pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis

–Supplementary Material–

1. Novel View Synthesis Details

We demonstrate a potential application of π-GAN: we can use a trained generator, without modifications, to perform single-view reconstruction. We base our method on the inverse projection procedure outlined by Karras et al. [4].

We freeze the parameters of our implicit representation and seek the frequencies $\boldsymbol{\gamma}_i$ and phase shifts $\boldsymbol{\beta}_i$ for each MLP layer i which produce a radiance field that, when rendered, best matches the target image. We initialize $\boldsymbol{\gamma}_i$ and $\boldsymbol{\beta}_i$ to $\bar{\boldsymbol{\gamma}}_i$ and $\bar{\boldsymbol{\beta}}_i$, the center of mass of frequencies and phase shifts for each layer. We calculate $\bar{\boldsymbol{\gamma}}_i$ and $\bar{\boldsymbol{\beta}}_i$ simply by averaging the frequencies and phase shifts of ten thousand random noise vector inputs. We then run gradient descent to minimize the mean-squared-error image reconstruction loss. We additionally introduce an $\mathcal{L}_2$ penalty with a weight of 0.1 during the optimization process to prevent $\boldsymbol{\gamma}_i$ and $\boldsymbol{\beta}_i$ from straying too far from $\bar{\boldsymbol{\gamma}}_i$ and $\bar{\boldsymbol{\beta}}_i$. We optimize the frequencies and phase shifts with the Adam optimizer over 700 iterations. We initialize the learning rate to 0.01, decaying by a factor of 0.5 every 200 iterations.

2. Model Details

Mapping Network. The mapping network is parameterized as an MLP with three hidden layers of 256 units each. The mapping network uses leaky-ReLU activations with a negative slope of 0.2.

SIREN-based Implicit Radiance Field. The FiLMed-SIREN [9] backbone of the generator is parameterized as an MLP with eight FiLMed-SIREN hidden layers of 256 units each.

Discriminator. Table 1 shows the architecture of the progressive discriminator. We begin training at low resolutions and progressively add discriminator stages while upsampling image size. In order to smooth transitions between upsamples, we fade in the contributions of new layers over ten-thousand iterations. We utilized CoordConv layers [6] and residual connections [2] throughout the discriminator.We considered using a patch discriminator similar to GRAF, but found it leads to uneven image quality as SIREN is prone to local overfitting to the last batch if sufficient coverage of the space is not maintained.

Table 1. Discriminator architecture, showing progressive growing stages.

	Activation	Output Shape
Input Image	-	3×128×128
Adapter Block (1×1)	LeakyReLU (0.2)	64×128×128
Coord Conv 1 (3×3)	LeakyReLU (0.2)	128×128×128
Coord Conv 2 (3×3)	LeakyReLU (0.2)	128×128×128
Avg Pool Downsample	-	128×64×64
Coord Conv 1 (3×3)	LeakyReLU (0.2)	256×64×64
Coord Conv 2 (3×3)	LeakyReLU (0.2)	256×64×64
Avg Pool Downsample	-	256×32×32
Coord Conv 1 (3×3)	LeakyReLU (0.2)	400×32×32
Coord Conv 2 (3×3)	LeakyReLU (0.2)	400×32×32
Avg Pool Downsample	-	400×16×16
Coord Conv 1 (3×3)	LeakyReLU (0.2)	400×16×16
Coord Conv 2 (3x3)	LeakyReLU (0.2)	400×16×16
Avg Pool Downsample	-	400×8×8
Coord Conv 1 (3×3)	LeakyReLU (0.2)	400×4×4
Coord Conv 2 (3×3)	LeakyReLU (0.2)	400×4×4
Avg Pool Downsample	-	400×2×2
Conv 2d (2×2)		1×1×1

Table 2. FID, KID mean × 100, and IS for π-GAN on CelebA, Cats, and CARLA datasets.

	FID ↓	KID ↓	IS ↑
CelebA @ 64 × 64	5.15	0.09	2.28
Cats @ 64 × 64	7.36	0.23	2.07
CARLA @ 64 × 64	13.59	0.34	3.85

Figure 1. COLMAP reconstructions for models trained on CelebA, obtained by running COLMAP with default parameters and no known camera poses; GRAF's results were from their supplement.

