Table A.1. The hyperparameters for implementing RQ-Transformer. We follow the same notation in the main paper. n_e represents the dimensionality of features in RQ-Transformer, and # heads represents the number of heads in self-attentions of RQ-Transformer.

Dataset	N_{spatial}	N_{depth}	# params	K	T	D	n_z	n_e	# heads	τ
LSUN-cat [12]	26	4	612M	16384	64	4	256	1280	20	0.5
LSUN-bedroom [12]	26	4	612M	16384	64	4	256	1280	20	0.5
LSUN-church [12]	24	4	370M	16384	64	4	256	1024	16	0.5
FFHQ [6]	24	4	370M	2048	64	4	256	1024	16	1.0
ImageNet [2]	12	4	480M	16384	12	4	256	1536	24	0.5
ImageNet [2]	24	4	821M	16384	64	4	256	1536	24	0.5
ImageNet [2]	42	6	1388M	16384	64	4	256	1536	24	0.5
ImageNet [2]	42	6	3822M	16384	64	4	256	2560	40	0.5
CC-3M [9]	24	4	654M	16384	95	4	256	1280	20	0.5

A. Implementation Details

A.1. Architecture of RQ-VAE

For the architecture of RQ-VAE, we follow the architecture of VQ-GAN [4] for a fair comparison. However, we add two residual blocks with 512 channels each followed by a down-/up-sampling block to extract feature maps of resolution 8×8.

A.2. Architecture of RQ-Transformer

The RQ-Transformer, which consists of the spatial transformer and the depth transformer, adopts a stack of self-attention blocks [11] for each compartment. In Table A.1, we include the detailed information of hyperparameters to implement our RQ-Transformers. All RQ-Transformers in Table A.1 uses RQ-VAE with 8×8×4 shape of codes. For CC-3M, the length of text conditions is 32, and the last token in text conditions predicts the code at the first position of images. Thus, the total sequence length (T) of RQ-Transformer is 95.

A.3. Training Details

For ImageNet, RQ-VAE is trained for 10 epochs with batch size 128. We use the Adam optimizer [7] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$, and learning rate is set 0.00004. The learning rate is linearly warmed up during the first 0.5 epoch. We do not use learning rate decay, weight decaying, nor dropout. For the adversarial and perceptual loss, we follow the experimental setting of VQ-GAN [4]. In particular, the weight for the adversarial loss is set 0.75 and the weight for the perceptual loss is set 1.0. To increase the codebook usage of RQ-VAE, we use random restart of unused codes proposed in JukeBox [3]. For LSUN-{cat, bedroom, church}, we use the pretrained RQ-VAE on ImageNet and finetune it for one epoch with 0.000004 of learning rate. For FFHQ, we train RQ-VAE for 150 epochs of training data with 0.00004 of learning rate and five epochs of warm-up. For CC-3M, we use the pretrained RQ-VAE on ImageNet without finetuning.

All RQ-Transformers are trained using the AdamW optimizer [8] with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We use the cosine learning rate schedule with 0.0005 of the initial learning rate. The RQ-Transformer is trained for 90, 200, 300 epochs for LSUN-bedroom, -cat, and -church respectively. The weight decay is set 0.0001, and the batch size is 16 for FFHQ and 2048 for other datasets. In all experiments, the dropout rate of each self-attention block is set 0.1 except 0.3 for 3.8B parameters of RQ-Transformer. We use eight NVIDIA A100 GPUs to train RQ-Transformer of 1.4B parameters, and four GPUs to train RQ-Transformers of other sizes. The training time is <9 days for LSUN-cat, LSUN-bedroom, <4.5 days for ImageNet, and CC-3M, and <1 day for LSUN-church and FFHQ. We use the early stopping at 39 epoch for the FFHQ dataset, considering the overfitting of the RQ-Transformer due to the small scale of the dataset.

B. Additional Results of Generated Images by RQ-Transformer

B.1. Additional Examples of Unconditional Image Generation for LSUNs and FFHQ

We show the additional examples of unconditional image generation by RQ-VAEs trained on LSUN-{cat, bedroom, church} and FFHQ. Figure A.1, A.2, A.3, and A.4 show the results of LSUN-cat, LSUN-bedroom LSUN-church, and FFHQ, respectively. For the top-k (top-p) sampling, 512 (0.9), 8192 (0.85), 1400 (1.0), and 2048 (0.95) are used respectively.

Table A.2. Results of coarse-to-fine approximation by the RQ-VAE with $8 \times 8 \times 4$ shape of $\mathbf{M}$. Reconstruction loss $\mathcal{L}_{\text{recon}}$, commitment loss $\mathcal{L}_{\text{commit}}$, perceptual loss, and reconstruction FID (rFID) are measured on ImageNet validation data.

$\hat{\mathbf{X}}$	$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{commit}}$	Perceptual loss	rFID
$G(\hat{\mathbf{Z}}^{(1)})$	0.018	0.12	0.12	100.86
$G(\hat{\mathbf{Z}}^{(2)})$	0.014	0.10	0.090	22.74
$G(\hat{\mathbf{Z}}^{(3)})$	0.012	0.091	0.075	7.66
$G(\hat{\mathbf{Z}}^{(4)})$	0.010	0.082	0.068	4.73

B.2. Nearest Neighbor Search of Generated Images for FFHQ

For the training of FFHQ, we use early stopping for RQ-Transformer when the validation loss is minimized, since RQ-Transformer can memorize all training samples due to the small scale of FFHQ. Despite the use of early stopping, we further examine whether our model memorizes the training samples or generates new images. To visualize the nearest neighbors in the training images of FFHQ to generated images, we use a KD-tree [1], which is constructed by the VGG-16 features [10] of training images. Figure A.5 shows that our model does not memorize the training data, but generates new face images for unconditional sample generation of FFHQ.

B.3. Ablation Study on Soft Labeling and Stochastic Sampling

For 821M parameters of RQ-Transformer trained on ImageNet, RQ-Transformer achieves 14.06 of FID score when neither stochastic sampling nor soft labeling is used. When stochastic sampling is applied to the training of RQ-Transformer, 13.24 of FID score is achieved. When only soft labeling is used without stochastic sampling, RQ-Transformer achieves 14.87 of FID score, and the performance worsens. However, when both stochastic sampling and soft labeling are used together, RQ-Transformer achieves 13.11 of FID score, which is improved performance than baseline.

B.4. Additional Examples of Class-Conditioned Image Generation for ImageNet

We visualize the additional examples of class-conditional image generation by RQ-Transformer trained on ImageNet. Figure A.6 and A.7 show the generated samples by RQ-Transformer with 1.4B parameters conditioned on a few selected classes. Those images are sampled with top-k 512 and top-p 0.95.The (top-k, acceptance rate)s are (512, 0,5), (1024, 0.25), and (2048, 0.05), and their corresponding FID scores are 7.08, 5.62, and 4.45. Figure A.8 shows the generated samples of RQ-Transformer with 3.8B parameters using rejection sampling with (4098, 0.125).

B.5. Additional Examples of Text-Conditioned Image Generation for CC-3M

We visualize the additional examples of text-conditioned image generation by RQ-Transformer trained on CC-3M. Figure A.9 shows the generated samples conditioned by various texts, which are unseen during training. Specifically, we manually choose four pairs of sentences, which share visual content with different contexts and styles, to validate the compositional generalization of our model. All images are sampled with top-k 1024 and top-p 0.9.

B.6. The Effects of Top-k & Top-p Sampling on FID Scores

In this section, we show the FID scores of the RQ-Transformer trained on ImageNet according to the choice of k and p for top-k and top-p sampling, respectively. Figure A.10 and A.11 shows the FID scores of 821M and 1400M parameters of RQ-Transformer according to different ks and ps. Although we report the global minimum FID score, the minimum FID score at each k is not significantly deviating from the global minimum. For instance, the minimum FID attained by RQ-Transformer with 1.4B parameters is 11.58 while the minimum for each k is at most 11.87. When the rejection sampling of generated images is used to select high-quality images, Figure A.12 shows that higher top-k values are effective as the acceptance rate decreases, since various and high-quality samples can be generated with higher top-k values. Finally, for the CC-3M dataset, Figure A.13 shows the FID scores and CLIP similarity scores according to different top-k and top-p values.

