Disentangled3D: Learning a 3D Generative Model with Disentangled Geometry and Appearance from Monocular Images
—-Supplementary Document—-

Ayush Tewari[1,2] Mallikarjun B R[1] Xingang Pan[1] Ohad Fried[3]
Maneesh Agrawala[4] Christian Theobalt[1]

[1]Max Planck Insitute for Informatics [2]MIT [3]Interdisciplinary Center, Herzliya [4]Stanford University

In this document, we will present more implementation details (Sec. 1), and present more results (Sec. 2) of our method. We encourage the reader to watch the supplementary video for more results.

1. Training Details

Network Architecture Our generator network consists of a geometry deformation network $\mathbf{N}_G$, an appearance network $\mathbf{N}_A$, and a canonical geometry network $\mathbf{N}_C$. Both $\mathbf{N}_G$ and $\mathbf{N}_A$ include a mapping network and a main network following the design of π-GAN [1]. The mapping networks are implemented as MLPs with LeakyReLU activations, see Table 1. The randomly sampled inputs $\mathbf{z}_G \in \mathbb{R}^{256}$ and $\mathbf{z}_A \in \mathbb{R}^{256}$ are used as inputs to the mapping networks. The output of the mapping networks are one-dimensional vectors of dimensions $256 \times 2 \times d_G$ and $256 \times 2 \times d_A$, where d_G and d_A are the number of SIREN layers in the main networks of $\mathbf{N}_G$ and $\mathbf{N}_A$ respectively. The main networks are implemented as MLPs with SIREN layers [10] and FiLM conditioning [8], see Table 3 and Table 4. Each layer of the main network receives one 256×2-dimensional component of the output of the mapping network. The canonical network $\mathbf{N}_C$ does not receive any input other than the co-ordinates in the canonical space. We follow the initialization method of [10] for $\mathbf{N}_G$, $\mathbf{N}_A$, and $\mathbf{N}_C$, where the first layer is initialized with larger values. The final layer of $\mathbf{N}_G$ is initialized such that the deformations at the first iteration are all zeros. The inverse deformation network is implemented exactly as $\mathbf{N}_G$, except that it receives the input in the canonical space and models the inverse deformation. As for the discriminator, we adopt the same model architecture as in [1], which is a convolutional neural network with residual connections [4] and CoordConv layers [6].

As explained in the main paper, we control the level of disentanglement using the number of SIREN layers in $\mathbf{N}_G$ and $\mathbf{N}_A$, i.e., d_G and d_A, respectively. We set $d_G = 5$ and $d_A = 3$ for FFHQ [5], VoxCeleb2 [2], and Cats [12]. For Carla [3], we set $d_G = 3$ and $d_A = 6$. We will show results where changing the relative depths of these networks can lead to poor disentanglement.

Input	Layer	Activation	Output Dim.
$\mathbf{z}_G$ or $\mathbf{z}_A$	Linear	LeakyReLU (0.2)	256
-	Linear	LeakyReLU (0.2)	256
-	Linear	LeakyReLU (0.2)	256
-	Linear	None	$256 \times 2 \times (d_G$ or $d_A)$

Table 1. Mapping Network, denoted as Map($\cdot$). We use a separate mapping network for the geometry and appearance networks.

Input	Layer	Activation	Output Dim.
$\mathbf{x}'$	Linear	Sine	256
-	Linear	Sine	256
-	Linear	Sine	256
-	Linear	Sine	256
-	Linear	None	1

Table 2. Canonical Network, denoted as $\mathbf{N}_C(\cdot)$. The input $\mathbf{x}'$ is a point in the canonical space, computed using the goemetry deformation network.