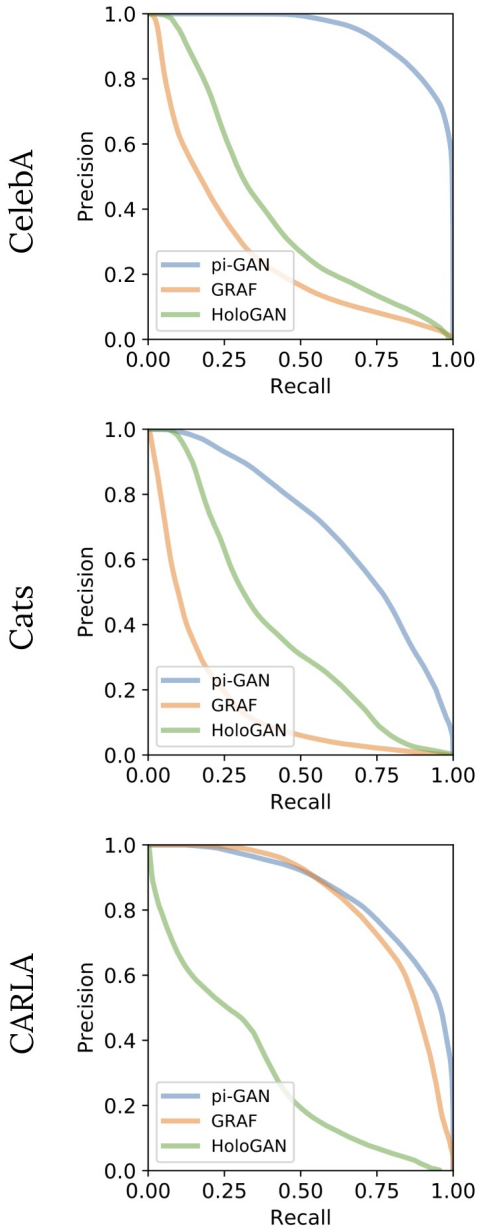

Figure 2. Precision-recall plots for π-GAN, GRAF, and HoloGAN on CelebA, Cats, and CARLA.

3. Additional Training Details

We train the majority of our models across two RTX 6000 GPUs. We begin training at a resolution of 32×32, with an initial batch size of 120. At each upsample, we drop the batch size by a factor of four to keep the models and generated images in memory. At higher resolutions, we aggregate across mini-batches to keep an effective batch size at or above 12, given our GPU constraints. To further reduce memory usage, we used PyTorch's Automatic Mixed Precision (AMP). π-GAN trained for 10 hours at 32×32, 10 hours at 64×64, and 36 hours at 128×128. Certain rendering and camera parameters were tuned according to the dataset. We use the true pose distribution when it is known, e.g. for synthetic datasets, otherwise we make a guess and tune the distribution as a hyperparameter. We sample camera poses for CelebA from a normal distribution, with a vertical standard deviation of 0.15 radians and a horizontal standard deviation of 0.3 radians. We sample camera poses for Cats from a uniform distribution, with horizontal range $(-0.75, 0.75)$ and vertical range $(-0.4, 0.4)$. We sample poses for CARLA uniformly from the upper hemisphere. We tune the number of samples along each ray to balance memory consumption and depth resolution. We use 24 samples per ray for CelebA and Cats and 64 samples per ray for CARLA. We utilize a pinhole perspective camera with a field of view of 12^o for CelebA, 12^o for Cats, and 30^o for CARLA.

4. π-GAN results @ 64×64

Table 2 includes additional quantitative results, evaluated at 64×64, in order to allow for comparisons of π-GAN against models evaluated at lower resolutions.

5. Additional Visual Results

We include additional visual results to show the image quality and view consistency of π-GAN. Figures 4 and 5 demonstrate the wide range of camera poses supported by π-GAN for generated faces and cats. Figure 3 shows the fine detail that π-GAN renders on larger images. Figure 6 shows additional cars with varying elevation and rotation. We include several videos of faces and cats with the camera following an elliptical trajectory in our supplementary video.

