HoloGAN: Unsupervised Learning of 3D Representations From Natural Images — Supplemental Document —

Thu Nguyen-Phuoc [1] Chuan Li [2] Lucas Theis [3] Christian Richardt [1] Yong-Liang Yang [1]

[1] University of Bath [2] Lambda Labs [3] Twitter

1. Additional qualitative results

We show additional qualitative results for the Basel Face dataset in Figure 1. Please see our supplemental video for further results.

Figure 1. HoloGAN results when trained on the Basel Face dataset. **Top:** Five sampled faces (one per row) for an azimuth range of 100°. **Bottom:** Five sampled faces for an elevation range of 40°.

Figure 2. Ablation study showing images with changing azimuths (from left to right). The identity regulariser helps learning the full range of poses (a), preventing the model from ignoring views such as the rear view (b).

2. Additional ablation study

Here we show the effectiveness of the identity regulariser. We find that this regulariser encourages HoloGAN to only use $\mathbf{z}$ for the identity, and use pose θ more effectively to capture the variety of poses in the dataset. As shown in Figure 2, HoloGAN trained with the identity regulariser successfully learns the full variation of poses.

3. Linear interpolation

Figure 3 shows the results of interpolating the latent vector $\mathbf{z}$, while keeping the pose fixed.

4. Additional style mixing results

Here we use different latent codes at different resolutions of the 3D features ($\mathbf{z}_1$ for tensor $8\times8\times8$, $\mathbf{z}_2$ for tensor $16\times16\times16$), and the same $\mathbf{z}_3$ for all 2D features. Figure 4 shows that the deeper features ($8\times8\times8$, controlled by $\mathbf{z}_1$) control more global features such the overall shape, while shallower features ($16\times16\times16$, controlled by $\mathbf{z}_2$) influence more fine-grained details such as gender or makeup (for CelebA), wind shield or doors (for Cars), pillows or windows (for LSUN bedroom). This observation agrees with Karras et al. [3]; however, our approach additionally factors out pose from identity, and therefore the global features no longer control poses. Note that the overall colour and lighting are the same in Figure 4, because we use the same $\mathbf{z}_3$ for the 2D features, which control appearance.

5. KID comparisons

In Figure 5, we show generated samples by DCGAN [8], LSGAN [6], WGAN-GP [2] and HoloGAN for three datasets: CelebA, Chairs and Cars. Compared to other models, HoloGAN produces samples with higher visual fidelity, and offers explicit control over poses of objects in the generated images.

6. Datasets

Here we provide more information on our training datasets. We list them in increasing order of complexity. All images are centre-cropped, apart from the Cars dataset, where we make use of the bounding box provided by the dataset, as well as adding extra 500 images crawled from Google image search. During training, we randomly flip the training images to augment the training data.

Name	Type	Diff. identities	# Images	Resolution	Azimuth	Elevation	Scaling
Basel Face [7]	Synthetic	Yes	80,000	128×128	220°–320°	70°–110°	None
Cats [11]	Real	Yes	9,033	64×64	220°–320°	60°–95°	None
CelebA [5]	Real	Yes	202,599	128×128	220°–320°	60°–95°	None
Chairs [1]	Synthetic	No	406,680	64×64	0°–359°	10°–170°	None
Cars [9]	Real	Yes	139,714	128×128	0°–359°	60°–95°	0.8–1.5
LSUN bedroom [10]	Real	Yes	3,033,042	128×128	220°–320°	60°–95°	0.8–1.5

7. Network architecture

Below are the description of our generator and discriminator network. For the 128×128 generator:

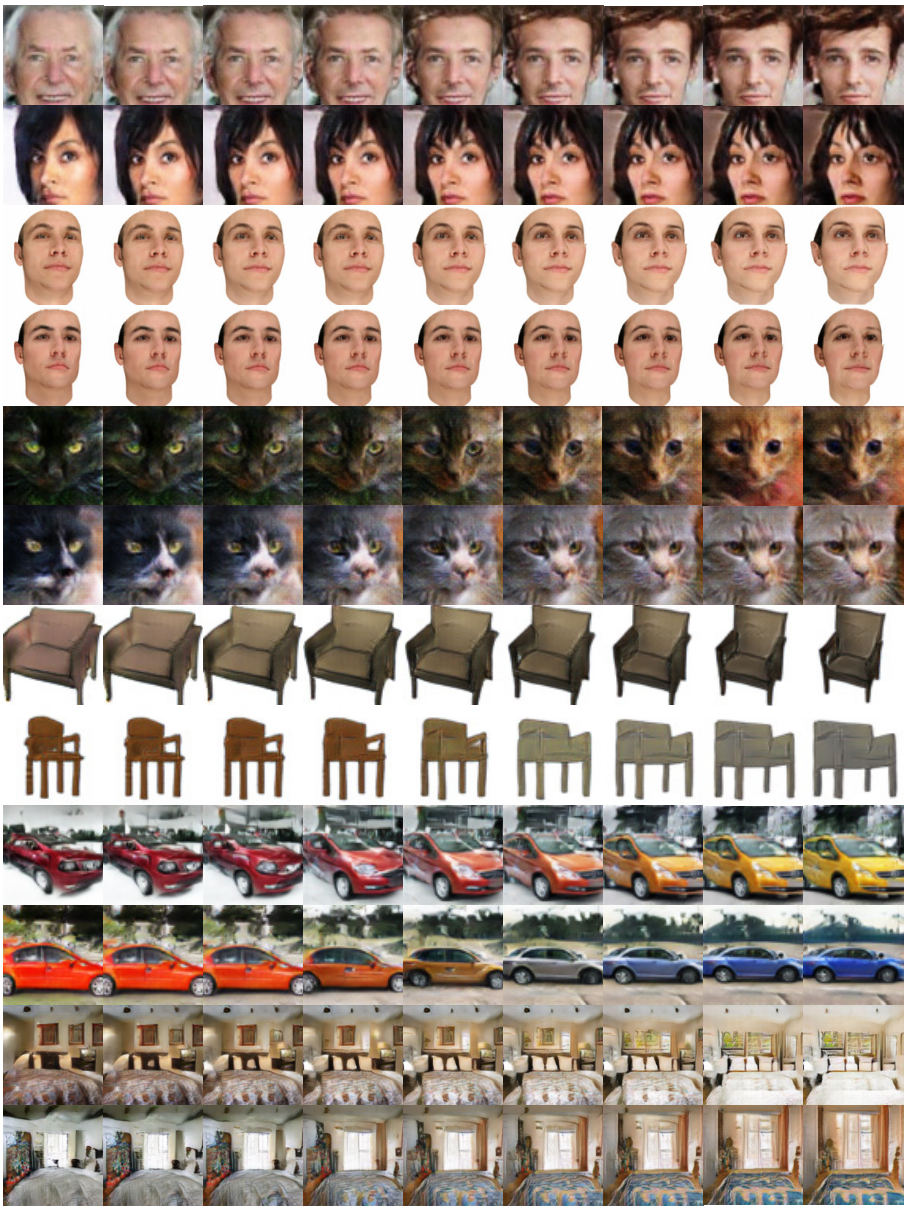

Figure 3. Linear interpolation of the latent vector $\mathbf{z}$ for fixed poses. Note that the identity is smoothly interpolated while the poses are unchanged.

Layer type	**Kernel size**	**Stride**	**Activation**	**Normalisation**	**Output dimension**
UpConv	$3\times3\times3$	2	LRelu	AdaIN	$8\times8\times8\times256$
UpConv	$3\times3\times3$	2	LRelu	AdaIN	$16\times16\times16\times128$
3D transformation	—	—	—	—	$16\times16\times16\times128$
Conv	$3\times3\times3$	1	LRelu	—	$16\times16\times16\times64$
Conv	$3\times3\times3$	1	LRelu	—	$16\times16\times16\times64$
Concatenate	—	—	—	—	$16\times16\times(16{\cdot}64)$
Conv	1×1	1	LRelu	—	$16\times16\times512$
UpConv	4×4	2	LRelu	AdaIN	$32\times32\times256$
UpConv	4×4	2	LRelu	AdaIN	$64\times64\times64$
UpConv	4×4	2	LRelu	AdaIN	$128\times128\times32$
UpConv	4×4	1	Tanh	—	$128\times128\times3$