C. Additional Results of Reconstruction Images by RQ-VAE

C.1. Coarse-to-Fine Approximation of Feature Maps by RQ-VAE

In this section, we further explain that RQ-VAE with depth D conducts the coarse-to-fine approximation of a feature map. Table A.2 shows the reconstruction error $\mathcal{L}_{\text{recon}}$, the commitment loss $\mathcal{L}_{\text{commit}}$, and the perceptual loss [5], when RQ-VAE uses the partial sum $\hat{\mathbf{Z}}^{(d)}$ of up to d code embeddings for the quantized feature map of an image. All three losses, which are the reconstruction and perceptual loss of a reconstructed image, and the commitment loss $\mathcal{L}_{\text{commit}}$ of the feature map, monotonically decrease as d increases. The results imply that RQ-VAE can precisely approximate the feature map of an image, when RQ-VAE iteratively quantizes the feature map and its residuals. Figure A.14 also shows that the reconstructed images contain more fine-grained information of the original images as d increases. Thus, the experimental results validate that our RQ-VAE conducts the coarse-to-fine approximation, and RQ-Transformer can learn to generate the feature vector at the next position in a coarse-to-fine manner.

We visualize the distribution of the code usage at each depth d over the norm of code embeddings in Figure A.15. Since RQ conducts the coarse-to-fine approximation of a feature map, a smaller norm of code embeddings are used as d increases. Moreover, the overlaps between the code usage distributions show that many codes are shared in different levels of depth d. Thus, the shared codebook of RQ-VAE can maximize the utility of its codes.

C.2. The Effects of Adversarial and Perceptual Losses on Training of RQ-VAE

In Figure A.16, we visualize the reconstructed images by RQ-VAEs, which are trained without and with adversarial and perceptual losses. When the adversarial and perceptual losses are not used (the second and third columns), the reconstructed images are blurry, since the codebook is insufficient to include all information of local details in the original images. However, despite the blurriness, note that RQ-VAE with $D = 4$ (the third column) much improves the quality of reconstructed images than VQ-VAE (or RQ-VAE with $D = 1$, the second column).

Although the adversarial and perceptual losses are used to improve the quality of image reconstruction, RQ is still important to generate high-quality reconstructed images with low distortion. When the adversarial and perceptual losses are used in the training of RQ-VAEs (the fourth and fifth columns), the reconstructed images are much clear and include fine-grained details of the original images. However, the reconstructed images by VQ-VAE (or RQ-VAE with $D = 1$, the fourth column) include the unrealistic artifacts and the high distortion of the original images. Contrastively, when RQ with $D = 4$ is used to encode the information of the original images, the reconstructed images by RQ-VAE (the fifth column) are significantly realistic and do not distort the visual information in the original images.

C.3. Using D Non-Shared Codebooks of Size D/K for RQ-VAE

As mentioned in Section 3.1.2, a single codebook $\mathcal{C}$ of size K is shared for every quantization depth D instead of D non-shared codebooks of size D/K. When we replace the shared codebook of size 16,384 with four non-shared codebooks of size 4,096, rFID of RQ-VAE increases from 4.73 to 5.73, since the non-shared codebooks can approximate at most $(K/D)^D$ clusters only. In fact, a shared codebook with K=4,096 has 5.94 of rFID, which is similar to 5.73 above. Thus, the shared codebook is more effective to increase the quality of image reconstruction with limited codebook size than non-shared codebooks.

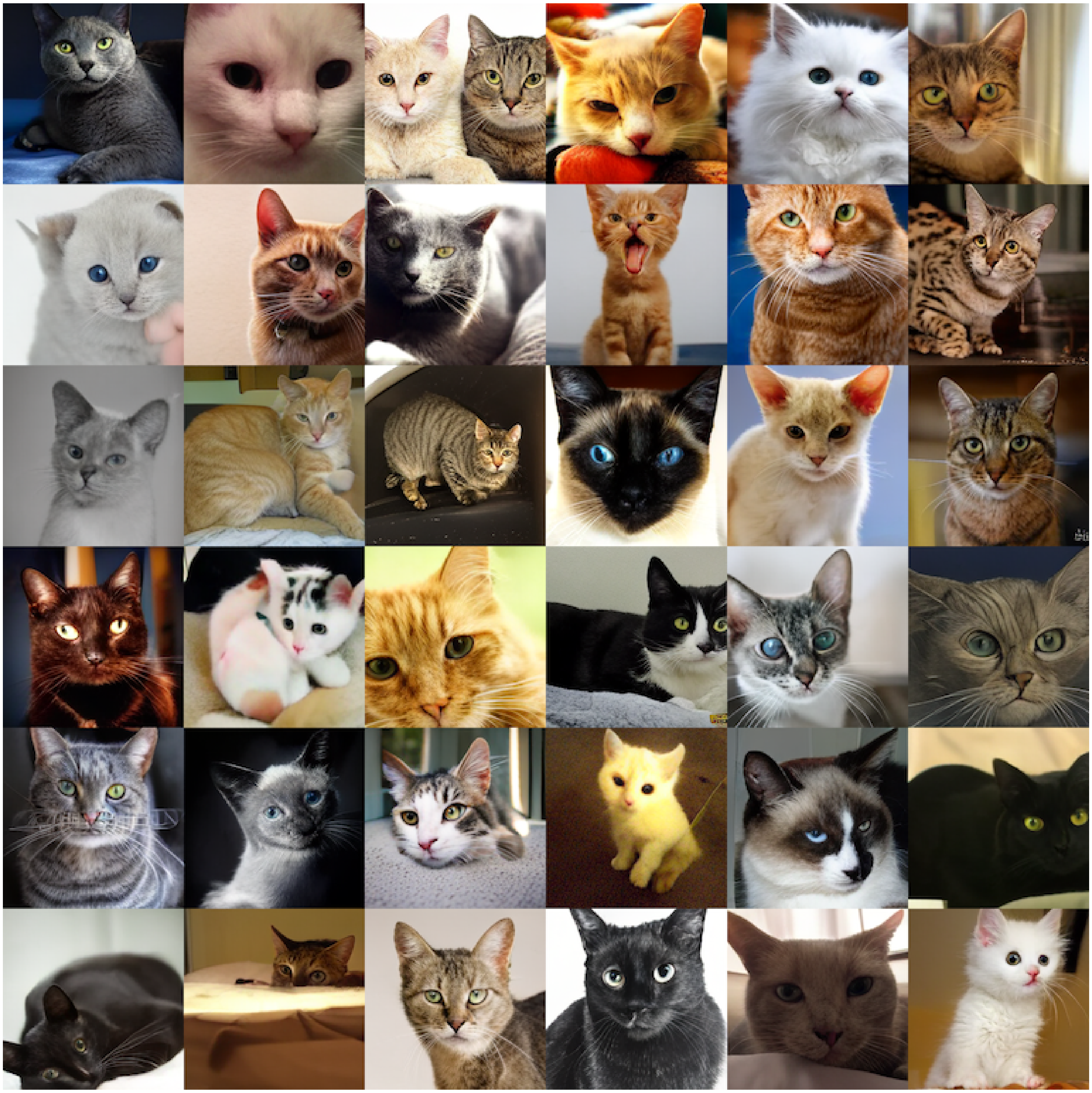

Figure A.1. Additional examples of unconditional image generation by our model trained on LSUN-cat.

