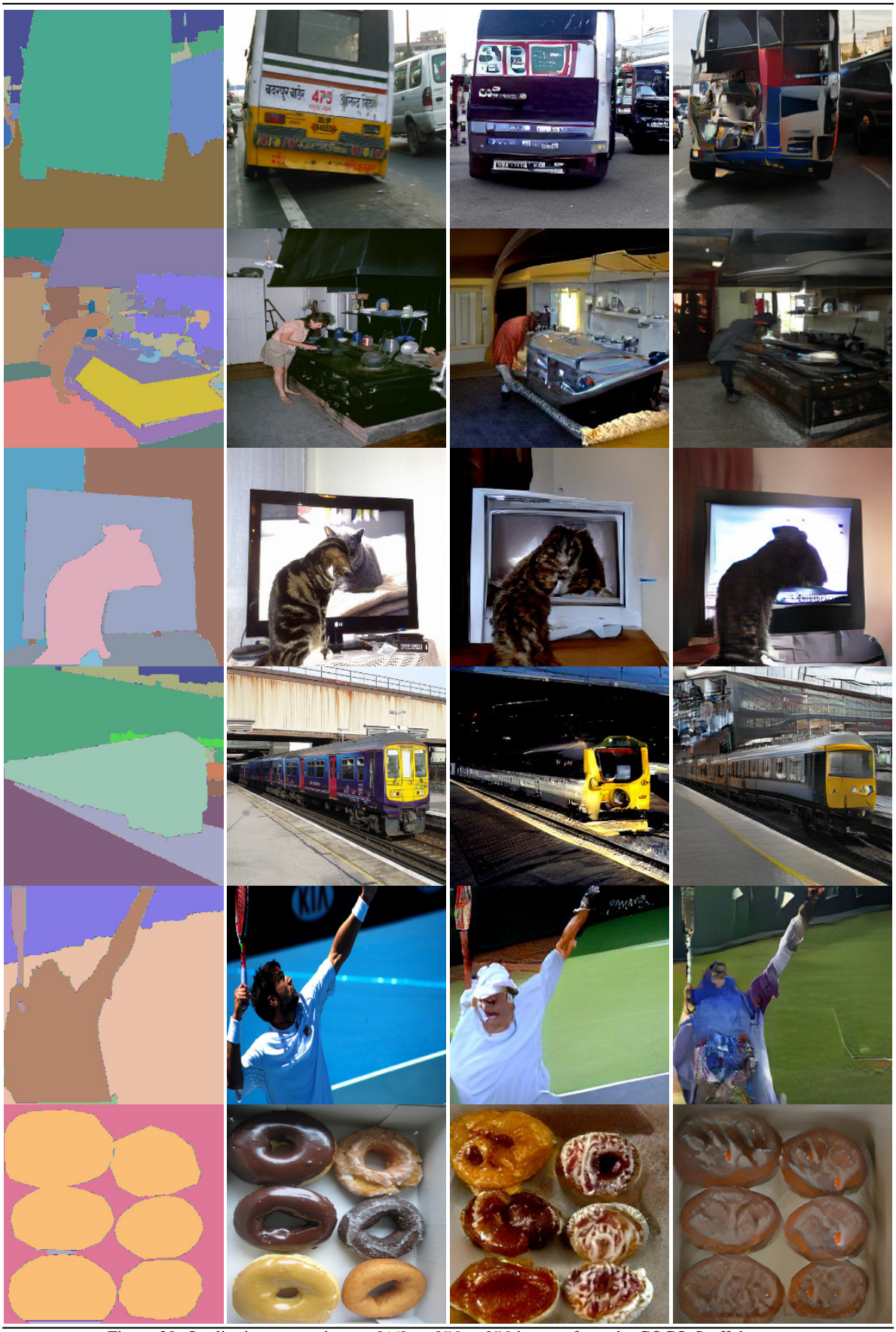

Figure 29. Qualitative comparison to [46] on 256 × 256 images from the COCO-Stuff dataset.