For the 64×64 generator, we drop the last convolution layer, and use a convolution layer with output size 64×64×3 as the last layer instead. For the 128×128 discriminator, using either layers '(A)' for real/fake prediction or layers '(B)' for computing the identity loss:

Layer type	Kernel size	Stride	Activation	Normalisation	Output dimension
Conv	3×3	2	LRelu	IN/Spectral	64×64×128
Conv	3×3	2	LRelu	IN/Spectral	32×32×256
Conv	3×3	2	LRelu	IN/Spectral	16×16×512
Conv	3×3	2	LRelu	IN/Spectral	8×8×1024
(A) FC	—	—	Sigmoid	None/Spectral	1
(B) FC	—	—	LRelu	None/Spectral	128
(B) FC	—	—	Tanh	None/Spectral	128

7.1. Style discriminator

In this work, we use the style discriminators in addition to the image discriminator. Instead of classifying images as real or fake, *style discriminators* perform the same task but at the feature level across different layers. In particular, style discriminators classify the mean and standard deviation of the features at different levels l (which are believed to describe the image "style").

The style-discriminators are implemented as MLPs with sigmoid activation function for binary classification. For each feature channel c, the mean $\mu(\mathbf{\Phi}_l(\mathbf{x}))$ and variance $\sigma(\mathbf{\Phi}_l(\mathbf{x}))$ of features $\mathbf{\Phi}_l(\mathbf{x})$ are computed across batch and spatial dimensions independently as:

$$\mu(\mathbf{\Phi}_l(\mathbf{x})) = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{\Phi}_l(\mathbf{x})_{nhwc}. \tag{1}$$

$$\sigma(\mathbf{\Phi}_l(\mathbf{x})) = \sqrt{\frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (\mathbf{\Phi}_l(\mathbf{x})_{nhwc} - \mu(\mathbf{\Phi}_l(\mathbf{x})))^2 + \epsilon}. \tag{2}$$

8. Training details

We train all of our networks from scratch using the Adam optimizer [4]. We initialise all weights using $\mathcal{N}(0, 0.2)$, and biases as 0. We use $|\mathbf{z}| = 128$ for all datasets, apart from the Cars dataset, where we use $|\mathbf{z}| = 200$. Empirically, we find that updating the generator twice for every update of the discriminator achieves images with the best visual fidelity.

8.1. Pose sampling

During training, we sample poses uniformly to transform the 3D features. The range of views from which we sample from are listed in Section 6. Note that for the face datasets (Basel, CelebA, Cats), we only sample the frontal hemisphere since the training images do not include views of the back.

8.2. Learning rate

We use the same learning rate for both the generator G and the discriminator D. In particular, to train images at resolution 64×64 pixels, we use the learning rate of 0.0001, and for images at resolution 128×128 pixels, the learning rate is 0.00005. We keep the same learning rate for the first half of total number of epochs (we train each dataset for 25 epochs), and linearly decay the rate to zero over the remaining half.

References

[1] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. arXiv:1512.03012, 2015. 2

[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs. In *NIPS*, pages 5767–5777, 2017. 2

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[5] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2

[6] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 2

[7] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 2

[8] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2

[9] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981, 2015. 2

[10] 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:1506.03365, 2015. 2

[11] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection – how to effectively exploit shape and texture features. In *ECCV*, 2008. 2

