Supplementary Material for Learning 3D-Aware GANs from Unposed Images with Template Feature Field

Xinya Chen[1], Hanlei Guo[1], Yanrui Bin[2], Shangzhan Zhang[1], Yuanbo Yang[1], Yue Wang[1], Yujun Shen[3], Yiyi Liao[1⋆]

[1]Zhejiang University [2]The Hong Kong Polytechnic University [3]Ant Group

A Implementation Details

We evaluate on four datasets, including Shapenet Cars [1,2], CompCars [5], SDIP Elephant [4], and LSUN Plane [6]. CompCars contains 136k unposed images capturing the entire cars with different styles. The original dataset contains images with different aspect ratios. We preprocess the images by center cropping, padding to the squared images with the same length, and resizing them to 256×256. We use the mask to set the background black and filter the data with bad mask estimation and extreme scale, leading to 110k images. LSUN Plane is a dataset that contains unposed images of different planes. We use MMDetection [3] to detect the plane and filter the plane larger than 226×226 resolution and the occluded plane, leading to 130k images. We rescale the plane to make the large side equal to 226 and padding it to 256 resolution.

B Runtime Analysis

Runtime Breakdown: We report the average runtime of different processes during training in Tab. 1. The runtime analysis is conducted on an A6000 GPU. The "Template Rendering" indicates the time to render the template feature field at discretized azimuth and elevation angles θ and ϕ to obtain 2D template features $\{\overline{\mathbf{F}}\}_{k=1}^{N_\theta \times N_\phi}$. The batch size to render the template is 32. We update the template and render it once every 16 iterations before 3k iterations and then once every epoch. The template rendering time in the early stage is averaged over 16 iterations. Since the iterations of each epoch are different for different datasets, we report the averaged template rendering time in the late stage using the dataset of the smallest amount of images, i.e. SDIP Elephant dataset. The "Phase Correlation" refers to the time for estimating the scale r and in-plane rotation γ, and warping each feature template to yield $\{\tilde{\mathbf{F}}\}_{k=1}^{N_\theta \times N_\phi}$. The "Camera Pose Sampling" includes the time to calculate mean square error and to perform inverse sampling (see Eq. 2 and Eq. 3 of the main paper). "Training" indicates the training time without camera pose estimation, which includes the data loading

⋆ Corresponding author.

time, the network forward time, and the optimization time. The batch size for training is 4. Note that we use a fixed camera pose and do not perform pose estimation after 500k iterations. Compared to the overall iterations of 6250k, the increased time is acceptable.

Runtime Comparison with Naïve Grid Search: As mentioned in our main paper, one can implement a naïve grid search method by discretizing all four variables $(\theta, \phi, \gamma, r)$ we consider for the camera poses. We demonstrate that this naïve approach significantly increases the pose estimation time in Tab. 2. Here, we further discretize scale r and in-plane rotation γ, each discretized into 256 values, increasing the total amount of 2D feature templates by 256^2 times. While this omits the phase correlation module, it significantly increases the template rendering time and is hence intractable for training the 3D GAN.

Process	Template Rendering Early Stage	Template Rendering Late Stage	Phase Correlation	Camera Pose Sampling	Training
Time (s/iter)	0.0992	0.0023	0.3898	0.0156	1.4038

Table 1: Time Analysis of Different Processes.

Process	Template Rendering Early Stage	Template Rendering Late Stage	Phase Correlation	Camera Pose Sampling	Training
Time (s/iter)	4423.0519	102.5506	–	74.8994	1.4038

Table 2: Time Analysis of Naïve Grid Search.

C Ablation Study of the Number of Discretizations

We conduct an ablation study on the number of discretizations on ShapeNet Cars. We report FID and early-stage template rendering time in Tab. 3. Reducing discretized bins of $\theta \times \phi$ from 36×18 to 24×12 or lower worsens FID due to larger quantization steps. Increasing bins beyond 36×18 obtains comparable results but yields increasing costs. Note that our method with 12×6 bins still outperforms PoF3D in terms of FID_{gt}.