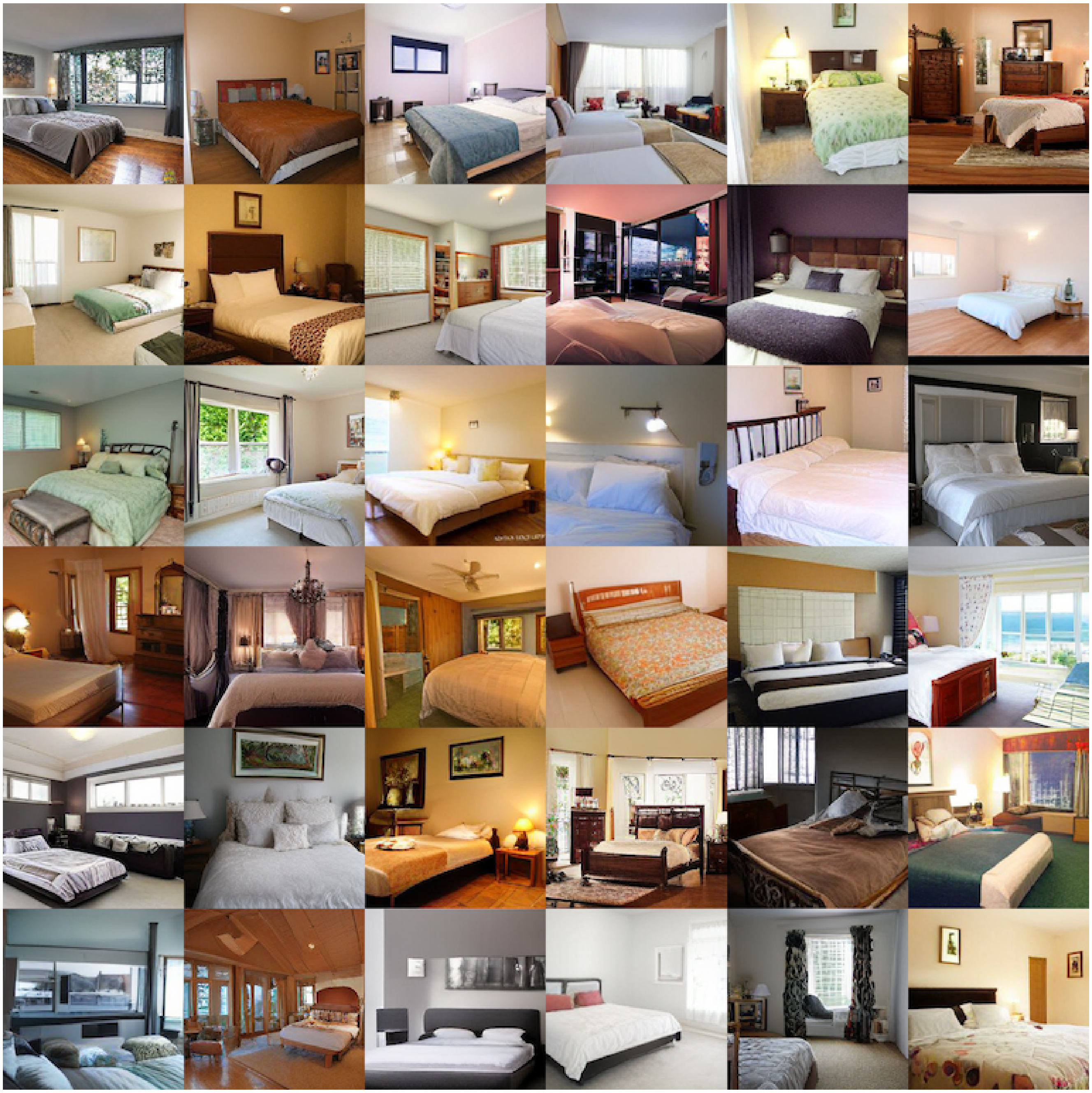

Figure A.2. Additional examples of unconditional image generation by our model trained on LSUN-bedroom.

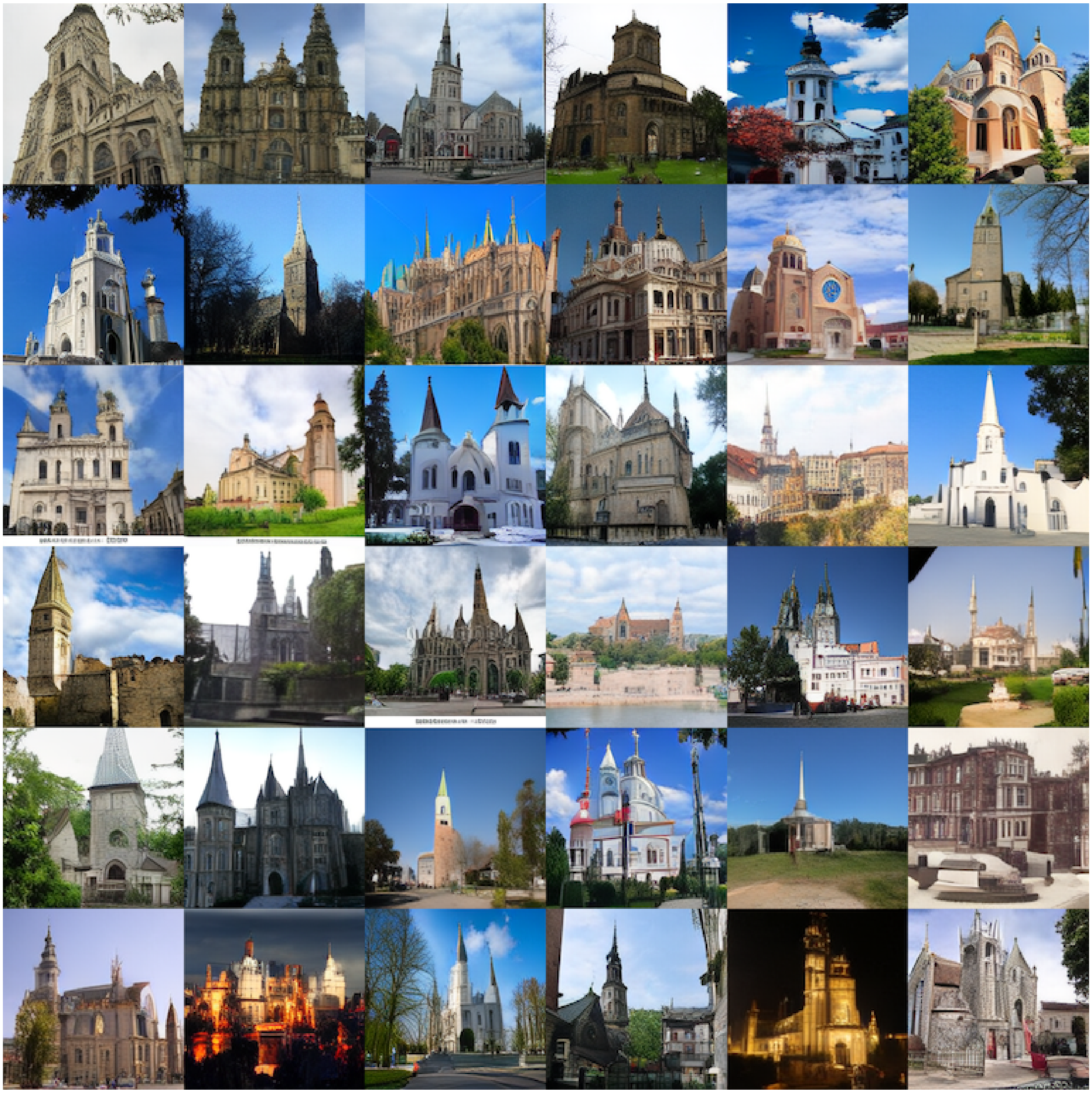

Figure A.3. Additional examples of unconditional image generation by our model trained on LSUN-church.