6. COLMAP Reconstruction

In order to demonstrate the images from π-GAN are multi-view consistent, we include a COLMAP reconstruction in Figure 1. We observe that proxy shapes extracted from pi-GAN lead to more pleasing novel views when projected to novel camera poses than those from GRAF.

7. Interpolation and Truncation

Following the method of StyleGAN [3] we can smoothly interpolate between two generated samples by linearly interpolating between the frequencies and phase shifts corresponding to the two latent codes. We include a result in Figure 8 in the paper. Along similar lines, it is also possible to trade off fidelity and diversity at test time following the method proposed in StyleGAN [3]. Because truncation reduced the diversity of generated images, we provided all evaluation metrics without truncation.

8. Precision and Recall

Recent work in generative models have investigated alternative metrics in order to independently evaluate fidelity and diversity [8, 5]. Figure 2 provides precision-recall plots on CelebA, Cats, and CARLA, comparing π-GAN to GRAF and HoloGAN.

References

[1] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Proc. CoRL*, 2017. 6

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1

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

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 1

[5] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Proc. NeurIPS*, 2019. 3

[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 *Proc. NeurIPS*, 2018. 1

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015. 4

[8] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Proc. NeurIPS*, 2018. 3

[9] 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

[10] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - how to effectively exploit shape and texture features. In *Proc. ECCV*, 2008. 5

