A U-Net Based Discriminator for Generative Adversarial Networks Supplementary Material

Edgar Schönfeld
Bosch Center for Artificial Intelligence
edgar.schoenfeld@bosch.com

Bernt Schiele
Max Planck Institute for Informatics
schiele@mpi-inf.mpg.com

Anna Khoreva
Bosch Center for Artificial Intelligence
anna.khoreva@bosch.com

Content

This supplementary material complements the presentation of U-Net GAN in the main paper with the following:

- Additional quantitative results in Section A.
- Details on the training dynamics in Section B.
- Exemplar synthetic images on FFHQ in Section C and on COCO-Animals in Section D.
- Details on the COCO-Animals dataset in Section E.
- Network architectures and hyperparameter settings in Section F.

A. Additional Evaluations

Here we provide more detailed evaluation of the results presented in the main paper. In Table S1 we report the inception metrics for images generated on FFHQ [4], COCO-Animals [7, 5] and CelebA [8] at resolution 256×256, 128×128, and 128×128, respectively. In particular, we report the Fréchet Inception distance (FID) [2] and the Inception score (IS) [9] computed by both the PyTorch[1] and TensorFlow[2] implementations. Note that the difference between two implementations lies in using either the TensorFlow or the PyTorch in-built inception network to calculate IS and FID, resulting in slightly different scores. In all experiments, FID and IS are computed using 50k synthetic images, following [3]. By default all reported numbers correspond to the best FID achieved with 400k training iterations for FFHQ and COCO-Animals, and 800k iterations for CelebA, using the PyTorch implementation.

In the unconditional case, on FFHQ, our model achieves FID of 7.48 (8.88 in TensorFlow), which is an improvement of 4.0 (6.04 in TensorFlow) FID points over the BigGAN discriminator [1]. The same effect is observed for the conditional image generation setting on COCO-Animals. Here, our U-Net GAN achieves FID of 13.73 (13.96 in TensorFlow), improving 2.64 (2.46 in TensorFlow) points over BigGAN. To compare with other state-of-the-art models we additionally evaluate U-Net GAN on CelebA for unconditional image synthesis. Our U-Net GAN achieves 2.95 FID (in TensorFlow), outperforming COCO-GAN [6], PG-GAN [3], and the BigGAN baseline [1].

Table S2 shows that U-Net GAN does not only outperform the BigGAN baseline in terms of the best recorded FID, but also with respect to the mean, median and standard deviation computed over 5 independent runs. Note the strong drop in standard deviation from 0.24 to 0.11 on COCO-Animals and from 0.16 to 0.04 on CelebA.

[1] https://github.com/ajbrock/BigGAN-PyTorch
[2] https://github.com/bioinf-jku/TTUR

		PyTorch		TensorFlow	
Dataset	Method	FID ↓	IS ↑	FID ↓	IS ↑
FFHQ	BigGAN [1]	11.48	3.97	14.92	3.96
(256 × 256)	U-Net GAN	**7.48**	**4.46**	**8.88**	**4.50**
COCO-Animals	BigGAN [1]	16.37	11.77	16.42	11.34
(128 × 128)	U-Net GAN	**13.73**	**12.29**	**13.96**	**11.77**
	PG-GAN [3]	–	–	7.30	–
CelebA	COCO-GAN [6]	–	–	5.74	–
(128 × 128)	BigGAN [1]	3.70	3.08	4.54	3.23
	U-Net GAN	**2.03**	**3.33**	**2.95**	**3.43**

Table S1: Evaluation results on FFHQ, COCO-Animals and CelebA with PyTorch and TensorFlow FID/IS scores. The difference lies in the choice of framework in which the inception network is implemented, which is used to extract the inception metrics. See Section A for discussion.

Method	Dataset	FID			
		Best	Median	Mean	Std
BigGAN	COCO-Animals	16.37	16.55	16.62	0.24
U-Net GAN		**13.73**	**13.87**	**13.88**	**0.11**
BigGAN	FFHQ	11.48	12.42	12.35	0.67
U-Net GAN		**7.48**	**7.63**	**7.73**	**0.56**
BigGAN	CelebA	3.70	3.89	3.94	0.16
U-Net GAN		**2.03**	**2.07**	**2.08**	**0.04**

Table S2: Best, median, mean and std of FID (5 runs).