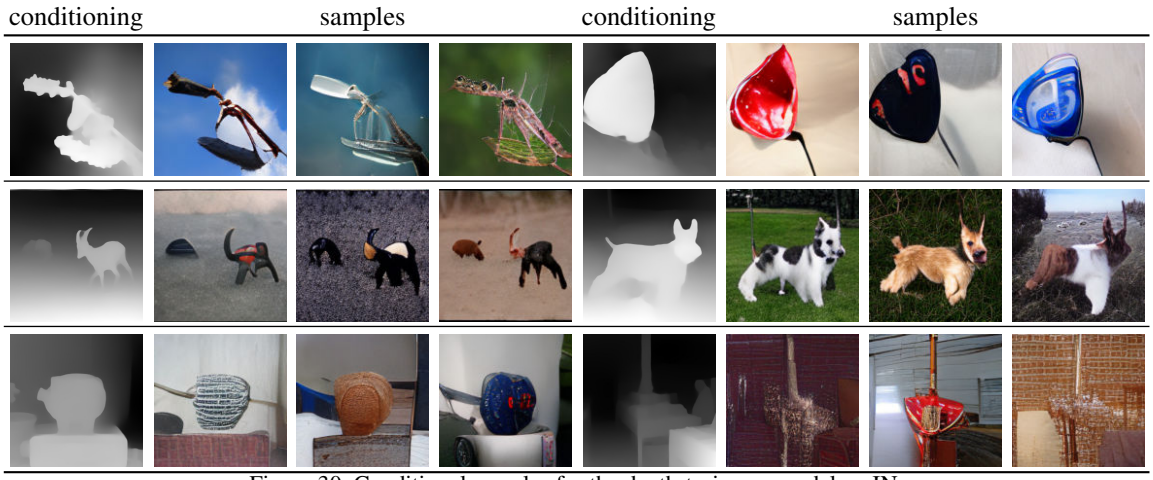

Figure 30. Conditional samples for the depth-to-image model on IN.

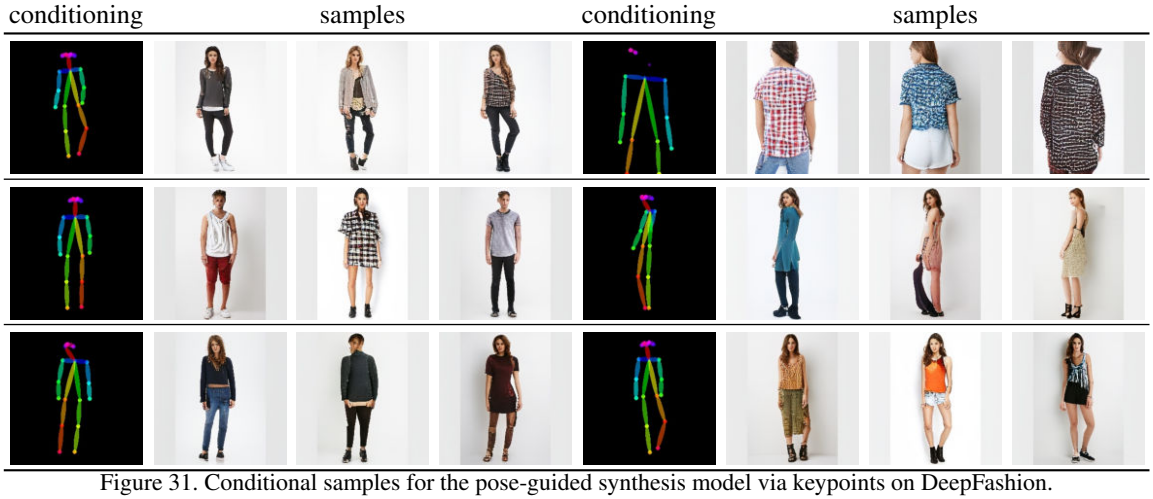

Figure 31. Conditional samples for the pose-guided synthesis model via keypoints on DeepFashion.

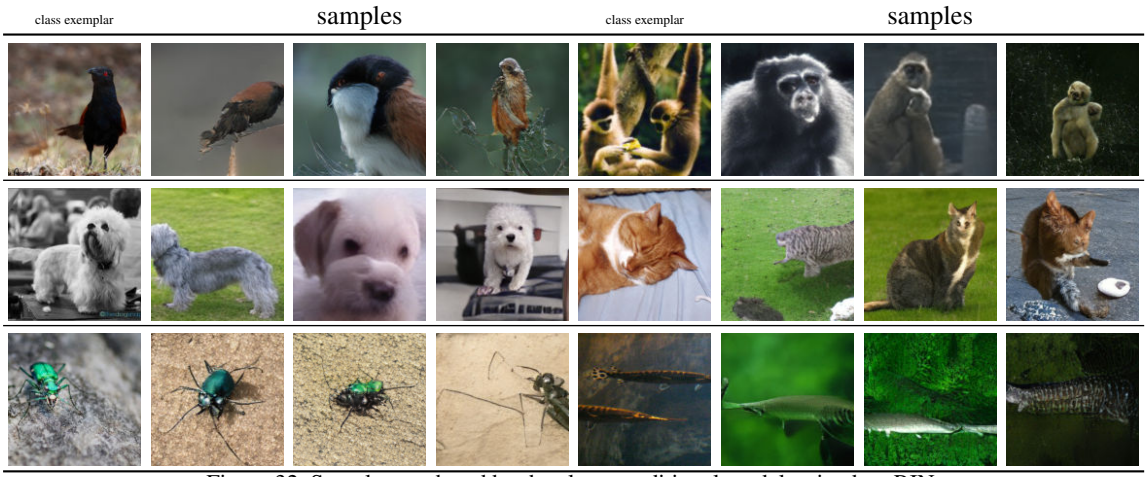

Figure 32. Samples produced by the class-conditional model trained on RIN.