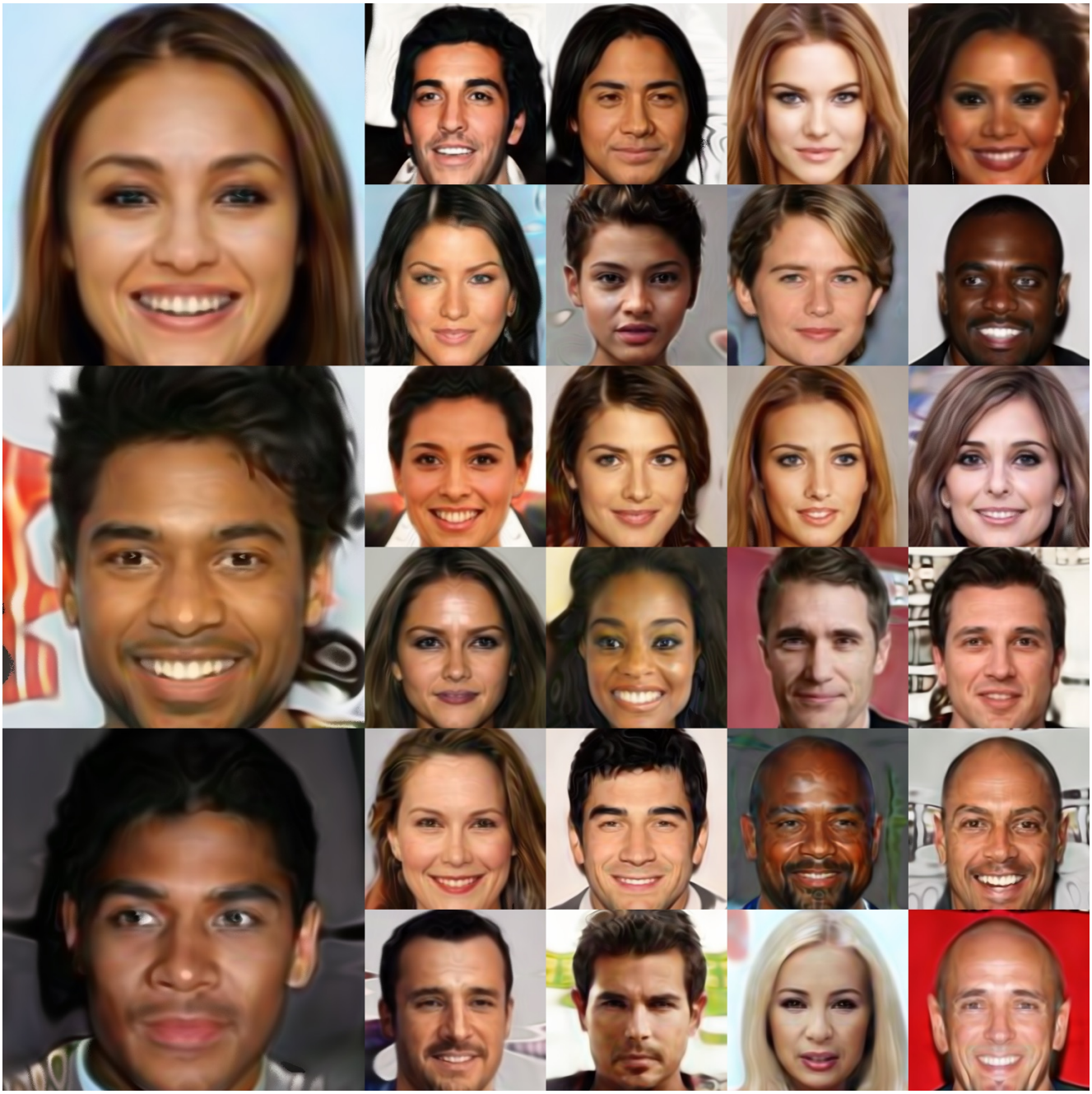

Figure 3. Curated examples from our model trained with CelebA [7].

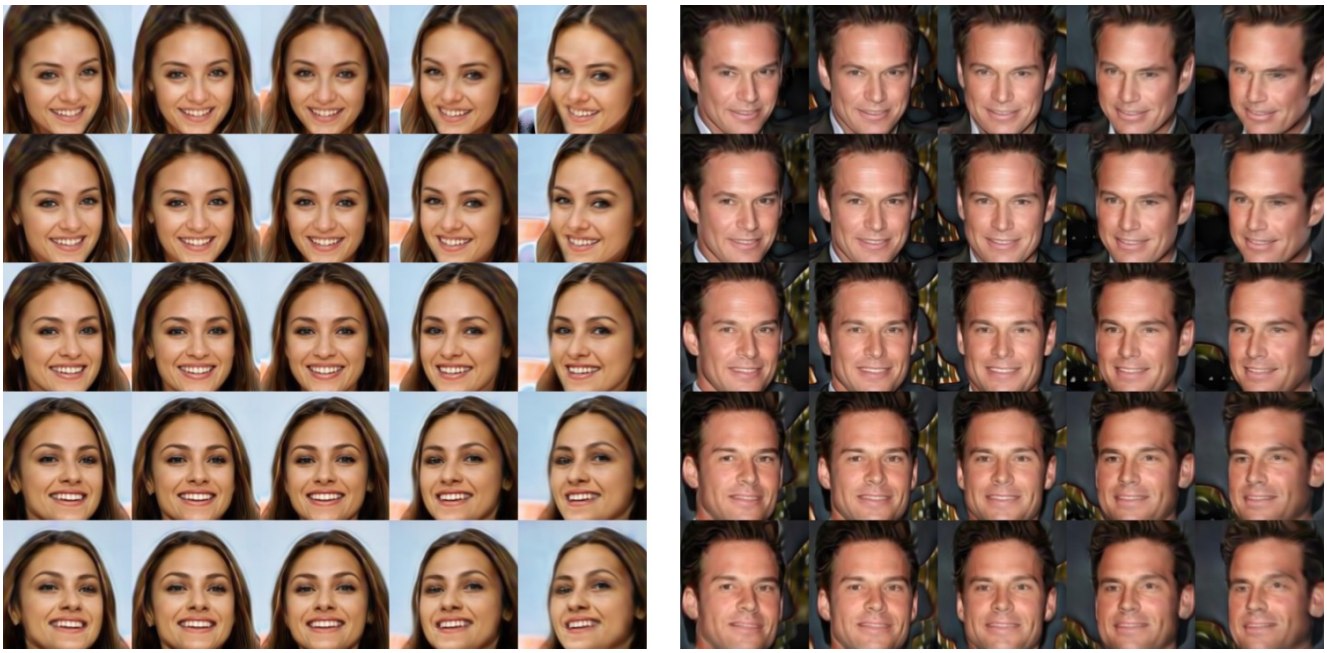

Figure 4. Curated examples from our model trained with CelebA, displayed from multiple viewing angles.