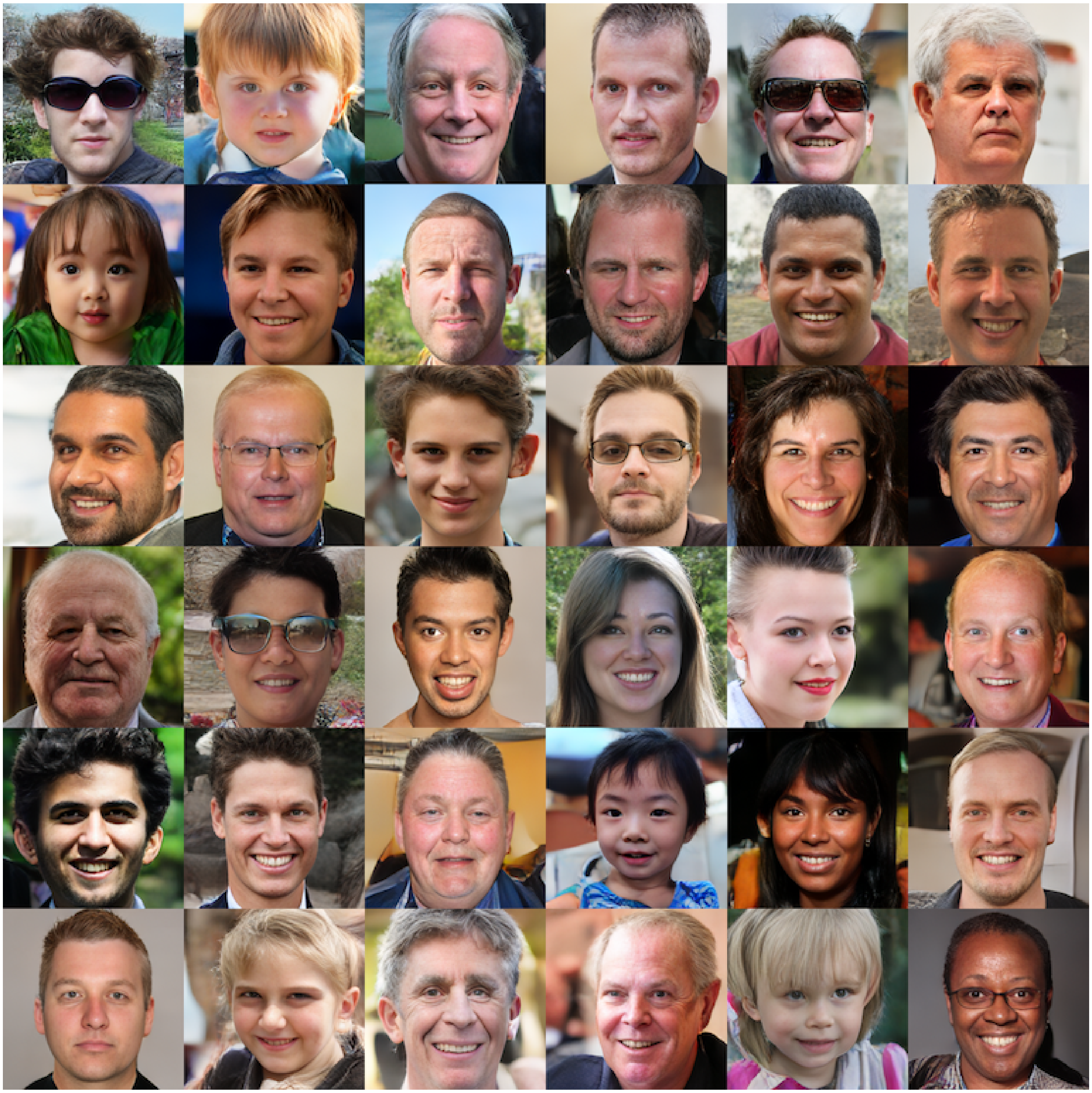

Figure A.4. Additional examples of unconditional image generation by our model trained on FFHQ.

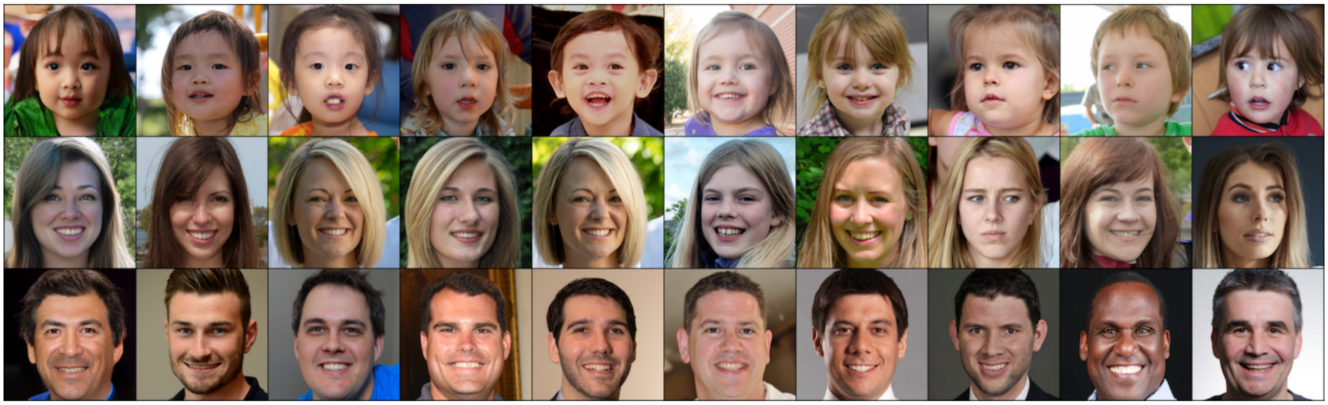

Figure A.5. Visualization of nearest neighbors in the FFHQ training samples to our generated samples. In each row, the first image is our generation. The nearest neighbors to the first image are visualized according to the similarity of VGG-16 features in descending order.