B. Characterizing the Training Dynamics

Both BigGAN and U-Net GAN experience similar stability issues, with $\sim 60\%$ of all runs being successful. For U-Net GAN, training collapse occurs generally much earlier ($\sim$ 30k iterations) than for BigGAN ($>$ 200k iterations, as also reported in [1]), allowing to discard failed runs earlier. Among successful runs for both models, we observe a lower standard deviation in the achieved FID scores, compared to the BigGAN baseline (see Table S2). Figure S1 depicts the evolution of the generator and discriminator losses (green and blue, respectively) for U-Net GAN and BigGAN over training. For U-Net GAN, the generator and discriminator losses are additionally split into the loss components of the U-Net encoder D^U_{enc} and decoder D^U_{dec}. The U-Net GAN discriminator loss decays slowly, while the BigGAN discriminator loss approaches zero rather quickly, which prevents further learning from the generator. This explains the FID gains of U-Net GAN and shows its potential to improve with longer training. The generator and discriminator loss parts from encoder (image-level) and decoder (pixel-level) show similar trends, i.e. we observe the same decay for D^U_{enc} and D^U_{dec} losses but with different scales. This is expected as D^U_{enc} can easily classify image as belonging to the real or fake class just by looking at one distinctive trait, while to achieve the same scale D^U_{dec} needs to make a uniform real or fake decision on all image pixels.

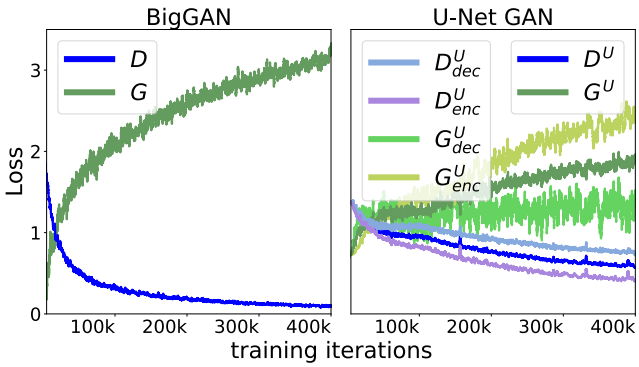

Figure S1: Comparison of the generator and discriminator loss behavior over training for U-Net GAN and BigGAN. The generator and discriminator loss of U-Net GAN is additionally split up into its encoder- and decoder components.

C. Qualitative Results on FFHQ

Here we present more qualitative results of U-Net GAN on FFHQ [4]. We use FFHQ for unconditional image synthesis and generate images with a resolution of 256×256.

Generated FFHQ samples

Figure S2 shows samples of human faces generated by U-Net GAN on FFHQ. We observe diverse images of high quality, maintaining local and global realism.

Per-pixel U-Net discriminator feedback

In Figure S3 we visualize synthetic images and their corresponding per-pixel feedback of the U-Net discriminator. Note that the U-Net discriminator provides a very detailed and spatially coherent response, which enables the generator to further improve the image quality.

Interpolations in the latent space

Figure S4 displays human faces generated by U-Net GAN through linear interpolation in the latent space between two synthetic samples. We observe that the interpolations are semantically smooth between faces, e.g. an open mouth gradually becomes a closed mouth, hair progressively grows or gets shorter in length, beards or glasses smoothly fade or appear, and hair color changes seamlessly.

Comparison between BigGAN and U-Net GAN

In Figure S5 we present a qualitative comparison of uncurated images generated with the unconditional BigGAN model [1] and our U-Net GAN. Note that the images generated by U-Net GAN exhibit finer details and maintain better local realism.

CutMix images and U-Net discriminator predictions

In Figure S6 we show more examples of the CutMix images and the corresponding U-Net based discriminator D^U predictions. Note that in many cases, the decoder output for fake image patches is darker than for real image ones. However, the predicted intensity for an identical local patch can change for different mixing scenarios. This indicates that the U-Net discriminator takes contextual information into account for local decisions.

Figure S2: Images generated by U-Net GAN trained on FFHQ with resolution 256×256.

Figure S3: Samples generated by U-Net GAN and the corresponding real-fake predictions of the U-Net decoder. Brighter colors correspond to the discriminator confidence of pixel being real (and darker of being fake).