Figure 5. Curated examples from our model trained with Cats [10], displayed from multiple viewing angles.

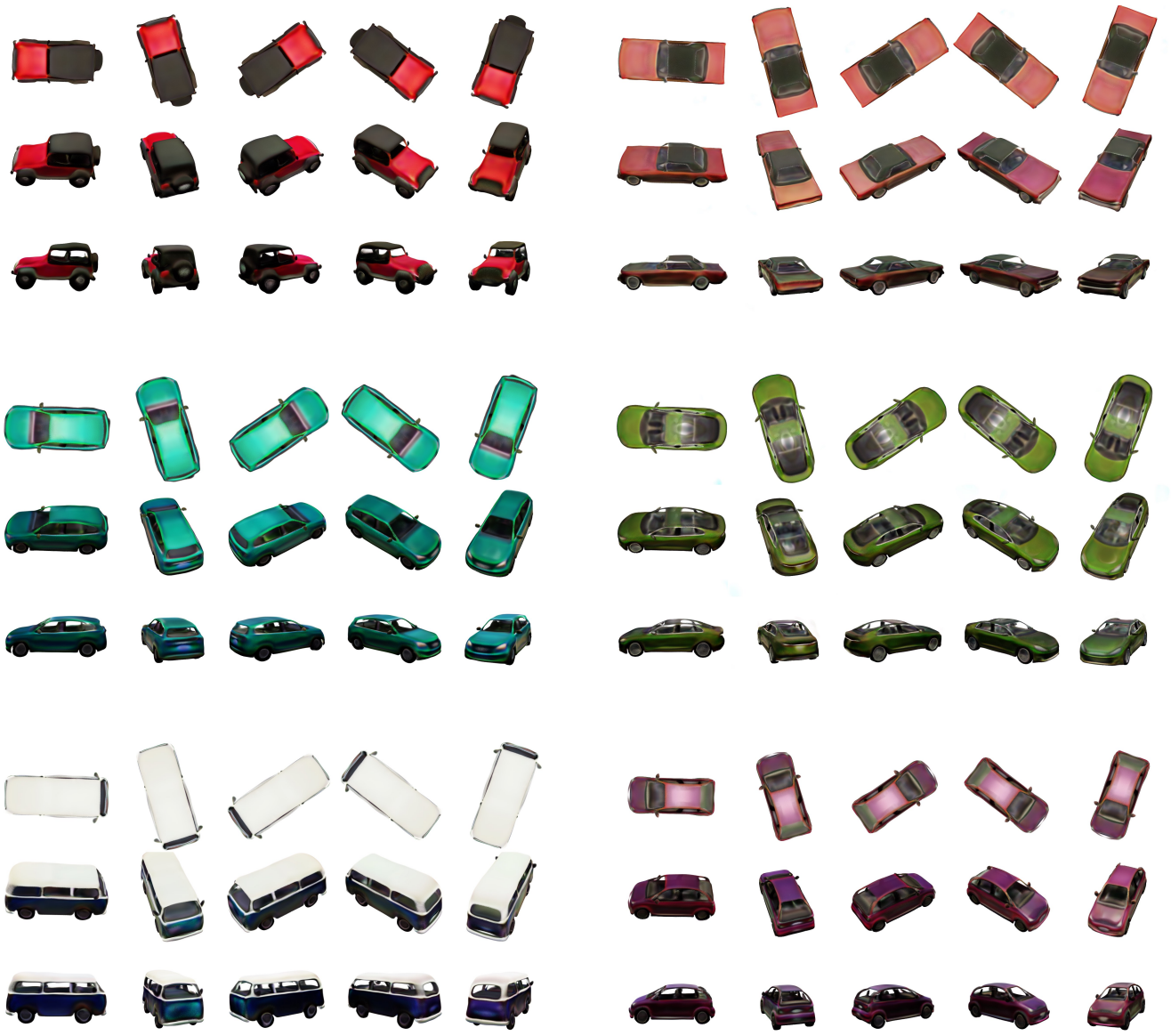

Figure 6. Curated examples from our model trained with CARLA [1], displayed from multiple viewing angles.