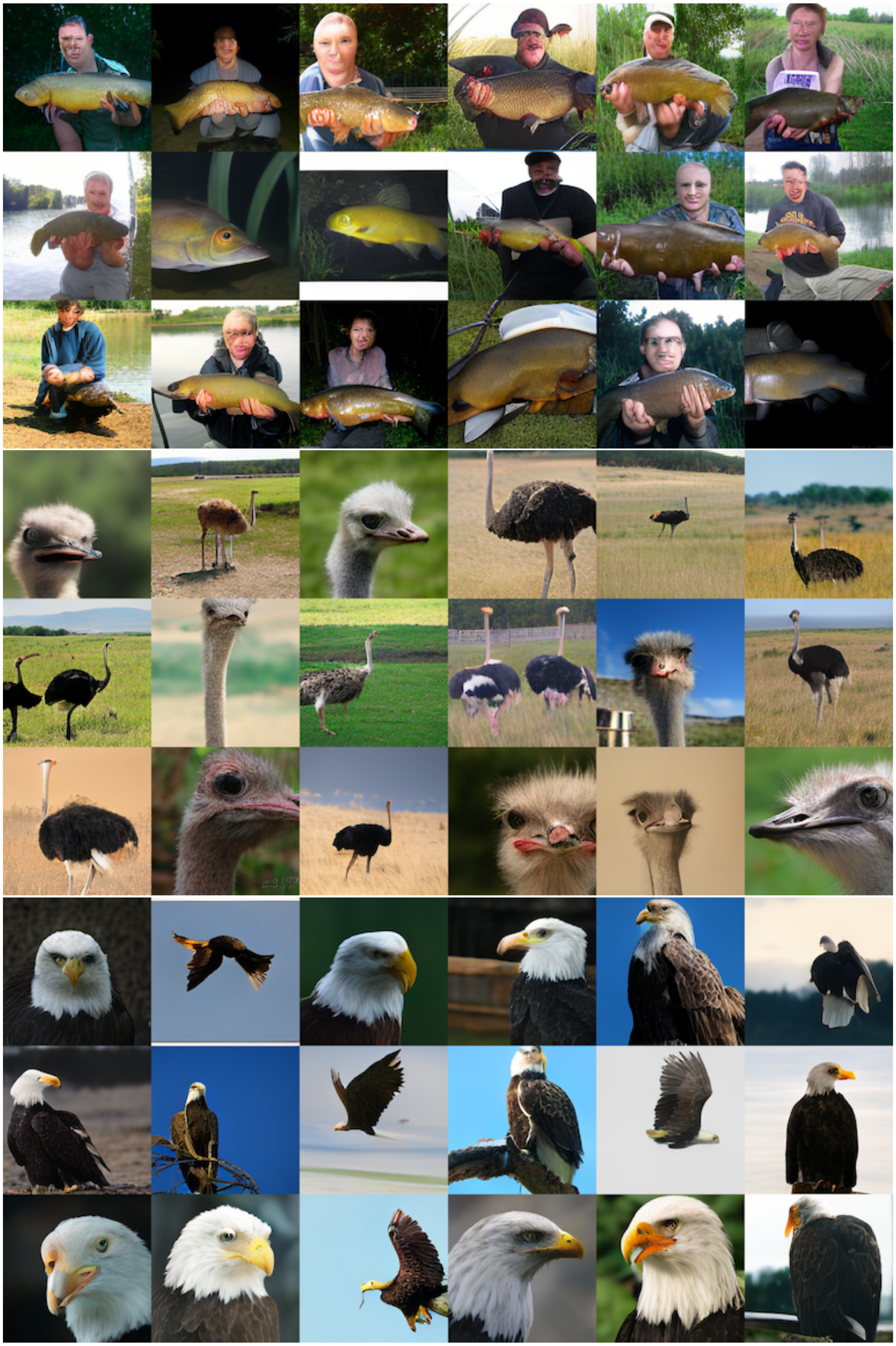

Figure A.6. Additional examples of conditional image generation by 1.4B parameters of RQ-Transformer trained on ImageNet. Top: Tench (0). Middle: Ostrich (9). Bottom: Bald eagle (22).

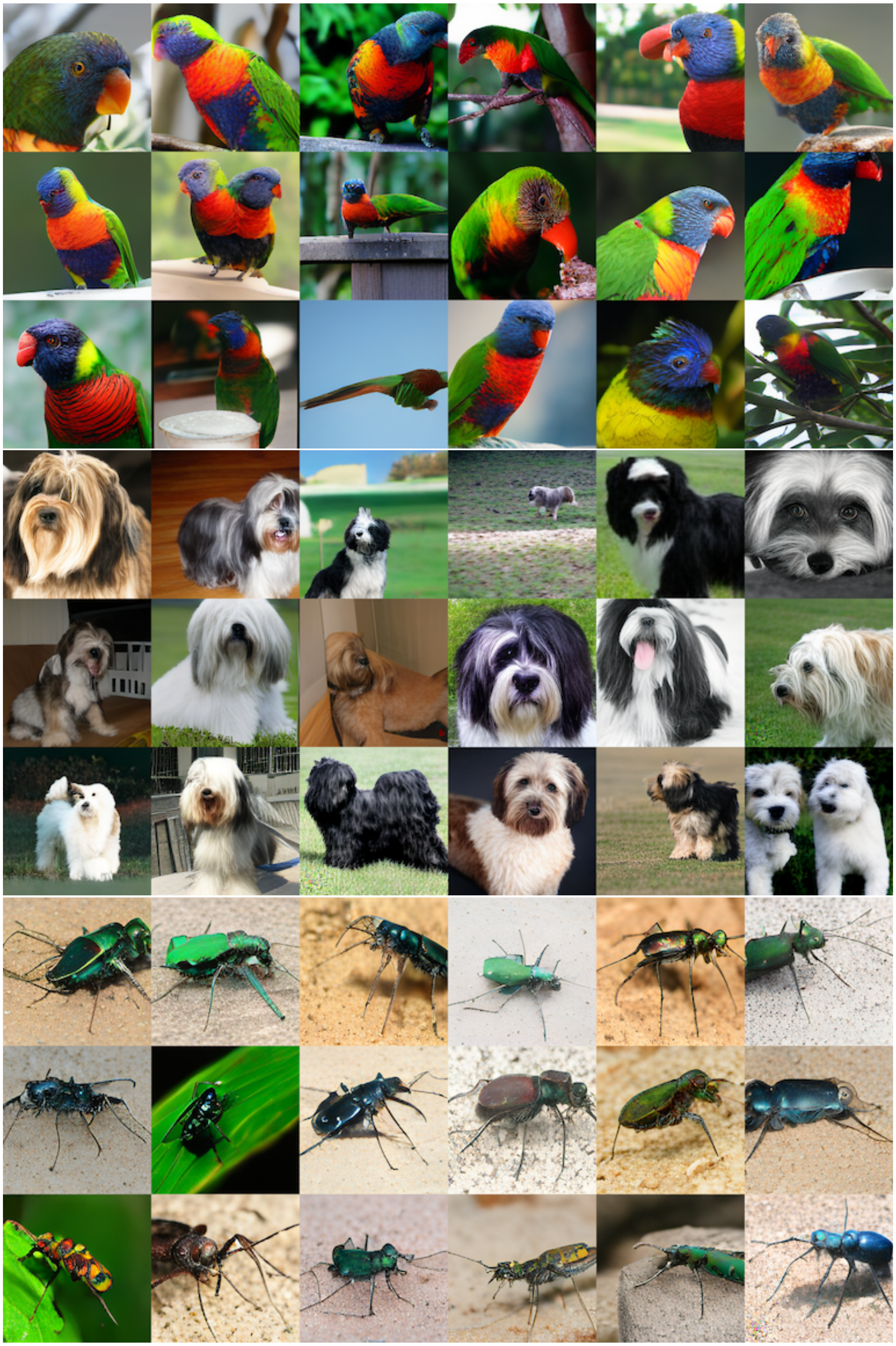

Figure A.7. Additional examples of conditional image generation by 1.4B parameters of RQ-Transformer trained on ImageNet. Top: Lorikeet (90). Middle: Tibetan terrier (200). Bottom: Tiger beetle (300).