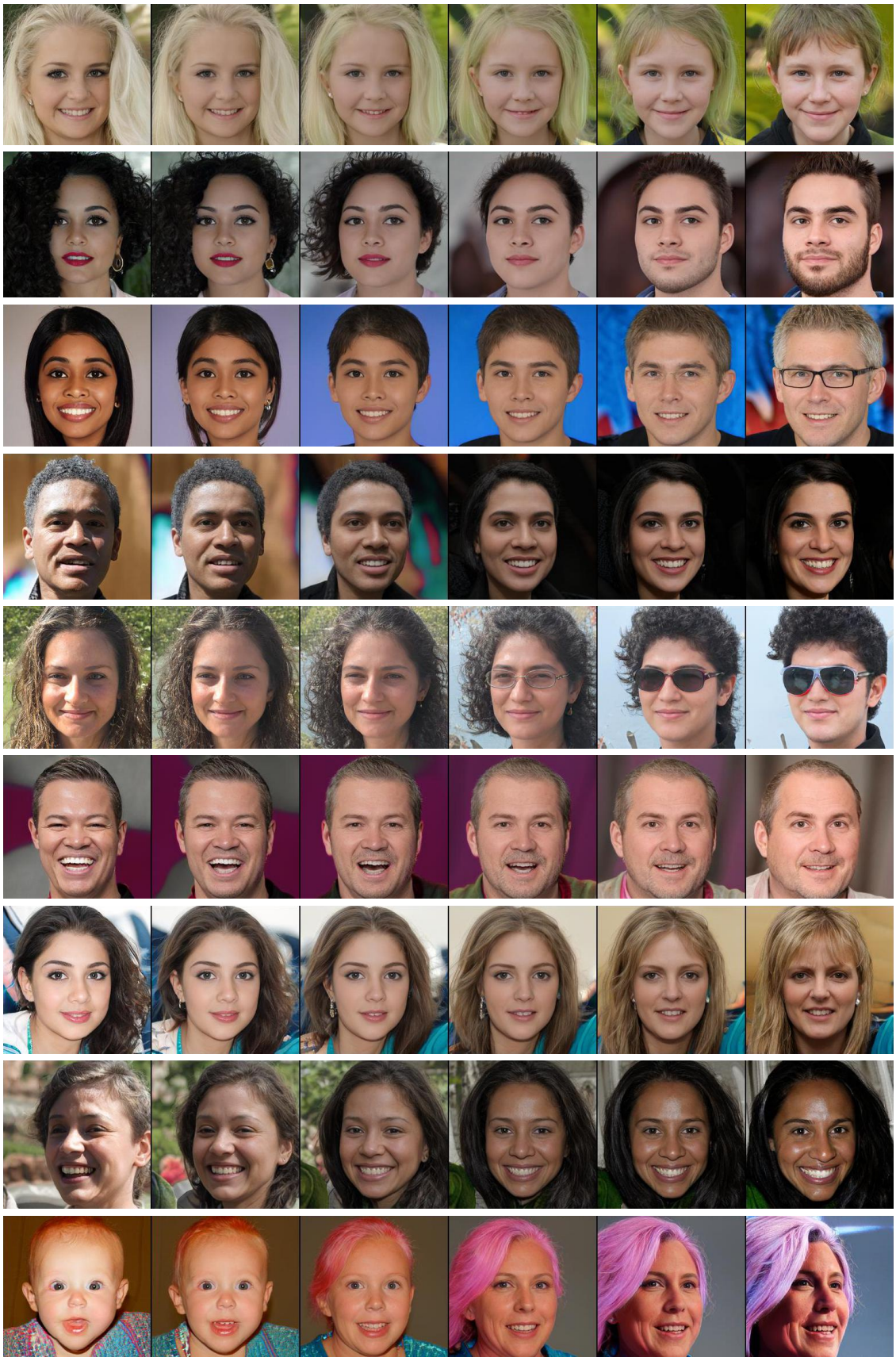

Figure S4: Images generated with U-Net GAN on FFHQ with resolution 256×256 when interpolating in the latent space.