$\theta \times \phi$	12×6	24×12	36×18	48×24	60×30
FID_{gt} / FID_{est}	11.29 / 9.57	7.48 / 7.35	5.95 / 6.55	5.51 / 6.25	5.96 / 6.48
time s/iter	0.0559	0.2126	0.4966	0.8465	1.3205

Table 3: Different number of bins for discretization.

D Comparison on the FFHQ and AFHQ datasets

We evaluate our method on the FFHQ and AFHQ datasets. We set the azimuthθ range as 120 degrees according to the distribution prior for the datasets, and discretize it into 36 values. Since the elevation variation is small, we directly set the elevation angle as 90 degrees.

We achieve comparable results with PoF3D on the FFHQ dataset and better results on the AFHQ dataset, as shown in Tab. 4. Despite PoF3D performing well in generated poses, their results degenerate in GT poses. We achieve comparable results with EG3D on the AFHQ dataset, even though EG3D uses GT poses (with pose condition) and we do not. Despite not using GT poses, our result is only slightly worse than EG3D on the FFHQ dataset. Fig. 1 demonstrates our qualitative results.

Fig. 1: Our generated samples on face and cat datasets.

	FFHQ				AFHQ	
Method	$\text{Depth}_{gt}\downarrow$	$\text{Depth}_{est}\downarrow$	$\text{FID}_{gt}\downarrow$	$\text{FID}_{est}\downarrow$	$\text{FID}_{gt}\downarrow$	$\text{FID}_{est}\downarrow$
EG3D	**0.29**	-	**4.80**	-	5.56	-
PoF3D	0.37	**0.29**	5.13	**4.99**	16.95	5.46
Ours	0.36	0.35	5.64	5.37	**4.52**	**4.37**

Table 4: Comparison on the FFHQ and the AFHQ datasets.

E Qualitative Results of Camera Pose Estimation

Fig. 2, Fig. 3, Fig. 4 and Fig. 5 show the estimated poses of real images on CompCars, SDIP Elephant, Shapenet Cars and LSUN Plane, respectively. The first row is the real image $\mathbf{I}$ and its corresponding DINO feature $\mathbf{F}$, and the second row is the best-matching template feature $\tilde{\mathbf{F}}_k^*$. The corresponding camera pose of $\tilde{\mathbf{F}}_k^*$ indicates the estimated pose of the real image. Note that we only perform phase correlation on CompCars and LSUN Plane since Shapenet Cars and SDIP Elephant do not have large variations in scale and in-plane rotation. The results demonstrate that our method can perform fairly accurate camera pose estimation. Fig. 6 shows the failure cases of pose estimation. The estimated pose may not be accurate due to the object articulation, partial observation, significant geometry difference, lack of modeling object translation, etc. This could be solved by introducing multiple templates, modeling more freedom of degrees of camera pose, and disentangling the geometry information during the matching process in the future. We remark that despite some instances of inaccurate pose estimation, the overall distribution of poses is generally precise. This accuracy enables the trained generator to produce objects with complete geometry and the capability for comprehensive 360-degree view synthesis.

Fig. 2: Qualitative Pose Estimation on CompCars.

F Uncurated Qualitative Results

Fig. 7, Fig. 8, Fig. 9 and Fig. 10 show the uncurated qualitative results on SDIP Elephant, CompCars, LSUN Plane and Shapenet Cars. Our results demonstrate good fidelity and diversity.