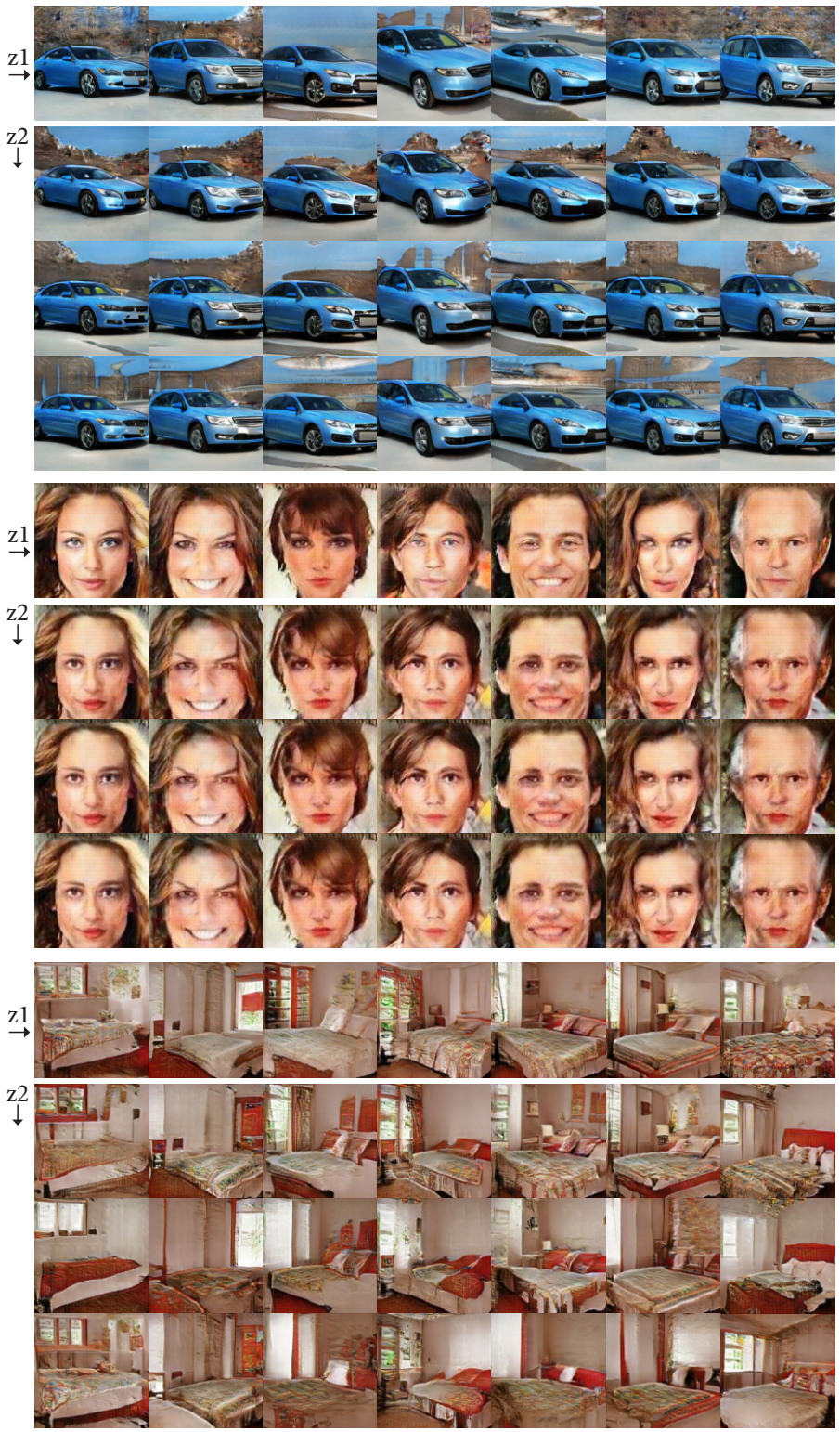

Figure 4. We keep the same $\mathbf{z}_1$ (columns, controlling 3D features at $8\times8\times8$ resolution) and $\mathbf{z}_3$ (controlling 2D features), and sample different $\mathbf{z}_2$ (rows, controlling 3D features at $16\times16\times16$ resolution). We do not include label for $\mathbf{z}_3$ in the figure.