Input	Layer	Activation	Output Dim.
$\mathbf{x}$, Map($\mathbf{z}_G$)	Linear	FiLM+Sine	256
-, Map($\mathbf{z}_G$)	...	...	...
-, Map($\mathbf{z}_G$)	...	...	...
-, Map($\mathbf{z}_G$)	Linear	None	3

Table 3. Geometry Deformation Network, denoted as $\mathbf{N}_G(\cdot)$. The input $\mathbf{x}$ is a point in the deformed or world space. The output can be added to $\mathbf{x}$ to compute $\mathbf{x}'$, the corresponding 3D point in the canonical space. The output of the shape mapping network is also provided as input for each layer.

Hyperparameters We describe the hyperparamters used in our method in Table 5. The training curriculum is de-

Input	Layer	Activation	Output Dim.
$\mathbf{x}'$, Map($\mathbf{z}_A$)	Linear	FiLM+Sine	256
-, Map($\mathbf{z}_A$)	...	...	...
-, Map($\mathbf{z}_A$)	...	...	...
-, Map($\mathbf{z}_A$), $\mathbf{d}$	Linear	FiLM+Sine	256
-, Map($\mathbf{z}_A$)	Linear	Sigmoid	3

Table 4. Appearance Network, denoted as $\mathbf{N}_A(\cdot)$. The input $\mathbf{x}'$ is a point in the canonical space, computed using the goemetry deformation network. The output is the color at this point. The other inputs are the output of the color mapping network, and the viewing direction.

Hyperparameter	Dataset	Value
λ	FFHQ	1.0
	VoxCeleb2	1.0
	Cats	0.5
	Carla	10.0
λ_{pose}	FFHQ	50.0
	VoxCeleb2	50.0
	Cats	5.0
	Carla	50.0
λ_{img}	FFHQ	0.001
	VoxCeleb2	0.001
	Cats	0.001
	Carla	0.001
λ_{inv}	FFHQ	1.0
	VoxCeleb2	1.0
	Cats	1.0
	Carla	1.0

Table 5. Hyperparameters of our method.

scribed in Table 6. Our networks are trained in a coarse-to-fine manner.

Embedding Architecture Our encoder network consists of a pretrained ResNet-18 [4] as the backbone. We add two linear layers to regress the camera pose and latent vectors. Inspired by π-GAN [1], we learn to directly regress the frequencies and phase shifts, i.e., the output space of the mapping networks for the geometry and appearance components. We train the encoder on FFHQ [5]. We set $\lambda_{\text{perc}} = 1$ and $\lambda_{\text{reg}} = 10$ and use a learning rate of 0.01.

At test time, to further improve the result, we fine-tune the regressed latent vectors using iterative optimization for $1.8k$ iterations with a learning rate of 0.01. We finally fine-tune the generator network for another 200 iterations with a learning rate of 0.0001. We show that this strategy leads to high-quality results without degrading the disentanglement properties (see Fig. 7) of the generator.

We also show that this approach works better than optimization-only method (see Fig. 6), where we iteratively optimize for the latent vectors and camera pose using reconstruction loss. For optimization-only approach, we update the latent vectors and camera pose while keeping the GAN fixed for $1.8k$ iterations with a learning rate of 0.01. And then finetune GAN as well for another 200 iterations with a learning rate of 0.0001. We can observe (Fig. 6, 7) that using encoder initialization helps obtain better results while still preserving the disentanglement properties of our model.