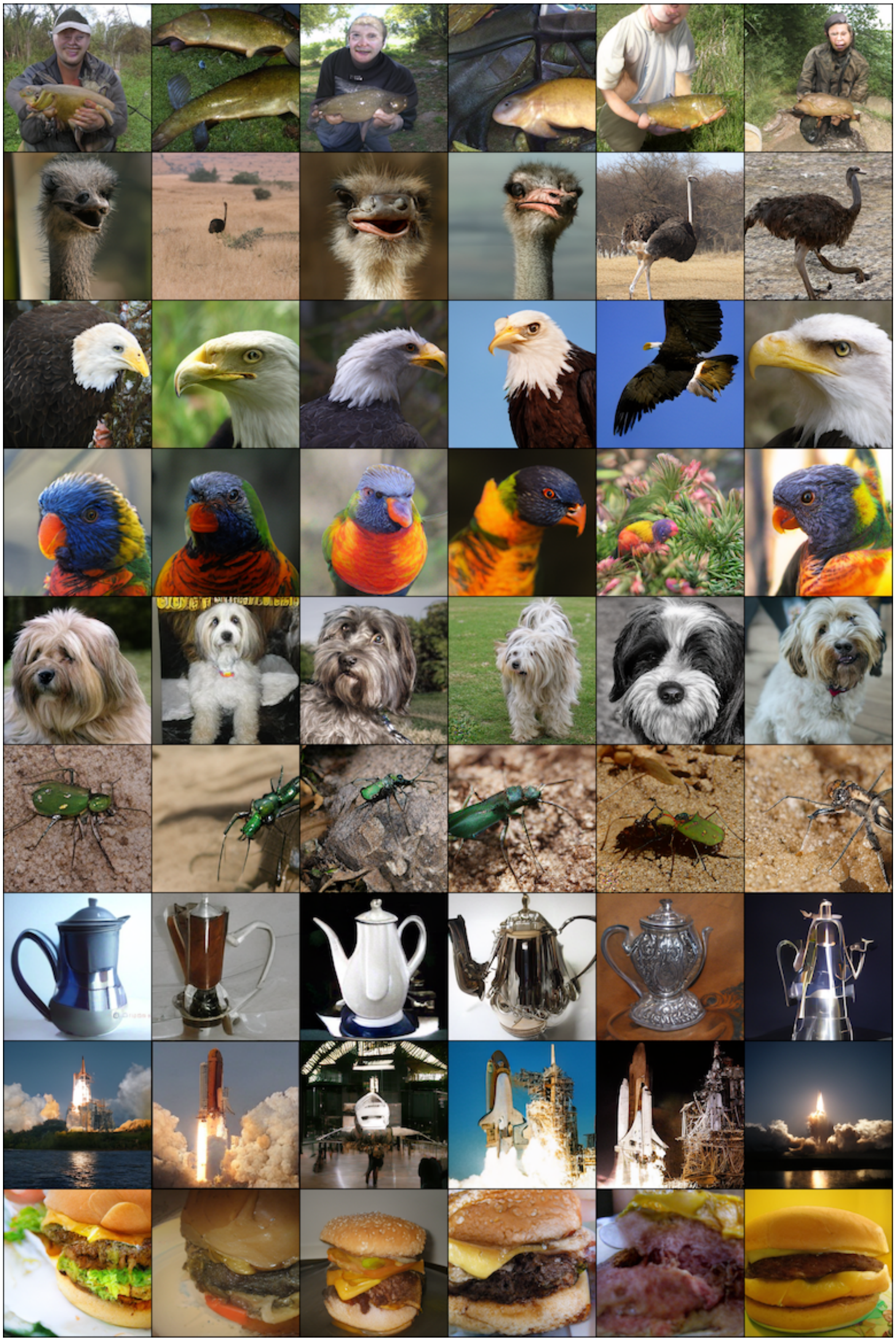

Figure A.8. Additional examples of conditional image generation by 3.8B parameters of RQ-Transformer trained on ImageNet. The classes of images in each line are tench (0), ostrich (9), bald eagle (22), lorikeet (90), tibetan terrier (200), tiger beetle (300), coffee pot (505), space shuttle (812), and cheeseburger (933), respectively.

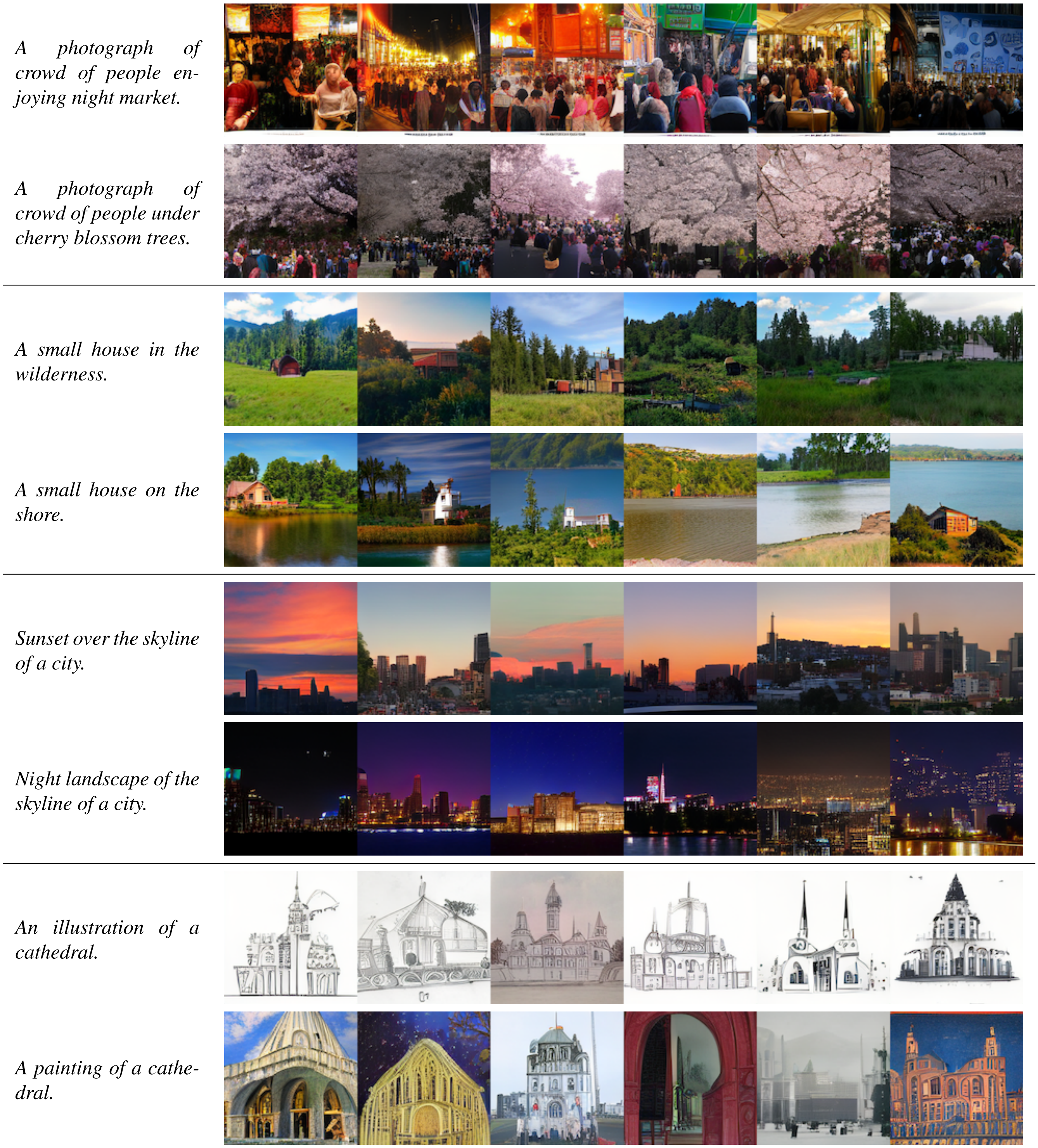
A photograph of crowd of people enjoying night market.
A photograph of crowd of people under cherry blossom trees.
A small house in the wilderness.
A small house on the shore.
Sunset over the skyline of a city.
Night landscape of the skyline of a city.
An illustration of a cathedral.
A painting of a cathedral.

Figure A.9. Additional examples of text-conditional image generation by our model trained on CC-3M. The text conditions are customized prompts, which are unseen during the training of RQ-Transformer. All images are sampled with top-k 1024 and top-p 0.9.

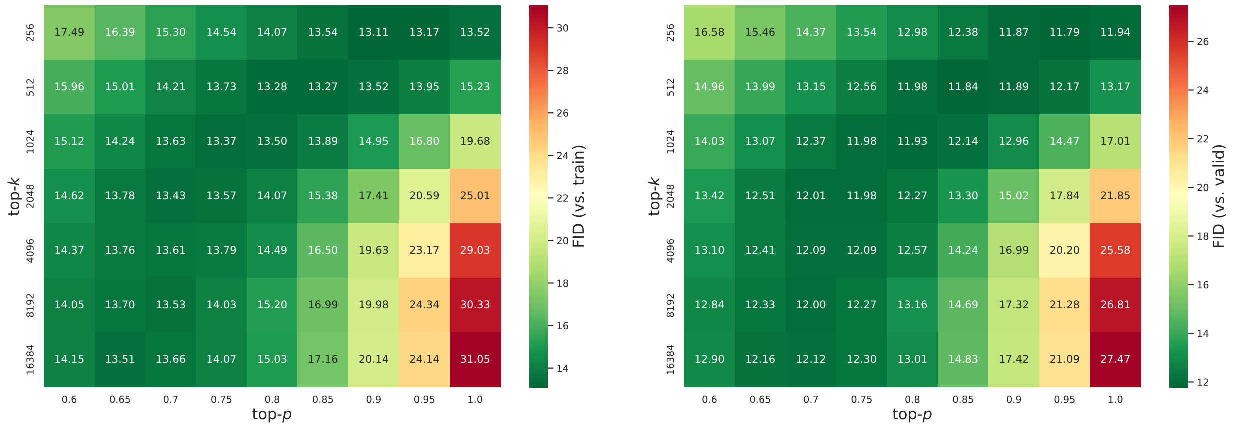

Figure A.10. FID of 50K generated samples of RQ-Transformer (821M) against the training and the validation split of ImageNet.