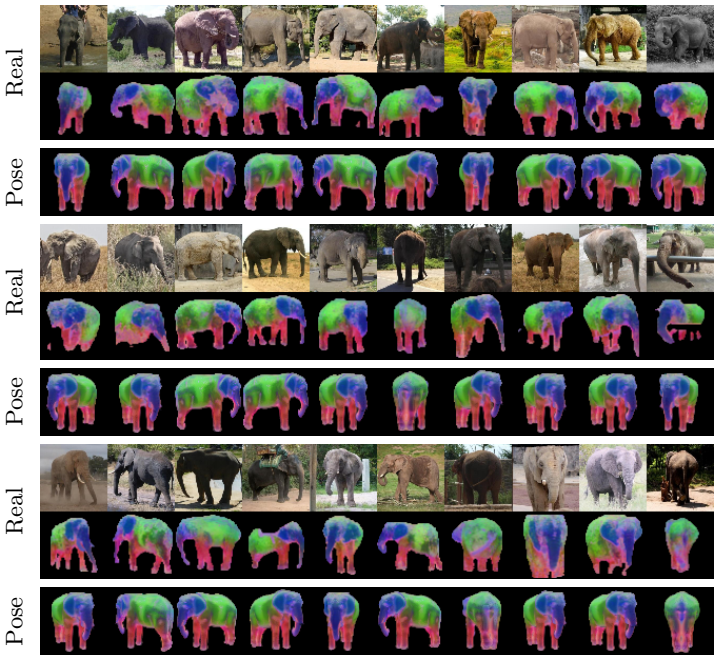

Fig. 3: Qualitative Camera Pose Estimation on SDIP Elephant.

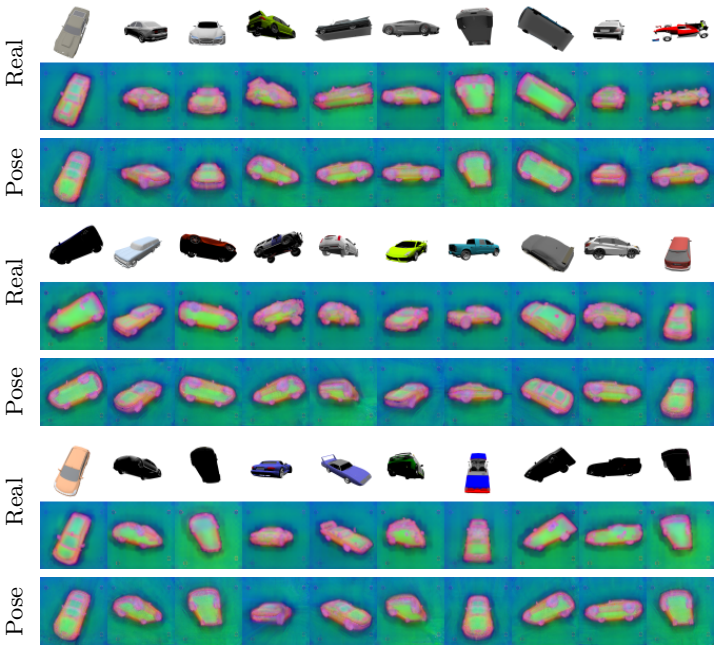

Fig. 4: Qualitative Camera Pose Estimation on Shapenet Cars.

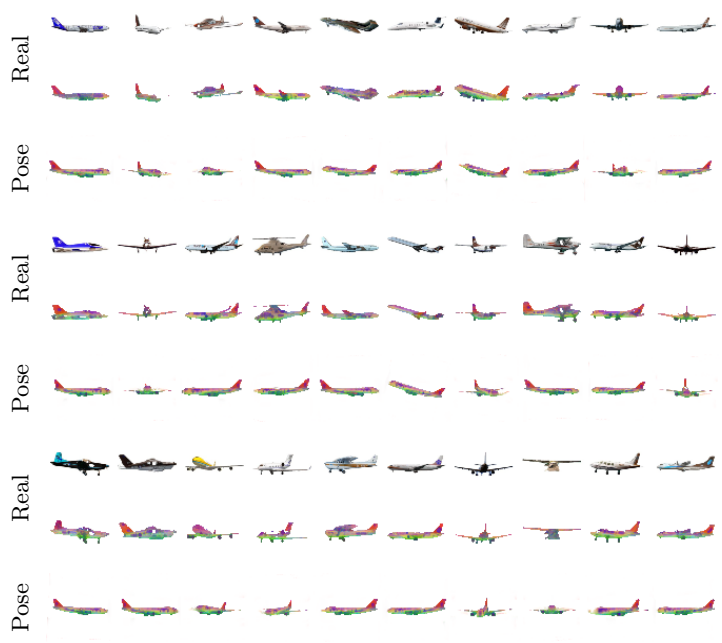

Fig. 5: Qualitative Camera Pose Estimation on LSUN Plane.