2. Results

Qualitative Results We show more results of our method along with visualizations of the learned canonical volume in Fig. 1. We present more visualizations of the learned correspondences in Fig. 2. The appearance of one sample is transferred to another using the correspondences. This shows the applicability of the correspondences for any task where one image annotation can be transferred to all other samples of the model. As mentioned earlier, the level of disentanglement is controlled using the relative depths of the geometry and appearance networks. We show in Fig. 3 that a large appearance network can lead to lower-quality disentanglement, where geometric features such as expressions are compensated by the appearance component. We set $d_G = 3$ and $d_A = 5$ for these results. In the main paper, we presented quantitative results of a baseline where the canonical network receives a high-dimensional input like GRAF [9]. Fig. 4 shows qualitative results of this baseline. As explained in the main paper, this baseline has similar limitations as GRAF, where the geometry network also changes the appearance of the object. Fig. 5 shows more results for evaluation of the pose regularization. Without our proposed regularization, the model does not properly disentangle the object and the camera pose. This limitation is also shared with π-GAN [1]. We further show some results of correspondence and depth visualizations on real images in Fig. 8. Unlike the encoders used in other results, we trained the encoder for this result on the generator which was trained with the inverse network. We also compare to GIRAFFE [7] in Fig. 9. Our method maintains the consistency of both pose and shape components better. Quantitatively, GIRAFFE achieves similar scores compared to our method on FFHQ using the metrics defined in the main paper. It achieves an appearance consistency score of 0.05, geometry consistency score of 0.32, and appearance variation score of 0.09. However, ours results have better multi-view consistency, and better qualitative disentanglement as shown in Fig. 9. We show several more results of our GAN in Fig. 10.

Quantitative results We present FID scores for FFHQ [5], VoxCeleb2 [2], and Cats [12] evaluated at

Dataset	Iteration (in k)	Batch Size	Image Size	G_{lr}	D_{lr}
FFHQ	0-20	208	32	2e-5	2e-4
	20-60	52	64	2e-5	2e-4
	60-	52	64	1e-5	1e-4
VoxCeleb2	0-20	208	32	2e-5	2e-4
	20-60	52	64	2e-5	2e-4
	60-	52	64	1e-5	1e-4
Cats	0-10	208	32	6e-5	2e-4
	10-	52	64	6e-5	2e-4
Carla	0-10	60	32	4e-5	4e-4
	10-26	20	64	2e-5	2e-4
	26-	18	128	10e-6	10e-5