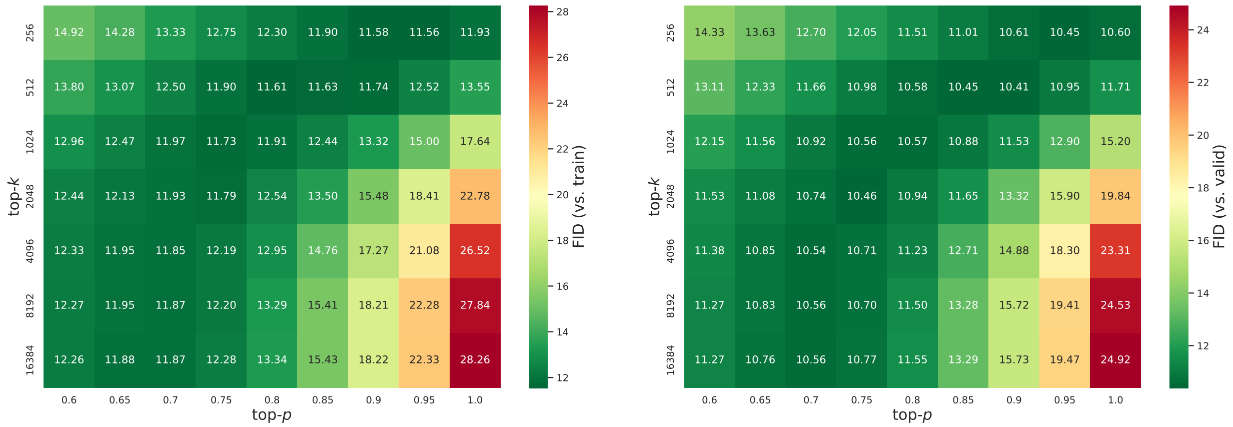

Figure A.11. FID of 50K generated samples of RQ-Transformer (1.4B) against the training and the validation split of ImageNet.

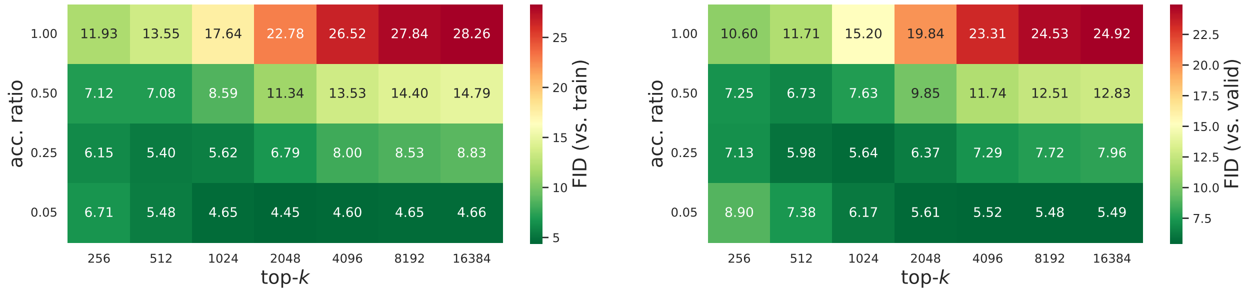

Figure A.12. FID of rejection-sampled 50K samples of RQ-Transformer (1.4B) against the training and the validation split of ImageNet.

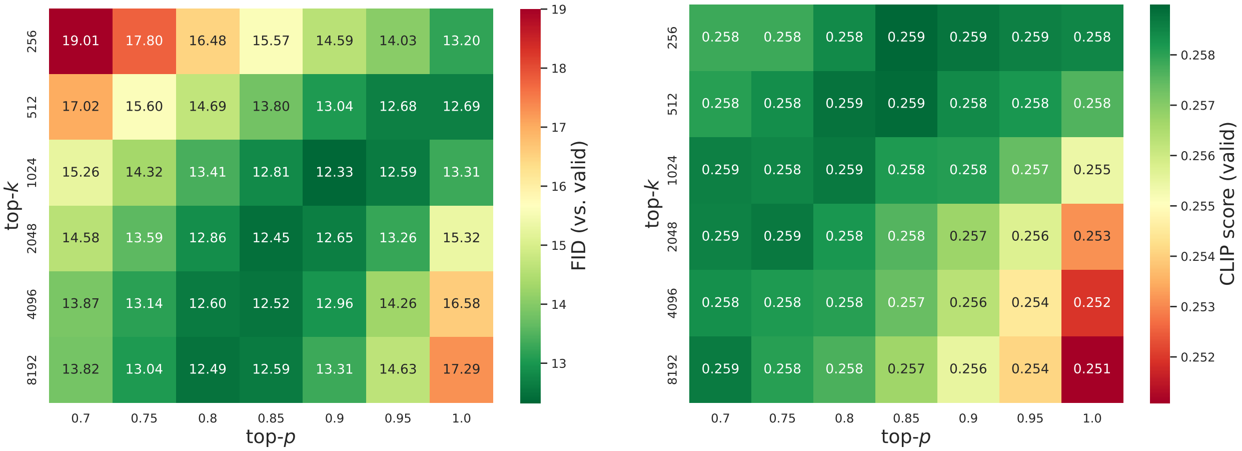

Figure A.13. FID and CLIP score of RQ-Transformer (654M) on CC-3M, evaluated against the validation set. Images are generated conditioned on each sentence in the validation set.