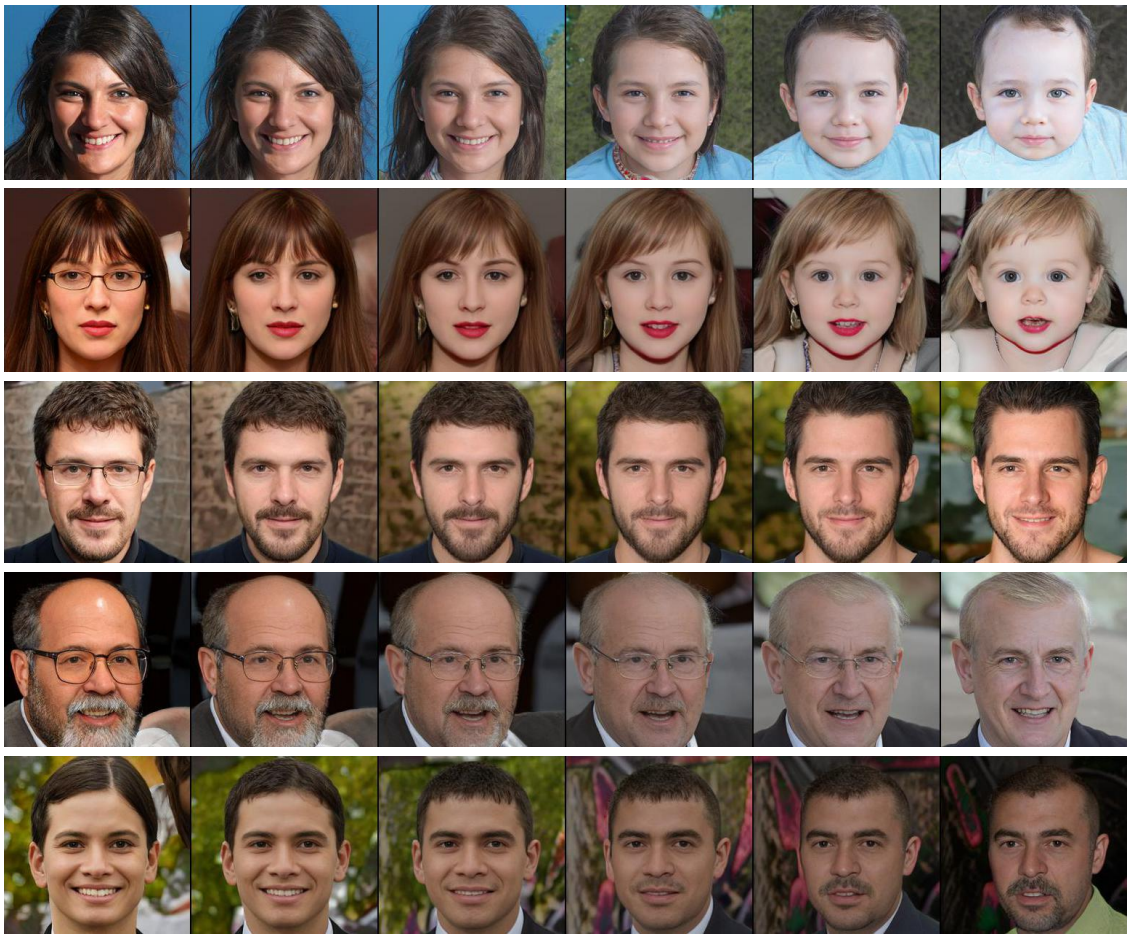

Figure S4: More images generated with U-Net GAN on FFHQ with resolution 256×256 when interpolating in the latent space.

BigGAN

U-Net GAN

Figure S5: Qualitative comparison of uncurated images generated with the unconditional BigGAN model (top) and our U-Net GAN (bottom) on FFHQ with resolution 256×256. Note that the images generated by U-Net GAN exhibit finer details and maintain better local realism.