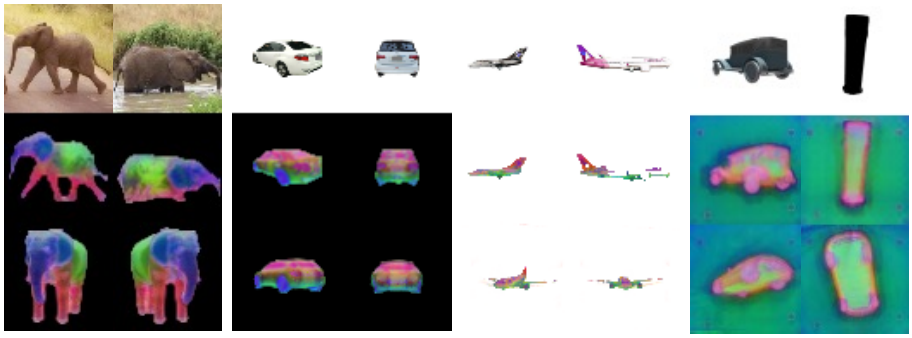

Fig. 6: Failure Camera Pose Estimation

Fig. 7: Uncurated Result on SDIP Elephant.

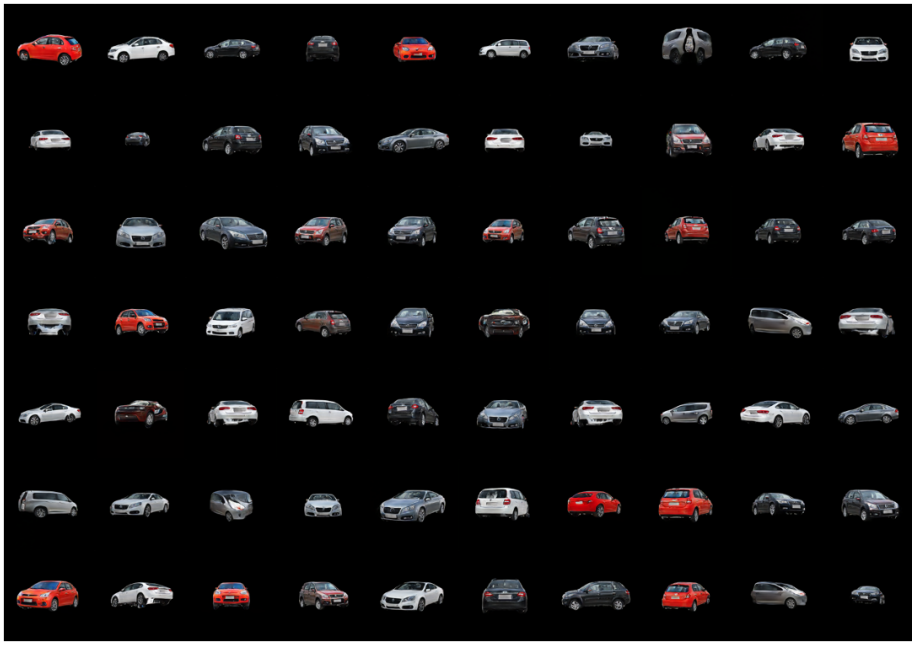

Fig. 8: Uncurated Result on CompCars.

Fig. 9: Uncurated Result on LSUN Plane.

Fig. 10: Uncurated Result on Shapenet Cars.

References

1. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks. In: CVPR (2022)
2. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv.org **1512.03012** (2015)
3. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
4. Mokady, R., Yarom, M., Tov, O., Lang, O., Daniel Cohen-Or, Tali Dekel, M.I., Mosseri, I.: Self-distilled stylegan: Towards generation from internet photos (2022)
5. Yang, L., Luo, P., Loy, C.C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 3973–3981. IEEE Computer Society (2015). https://doi.org/10.1109/CVPR.2015.7299023, https://doi.org/10.1109/CVPR.2015.7299023
6. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv.org **1506.03365** (2015)