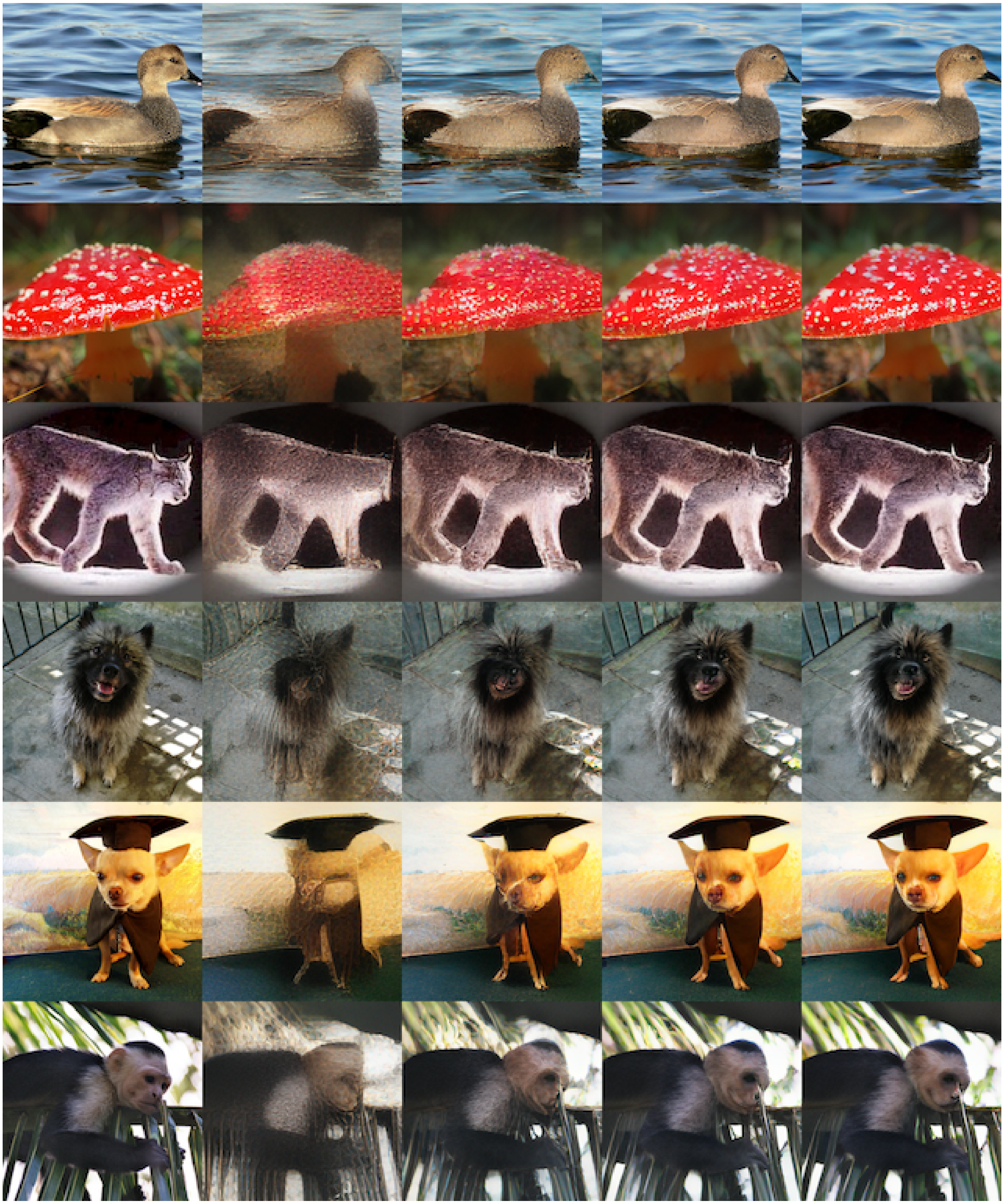

Figure A.14. Additional examples of coarse-to-fine approximation by RQ-VAE with the 8×8×4 code map. The first example in each row is the original image, and the others are constructed from $\hat{\mathbf{Z}}^{(d)}$ as d increases.

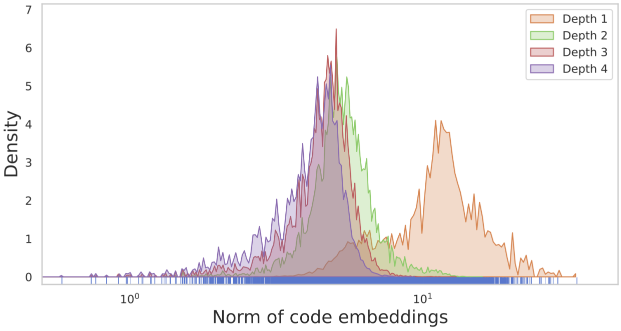

Figure A.15. The distribution of used codes at each quantization depth. The blue bar plot represents the code distribution according to the norm of embeddings. ImageNet validation data is used.

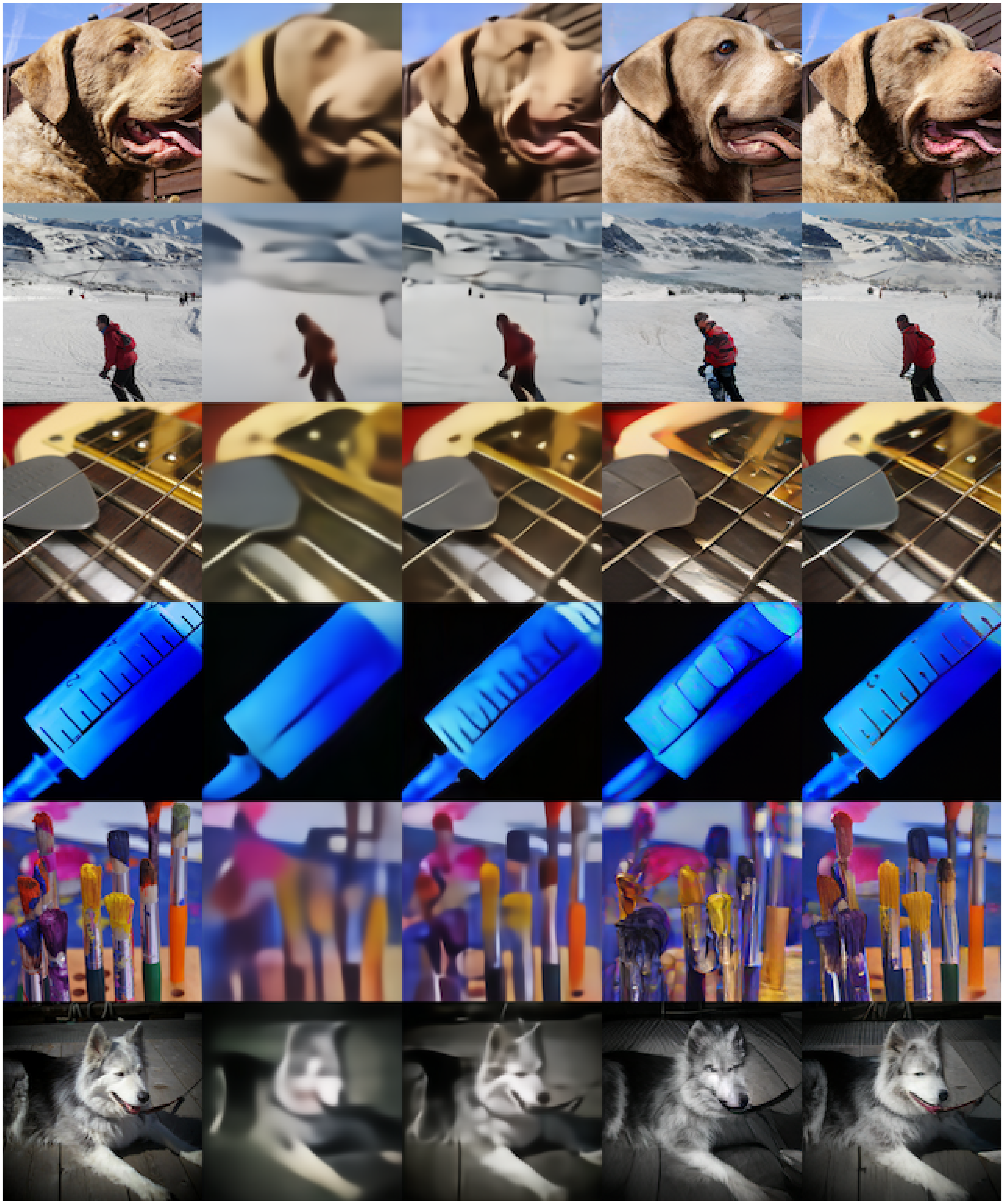

Figure A.16. Reconstruction images by RQ-VAE with and without adversarial training. The first image in each row is the original image. The second and third images are reconstructed images by RQ-VAE without adversarial training. The second image is reconstructed by RQ-VAE using 8×8×1 code map, and the third image is reconstructed by RQ-VAE using 8×8×4 code map. The fourth and fifth images are reconstructed images by RQ-VAE with adversarial training. The fourth images are reconstructed by 8×8×1 code map, and the fifth images are reconstructed by 8×8×4 code map,

References

[1] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013. 2

[2] 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 conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[3] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 1

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 1

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1

[9] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. 1

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[11] 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*, pages 5998–6008, 2017. 1

[12] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, 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. 1