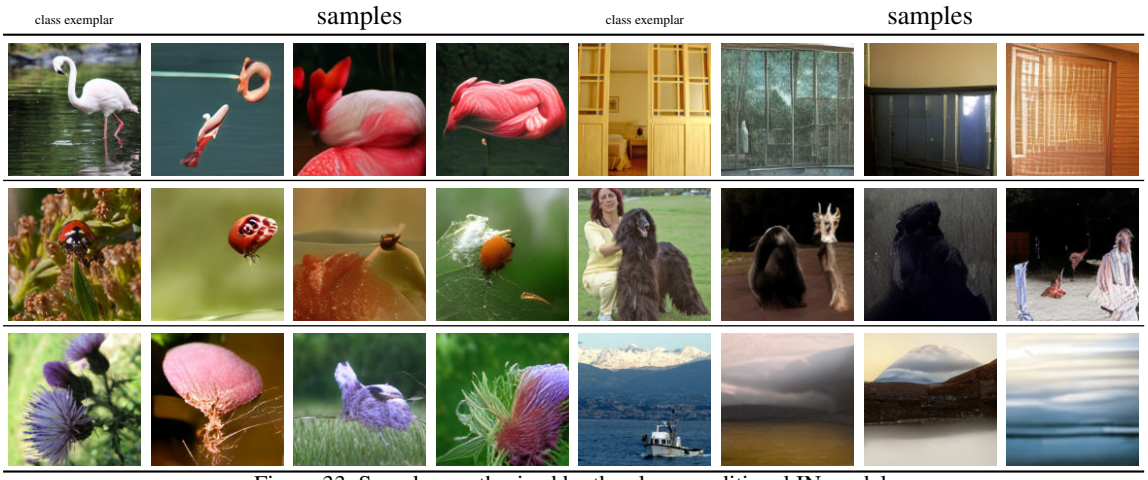

Figure 33. Samples synthesized by the class-conditional IN model.

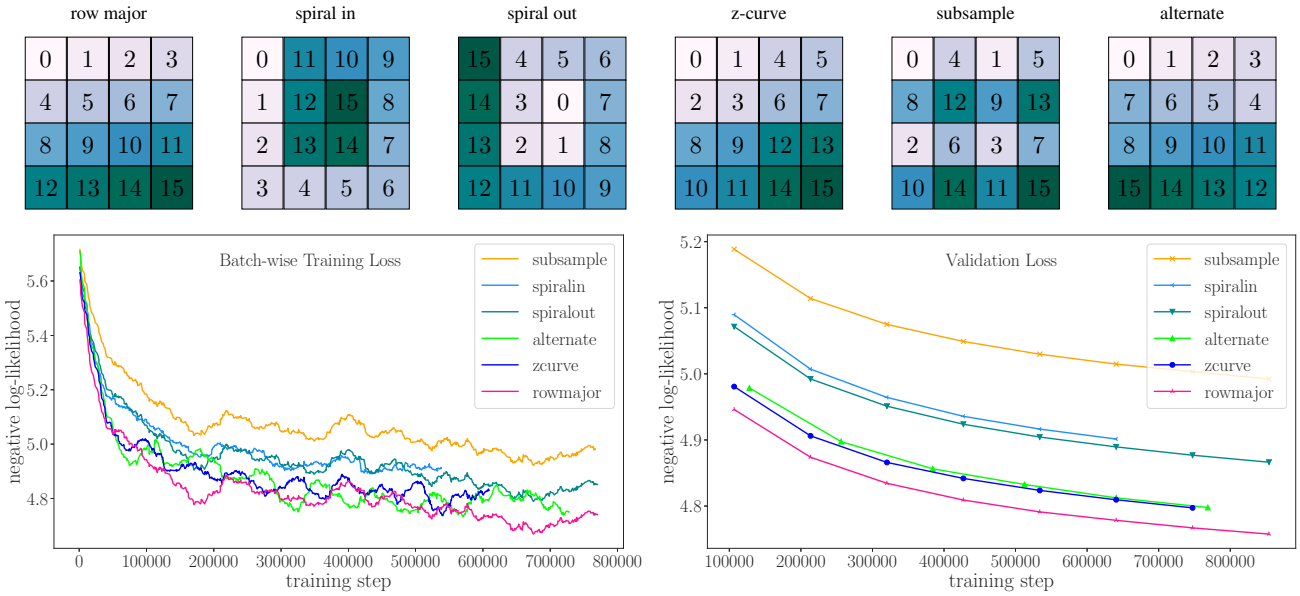

Figure 34. Top: All sequence permutations we investigate, illustrated on a 4×4 grid. Bottom: The transformer architecture is permutation invariant but next-token prediction is not: The average loss on the validation split of ImageNet, corresponding to the negative log-likelihood, differs significantly between different prediction orderings. Among our choices, the commonly used row-major order performs best.