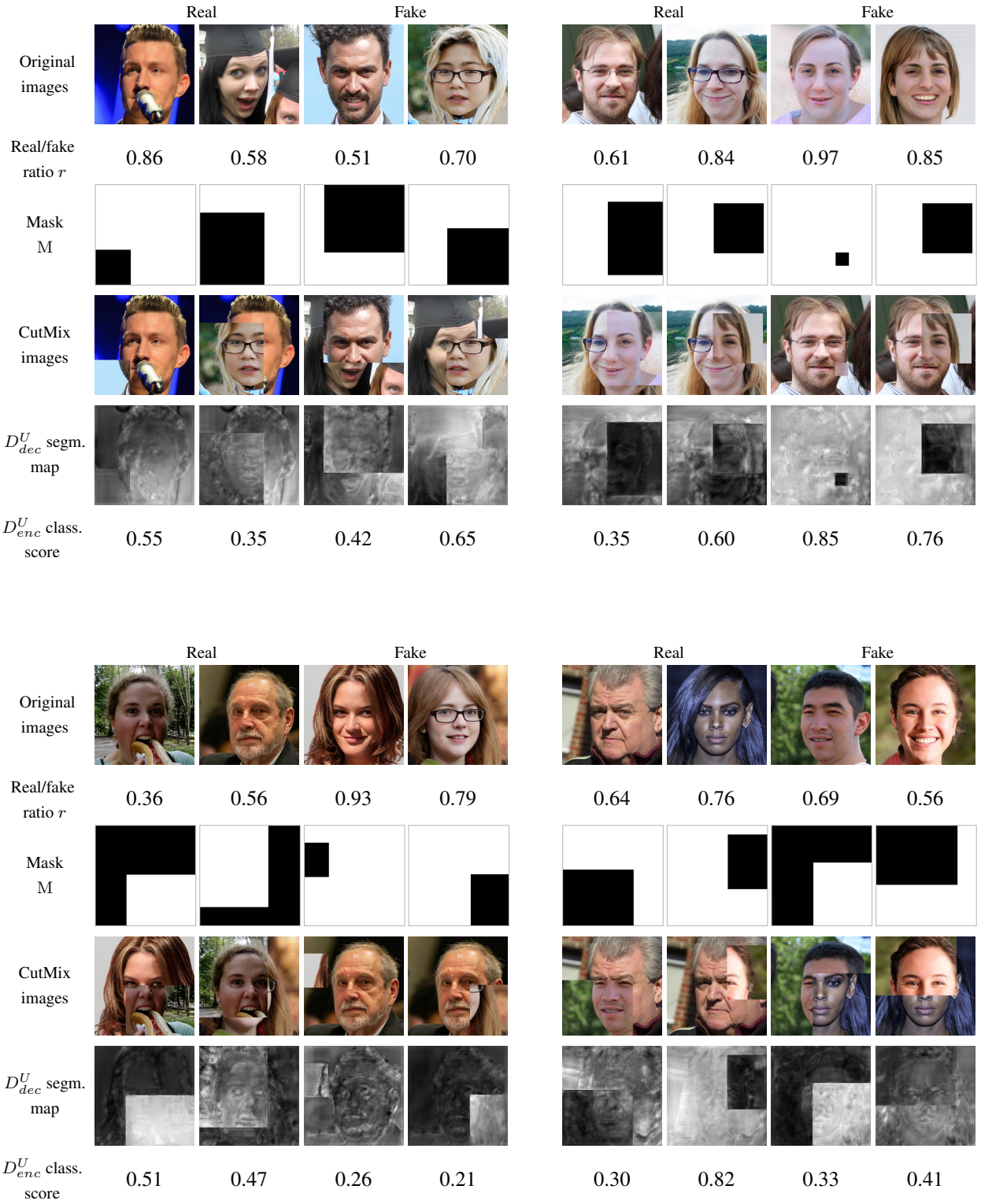

Figure S6: Visualization of the CutMix augmentation and the predictions of the U-Net discriminator on CutMix images. 1st row: real and fake samples. 2nd&3rd rows: sampled real/fake CutMix ratio r and corresponding binary masks M (color code: white for real, black for fake). 4th row: generated CutMix images from real and fake samples. 5th&6th row: the corresponding real/fake segmentation maps of the U-Net GAN decoder D^U_{dec} with the corresponding predicted classification scores by the encoder D^U_{enc} below.

D. Qualitative Results on COCO-Animals

Here we present more qualitative results of U-Net GAN on COCO-Animals [7, 5]. We use COCO-Animals for class conditional image synthesis and generate images with the resolution of 128×128.

Generated COCO-Animals samples

Figure S7 shows generated samples of different classes on COCO-Animals. We observe images of good quality and high intra-class variation. We further notice that employing the class-conditional projection (as used in BigGAN) in the pixel output space of the decoder does not introduce class leakage or influence the class separation in any other way. These observations further confirm that our U-Net GAN is effective in class-conditional image generation as well.

Per-pixel U-Net discriminator feedback

Figure S8 shows generated examples and the corresponding per-pixel predictions of the U-Net discriminator. We observe that the resulting maps often tend to exhibit a bias towards objects.

Interpolations in the latent space

Figure S9 displays images generated on COCO-Animals by U-Net GAN through linear interpolation in the latent space between two synthetic samples. We observe that the interpolations are semantically smooth between different classes of animals, e.g. background seamlessly changes between two scenes, number of instances gradually increases or decreases, shape and color of objects smoothly changes from left to right.

E. Details on the COCO-Animals Dataset

COCO-Animals is a medium-sized ($\sim$ $38k$) dataset composed of 10 animal classes, and is intended for experiments that demand a high-resolution equivalent for CIFAR10. The categories are *bird, cat, dog, horse, cow, sheep, giraffe, zebra, elephant, and monkey*. The images are taken from COCO [7] and the OpenImages [5] subset that provides semantic label maps and binary mask and is also human-verified. The two datasets have a great overlap in animal classes. We take *all* images from COCO and the aforementioned OpenImages split in the categories *horse, cow, sheep, giraffe, zebra* and *elephant*. The *monkey* images are taken over directly from OpenImages, since this category contained more training samples than the next biggest COCO animal class *bear*. The class *bear* and *monkey* are not shared between COCO and OpenImages. Lastly, the categories *bird, cat* and *dog* contained vastly more samples than all other categories. For this reason, we took over only a subset of the total of all images in these categories. These samples were picked from OpenImages only, for their better visual quality. To ensure good quality of the picked examples, we used the provided bounding boxes to filter out images in which the animal of interest is either too small or too big ($> 80\%, < 30\%$ of the image area for cats, $> 70\%, < 50\%$ for birds and dogs). The thresholds were chose such that the number of appropriate images is approximately equal.