Table 6. Training curriculum

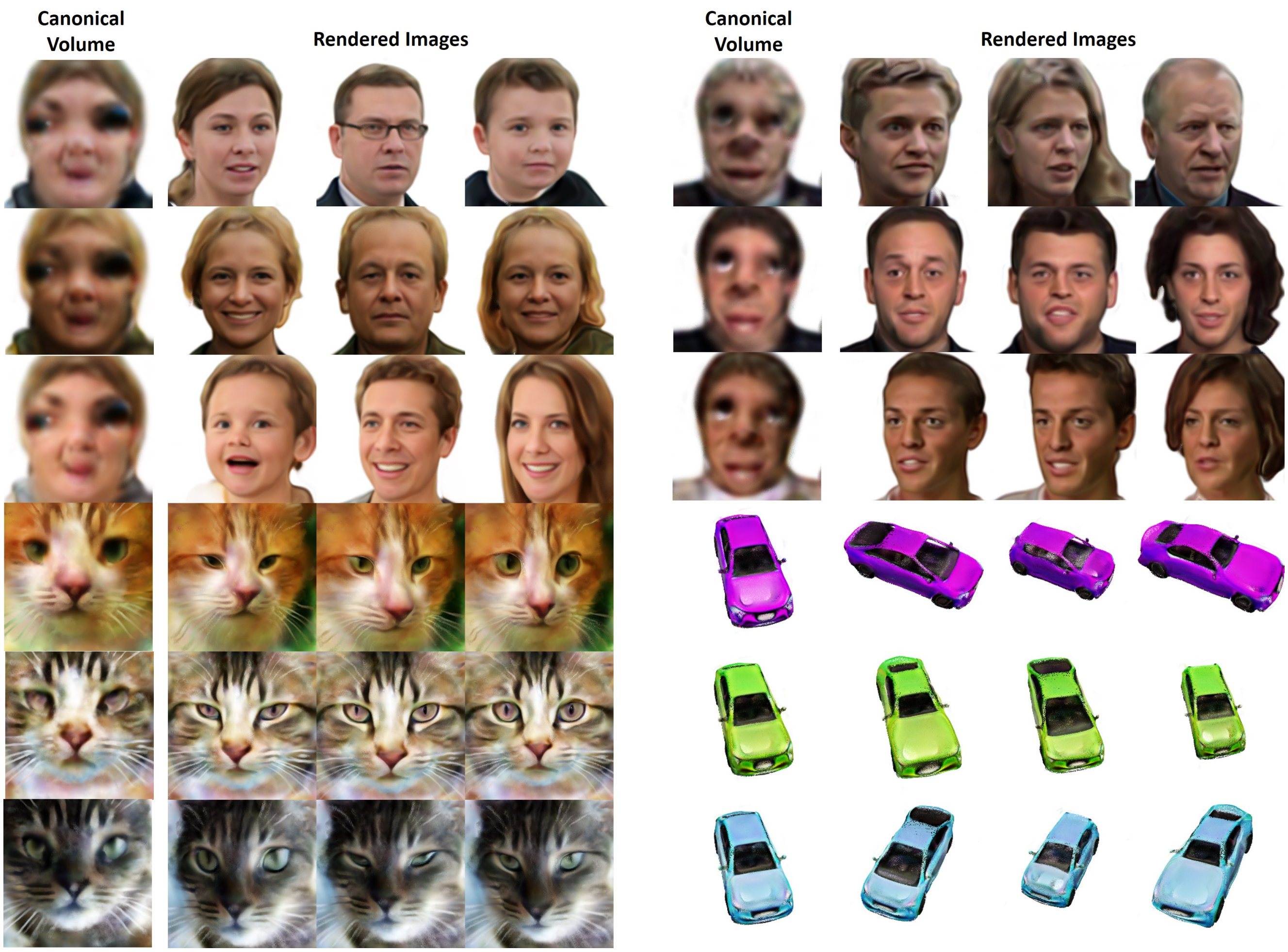

Figure 1. Results of our method on FFHQ (top-left), VoxCeleb2 (top-right), Cats (bottom-left) and Carla (bottom-right). Each row shows the canonical volume, and multiple rendered images with the same appearance and pose, but with different geometry. All canonical volumes for a dataset are rendered from the same pose. Notice that only the color of the canonical volume changes.