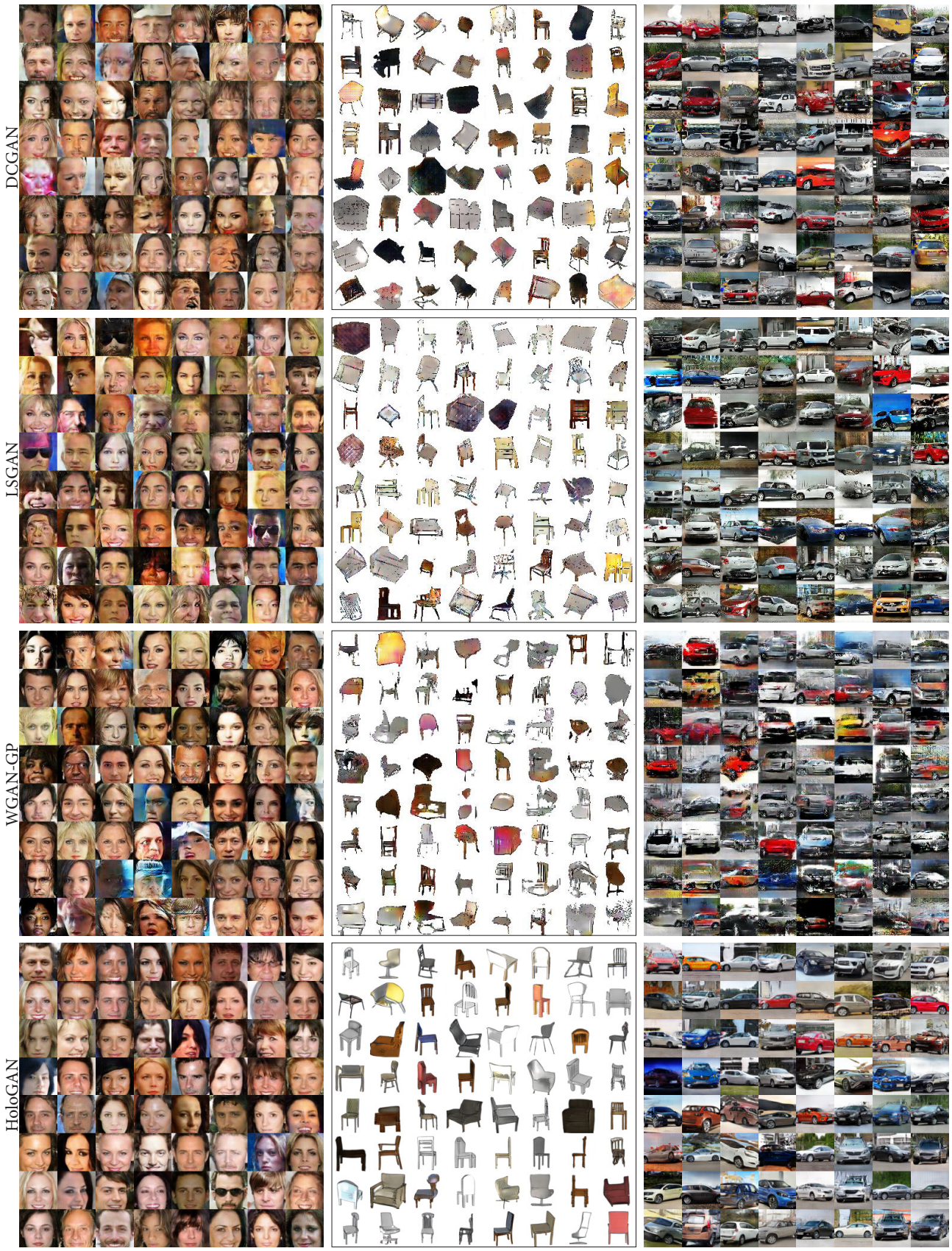

Figure 5. Samples from models used to compute the KID score for CelebA (left), Chairs (middle) and Cars (right).