F. Architectures and Training Details

Architecture details of the BigGAN model [1] and our U-Net discriminator are summarized in Table S3 and Table S4. From these tables it is easy to see that the encoder and decoder of the U-Net discriminator follow the original BigGAN discriminator and generator setups, respectively. One difference is that the number of input channels in the U-Net decoder is doubled, since encoder features are concatenated to the input features.

Table S4 presents two U-Net discriminator networks: a class-conditional discriminator for image resolution 128×128, and an unconditional discriminator for resolution 256×256. The decoder does not have 3 output channels (like the BigGAN generator that it is copied from), but $ch = 64$ channels, resulting in a feature map h of size $64 \times 128 \times 128$, to which a 1×1 convolution is applied to reduce the number of channels to 1. In the class-conditional architecture, a learned class-embedding is multiplied with the aforementioned 64-dimensional output h at every spatial position, and summed along the channel dimension (corresponding to the inner product). The resulting map of size $1 \times 128 \times 128$ is added to the output, leaving us with 128×128 logits.

We follow [1] for setting up the hyperparameters for training U-Net GAN, which are summarized in Table S5.

Hyperparameter	Value
Optimizer	Adam ($\beta_1 = 0, \beta_2 = 0.999$)
G's learning rate	1e-4 (256), 5e-5 (128)
D's learning rate	5e-4 (256), 2e-4 (128)
Batch size	20 (256), 80 (128)
Weight Initialization	Orthogonal

Table S5: Hyperparameters of U-Net GAN

Regarding the difference between class-conditional and unconditional image generation, it is worth noting that the CutMix regularization is applied only to samples within the same class. In other words, real and generated samples are mixed only within the class (e.g. real and fake zebras, but not real zebras with fake elephants).