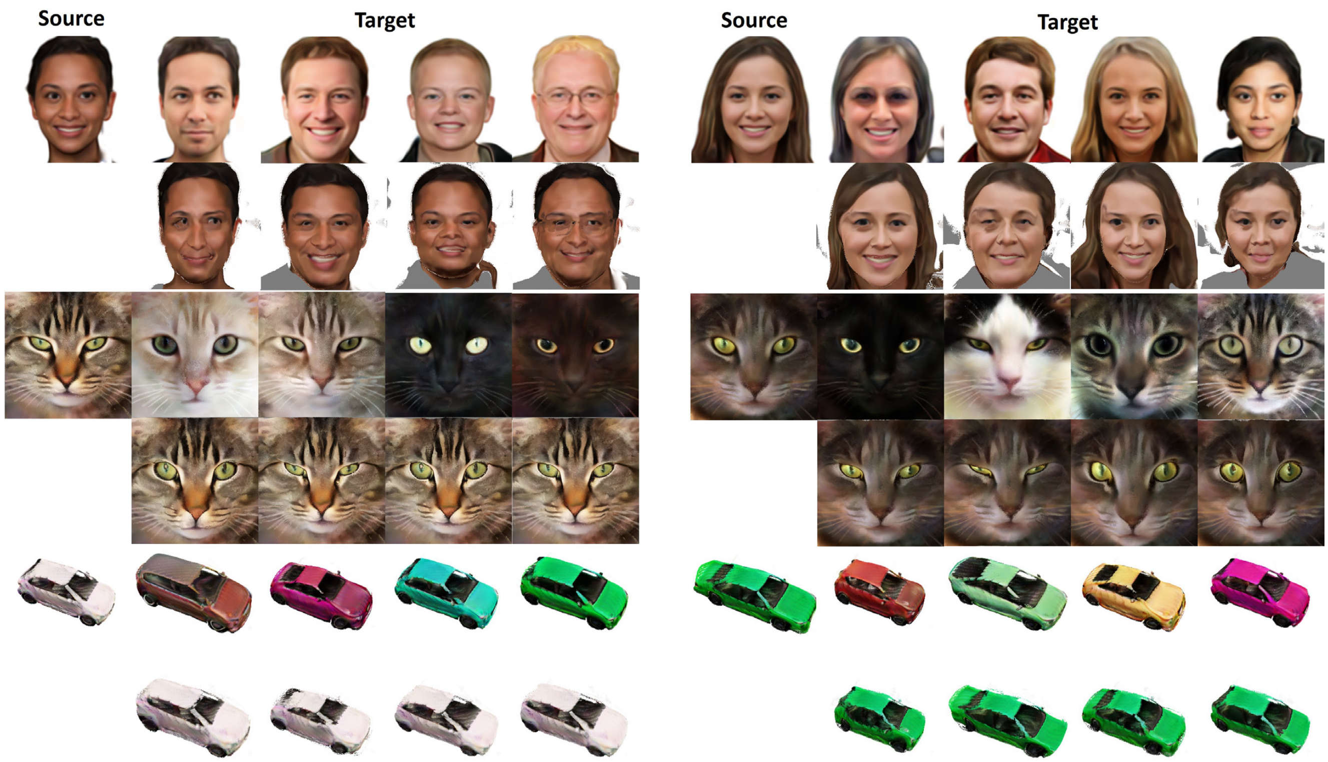

Figure 2. Appearance transfer using the learned correspondences. For each object class, the first row shows different random samples from our GAN. The left-most sample is used as the source texture. This texture is transferred to all other samples, visualized in the second row. Note that we only the source image, and not the full 3D model, in order to visualize pixel-to-pixel correspondences. We can faithfully transfer the source appearance while preserving the target geometry. Also note that not all pixels in the target image have a valid correspondence to the source image. For example, if the shirt is not visible in the source image, the shirt pixels in the target image do not have a valid correspondence. Thus, only the pixels whose corresponding points are visible in the source image achieve the correct appearance transfer. This visualization shows the applicability of our approach to various applications, such as one-shot semantic segmentation and sparse keypoint detection.

Figure 3. Results on FFHQ with a larger appearance network. Each row shows results with a fixed geometry and different appearances. With a large appearance network, geometric features such as expressions can be compensated incorrectly by the appearance component.

Figure 4. Results of the 256-baseline on FFHQ. Each row shows results with a fixed appearance and different geometry. This baseline uses a 256-dimension vector as input to the canonical volume. This results in poor disentanglement, where changing the geometry also changes the appearance. GRAF [9] uses a similar design choice and thus, suffers from the same limitation.

Figure 5. Evaluation of our pose regularization loss on VoxCeleb2. All images are rendered with a fixed frontal camera. Without pose regularization, the model cannot disentangle between the scene and the camera pose. This issue is also evident in pi-GAN.

Figure 6. Here we show that our embedding method which uses encoder output as initialization (row 3) results in higher-quality output (row 4) compared to optimization-only approach (row 2) for real in-the-wild input images (row 1).

	FFHQ	VoxCeleb2	Cats
GRAF [9]	25.36	21.76	18.26
Ours	**15.87**	**8.86**	**12.35**

Table 7. Quantitative comparisons using the FID score metric (a lower value is better) at 64×64 image resolution. We outperform GRAF on all datasets.