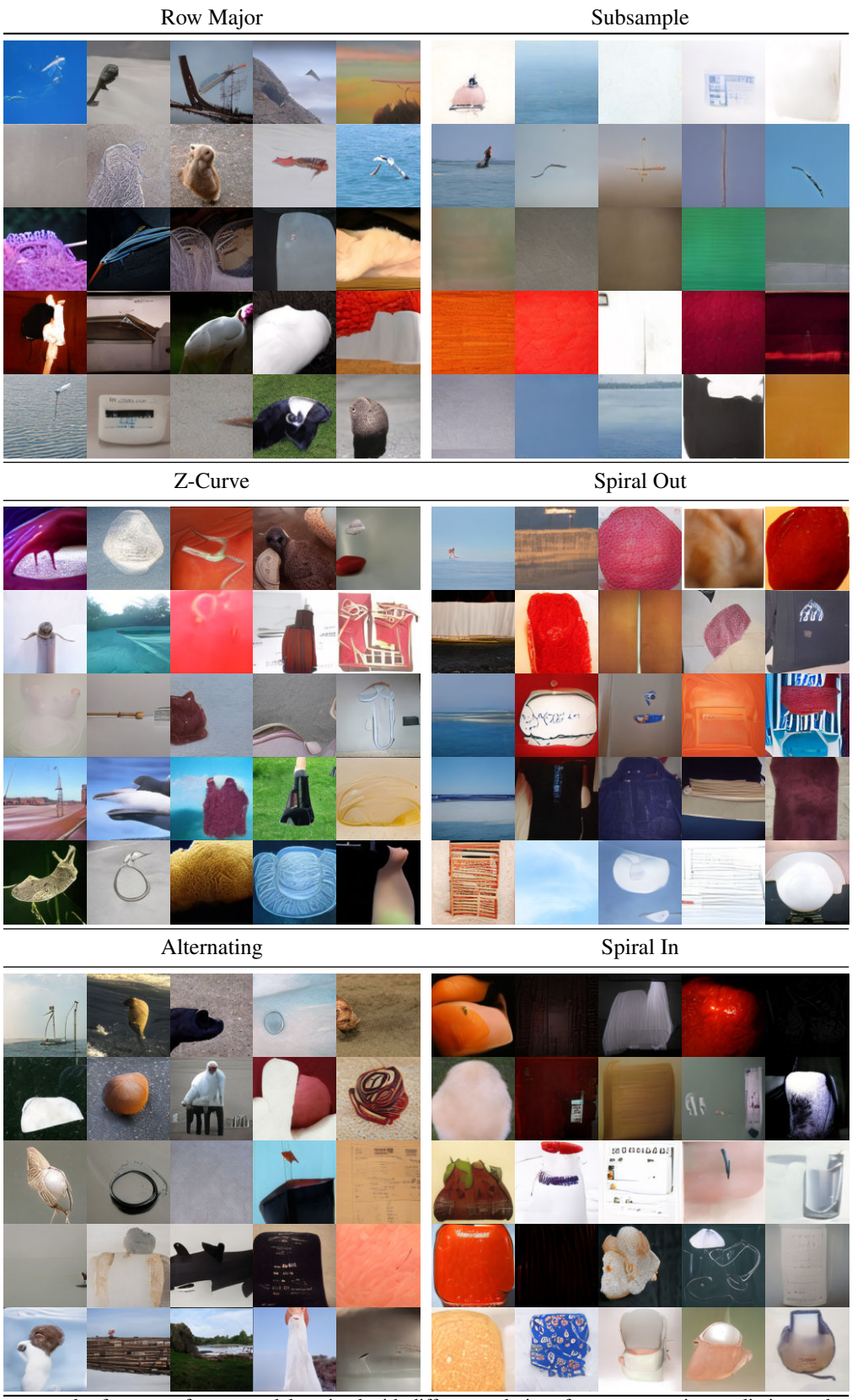

Figure 35. Random samples from transformer models trained with different orderings for autoregressive prediction as described in Sec. D.