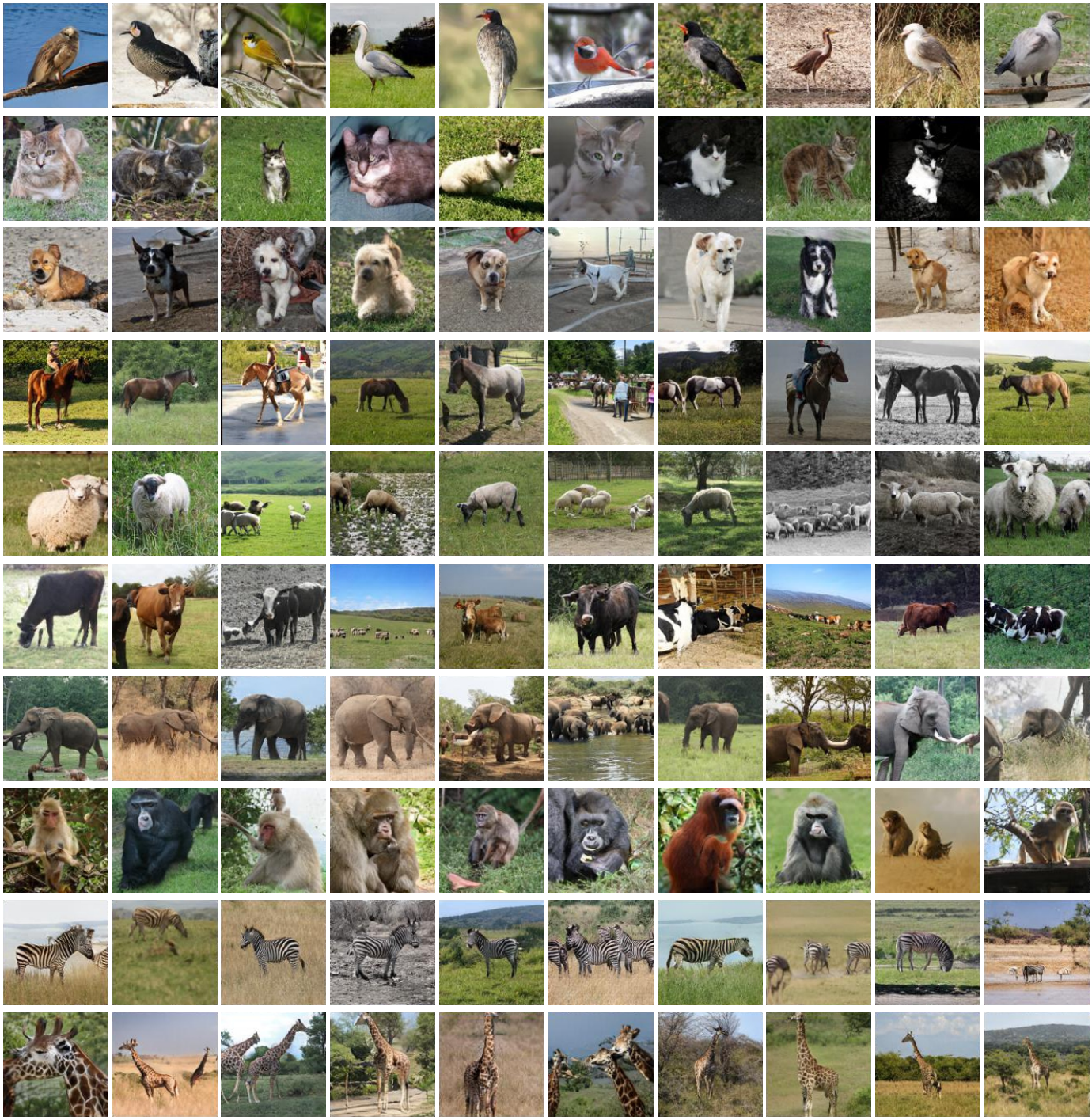

Figure S7: Images generated with U-Net GAN trained on COCO-Animals with resolution 128×128.

Figure S8: Generated samples on COCO-Animals and the corresponding U-Net decoder predictions. Brighter colors correspond to the discriminator confidence of pixel being real (and darker of being fake).

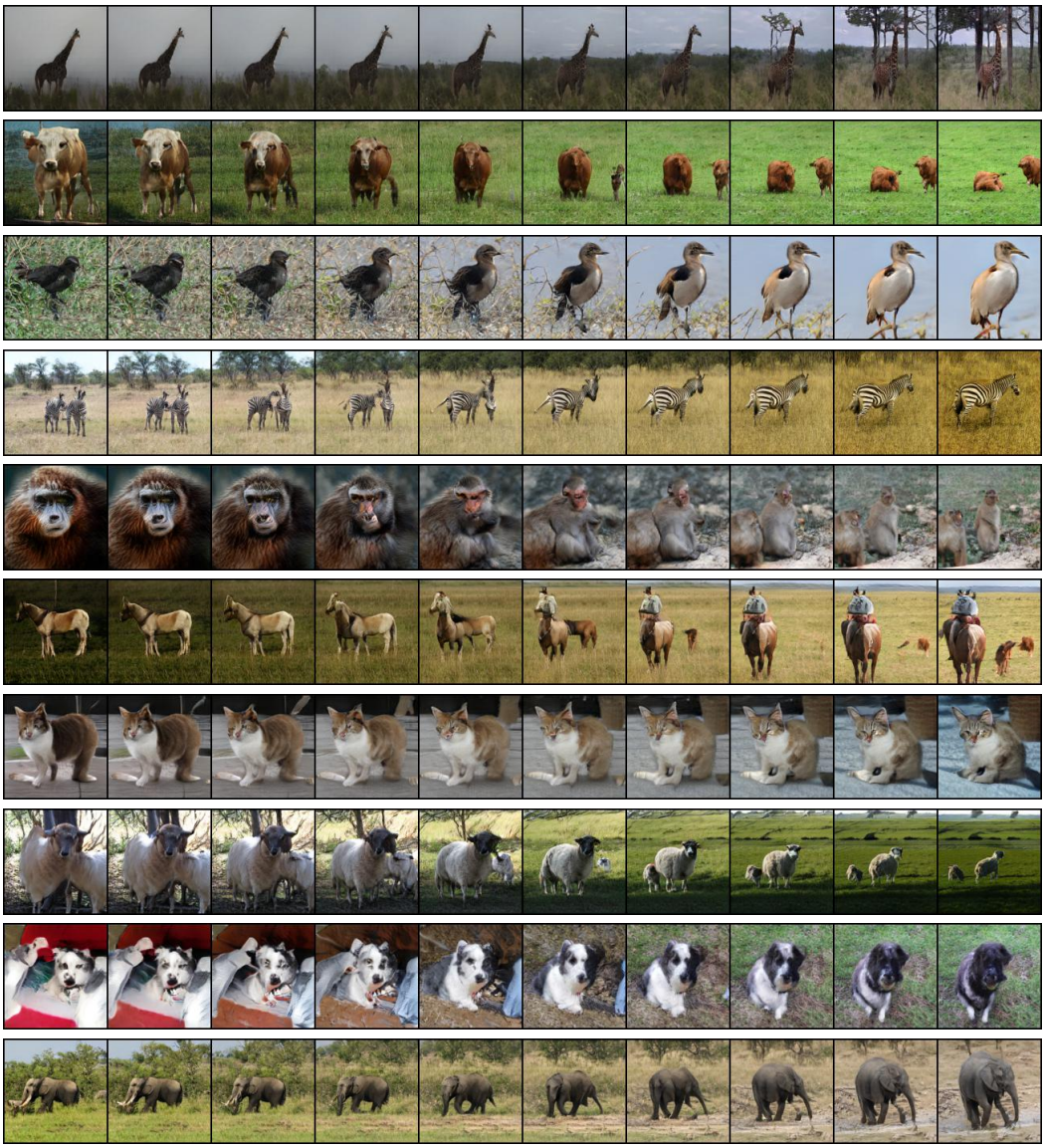

Figure S9: Images generated with U-Net GAN on COCO-Animals with resolution 128×128 when interpolating in the latent space between two synthetic samples (left to right).