64×64 image resolution in Table 7. All FID scores are calculated using 8k samples. We also present a quantitative evaluation of the pose regularization loss in Table 8. Specifically, we first render 1000 images from each method with a fixed camera. We then compute the head pose in the rendered results using the Model-based Face Autoencoder (MoFA) [11] method. The pose consistency metric is

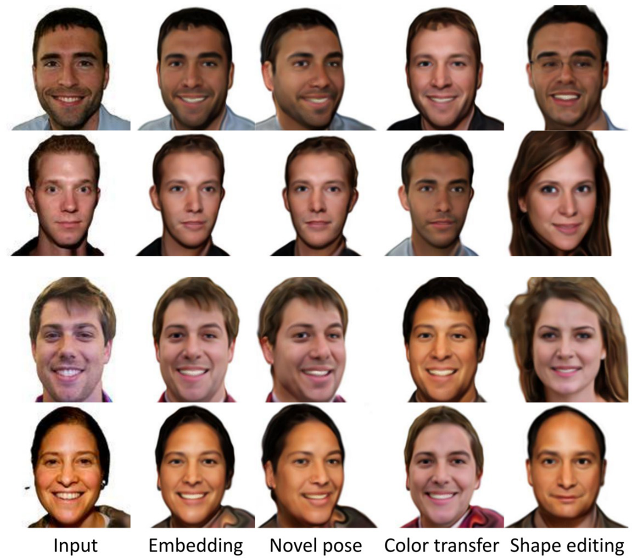

Figure 7. Given real images (col 1), we can embed them in our GAN space (col 2). This enables novel view synthesis (col 3), color transfer from the other real image (col 4), or shape editing using a random sample from the GAN. For color transfer results in col 4, we transfer the embedded color between 2 pairs (rows 1,2 and rows 3,4).

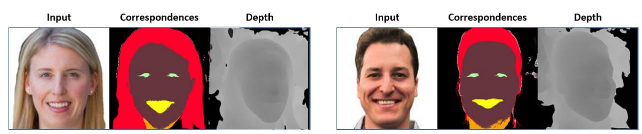

Figure 8. Results on real images. Reference from Fig.5-main is used for correspondences. Depth is rendered from a novel view.

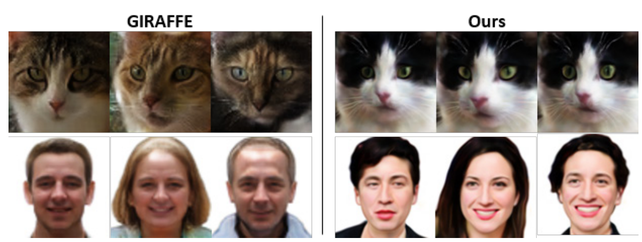

Figure 9. Comparisons with GIRAFFE. Visualized are three images with the same appearance code but different geometry codes.

computed as the standard deviation over the yaw angles. A lower number indicates good disentanglement of the camera pose and the 3D object. We can see that the proposed pose regularization loss significantly improves such disentanglement.

References

[1] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative ad-

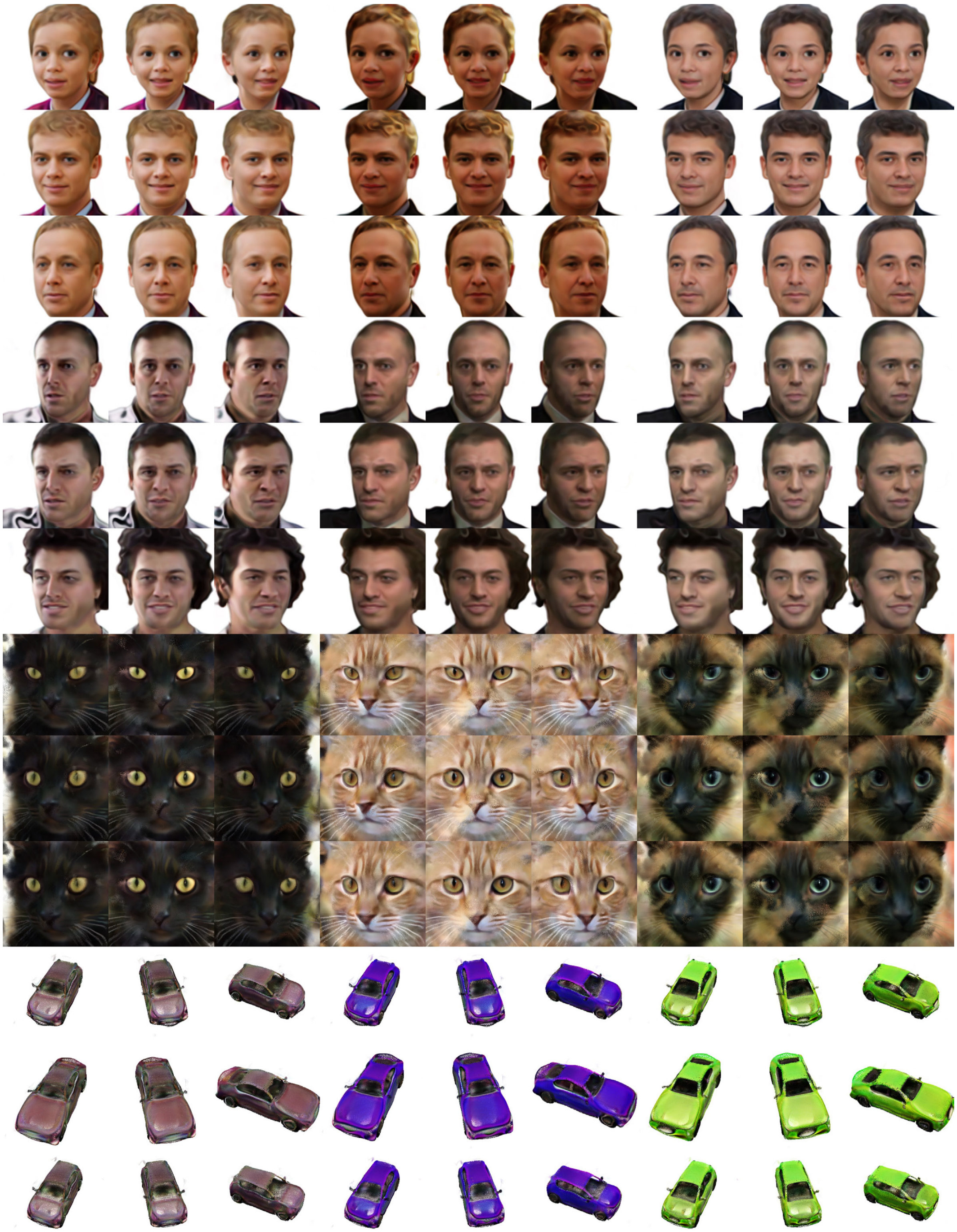

Figure 10. More results of our method on FFHQ (rows 1-3), VoxCeleb2 (rows 4-6), Cats (rows 6-8) and Carla (rows 10-12). Each row shows a fixed geometry with three different appearances and poses.

	Pose Consistency ↓
pi-GAN	0.34
Ours (no pose reg.)	0.16
Ours	**0.03**

Table 8. Quantitative evaluation of pose consistency. Pose consistency is measured as the standard deviation of the 3D yaw-component of head pose computed over 1000 images rendered from a fixed camera. The pose regularization significantly improves pose consistency, helping disentangle the camera pose from the scene.

versarial networks for 3d-aware image synthesis. In *arXiv*, 2020. 1, 2

[2] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 1, 2

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 1, 2

[5] 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 (CVPR)*, June 2019. 1, 2

[6] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018. 1

[7] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[8] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 1

[9] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 4, 5

[10] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 1

[11] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017. 5

[12] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pages 802–816. Springer, 2008. 1, 2