(a) BigGAN Generator (128×128, class-conditional)

$z \in \mathbb{R}^{120} \sim \mathcal{N}(0, I)$
Embed(y) $\in \mathbb{R}^{128}$
Linear (20 + 128) $\rightarrow 4 \times 4 \times 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$
ResBlock up $4ch \rightarrow 2ch$
Non-Local Block (64×64)
ResBlock up $2ch \rightarrow ch$
BN, ReLU, 3×3 Conv $ch \rightarrow 3$
Tanh

(b) BigGAN Discriminator (128×128, class-conditional)

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down $ch \rightarrow 2ch$
Non-Local Block (64×64)
ResBlock down $2ch \rightarrow 4ch$
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$
ResBlock down $16ch \rightarrow 16ch$
ReLU, Global sum pooling
Embed(y)$\cdot h$ + (linear$\rightarrow 1$)

(c) BigGAN Generator (256×256, unconditional)

$z \in \mathbb{R}^{140} \sim \mathcal{N}(0, I)$
Linear (20 + 128) $\rightarrow 4 \times 4 \times 16ch$
ResBlock up $16ch \rightarrow 16ch$
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$
ResBlock up $4ch \rightarrow 2ch$
Non-Local Block (128×128)
ResBlock up $2ch \rightarrow ch$
BN, ReLU, 3×3 Conv $ch \rightarrow 3$
Tanh

(d) BigGAN Discriminator (256×256, unconditional)

RGB image $x \in \mathbb{R}^{256 \times 256 \times 3}$
ResBlock down $ch \rightarrow 2ch$
ResBlock down $2ch \rightarrow 4ch$
Non-Local Block (64×64)
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$
ResBlock down $16ch \rightarrow 16ch$
ReLU, Global sum pooling
linear$\rightarrow 1$

Table S3: The BigGAN [1] generator and discriminator architectures for class-conditional and unconditional tasks of generating images at different resolutions. Top (a and b): The class-conditional BigGAN model for resolution 128×128. Bottom (c and d): The BigGAN model for resolution 256×256, modified to be *un*conditional.

(a) U-Net GAN Discriminator (256×256, unconditional)

RGB image $x \in \mathbb{R}^{256\times256\times3}$
ResBlock down $ch \rightarrow 2ch$
ResBlock down $2ch \rightarrow 4ch$
Optional Non-Local Block (64×64)
ResBlock down $4ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$ *(see below)
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $(8+8)ch \rightarrow 8ch$
ResBlock up $(8+8)ch \rightarrow 4ch$
ResBlock up $(4+4)ch \rightarrow 2ch$
ResBlock up $(2+2)ch \rightarrow ch$
ResBlock up $(ch+ch) \rightarrow ch$
ResBlock $ch \rightarrow 1$
Sigmoid
* ReLU, Global sum pooling, linear$\rightarrow 1$

(b) U-Net GAN Discriminator(128×128, class-conditional)

RGB image $x \in \mathbb{R}^{128\times128\times3}$
ResBlock down $ch \rightarrow 2ch$
Optional Non-Local Block (64×64)
ResBlock down $2ch \rightarrow 4ch$
ResBlock down $8ch \rightarrow 8ch$
ResBlock down $8ch \rightarrow 16ch$ *(see below)
ResBlock up $16ch \rightarrow 8ch$
ResBlock up $(8+8)ch \rightarrow 4ch$
ResBlock up $(4+4)ch \rightarrow 2ch$
ResBlock up $(2+2)ch \rightarrow ch$
ResBlock up $(ch+ch) \rightarrow ch$
Embed(y)$\cdot h$ + (Conv $ch \rightarrow 1$)
Sigmoid
* ReLU, Global sum pooling
Embed(y)$\cdot h$ + (linear$\rightarrow 1$)

Table S4: The U-Net GAN discriminator architectures for class-conditional (a) and unconditional (b) tasks of generating images at resolution 128×128 and 256×256, respectively.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019. 1, 2, 9, 12

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[5] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 1, 9

[6] 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 *International Conference on Computer Vision (ICCV)*, 2019. 1

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 9

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015. 1

[9] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1