Learning 3D-aware Image Synthesis with Unknown Pose Distribution
– *Supplementary Material* –

Zifan Shi[†*1] Yujun Shen[†2] Yinghao Xu[*3] Sida Peng[4] Yiyi Liao[4] Sheng Guo[2]
Qifeng Chen[1] Dit-Yan Yeung[1]
[1]HKUST [2]Ant Group [3]CUHK [4]Zhejiang University

In this supplementary material, we first provide the training details of our PoF3D in Sec. A. In Sec. B, we describe the details of implementations of baselines. Sec. C provides more qualitative results. Moreover, we show the syntheses under steep angles. Sec. D discusses the limitations as well as the potential future work of PoF3D. Ethical considerations are also provided.

A. Training and Implementation Details

Training Details. Most of our training parameters are the same as those in EG3D [1]. We reset the loss weight λ for gradient penalty to 1.0, 5.0, 0.3 for FFHQ [5], Cats [8], and Shapenet Cars [2], respectively. γ, the weight for pose loss, is set to 2, 10, 2 for FFHQ, Cats, and Shapenet Cars. All losses are used for training iteratively. For FFHQ and Cats, models are trained on the NeRF resolution of 64×64 and the image resolution of 256×256. While for models on Shapenet Cars, the NeRF resolution is 64×64 and the image resolution of 128×128, following the setting in [1]. Models on FFHQ and Shapenet Cars are trained end-to-end on 25000K images for around 6 days on 8 NVIDIA A100 GPUs. Due to the limited amount of data in Cats dataset, we follow the setting in EG3D [1] to finetune the pretrained model of FFHQ on Cats dataset for 600K images.

Additional Implementation Details. We would like to illustrate more implementation details in addtion to details in Sec.3.5. PoF3D is built upon EG3D [1], including the triplane generator, decoder, volume rendering, super-resolution module and dual discriminator. In the triplane generator, we disable the pose conditioning and add a pose learner. The pose learner consists of two linear layers with hidden size 512 and a leaky ReLU in between. It takes in a w-space code of size 512 and outputs camera poses of dimension 2, an azimuth angle and an elevation angle. In the dual discriminator, we add a pose predictor. The pose predictor has the same structure as the pose learner except that the hidden size is 4096 and the input is feature maps of resolution 4 in the discriminator.

† indicates equal contribution.
* This work was done during an internship at Ant Group.

B. Baselines

CAMPARI [6] is a 3D-aware image synthesis method that models camera distribution during training. We use the official implementation for all experiments. For FFHQ, Cats and Shapenet Cars dataset, we keep the settings identical to the provided configurations for CelebA, Cats and Carla, but we allow the learning of azimuth and elevation angle only. Following [6], the prior distribution is set to Gaussian distribution $\mathcal{N}(0, 13.5°)$ for azimuth and elevation on FFHQ and Cats, and a uniform distribution over the entire azimuth and elevation for Shapenet Cars. Other camera parameters are fixed to the one learned in the original settings. Besides, we follow the original setting that the camera distribution will be fixed for later stages of training on FFHQ.

EG3D [1] is also one of the state-of-the-art methods in 3D-aware image synthesis, which leverages ground-truth camera poses for training. We use the official implementation for all experiments. For FFHQ dataset, since the checkpoint for 256×256 has not been released yet, we use the provided configuration to train on the NeRF resolution of 64×64 and image resolution of 256×256. For Cats dataset, we make use of the pose annotations processed by [3]. Other settings are identical to the original one for cat dataset, and the model is trained on the NeRF resolution of 64×64 and image resolution of 256×256 as well. Moreover, we follow [1] to finetune the model with the checkpoint of FFHQ on Cats dataset rather than train the model from scratch. We adopt the checkpoint of Shapenet Cars provided by the authors for evaluation.

CAMPARI+EG3D is a combination of CAMPARI [6] and EG3D [1], where the pose distribution learning network in CAMPARI is merged into the framework of EG3D. Concretely, in EG3D, we do not sample poses from the collection of real poses for generation, but sample a pose from a prior distribution and transform it into a proper one with a network. The transformed pose is then used for rendering.

Figure S1. **Untruncated samples on FFHQ [5].** For each generated identity, we show the underlying geometry under two views and appearance under three views.

Figure S2. **Synthesized samples on Cats [8] with truncation 0.7.** For each generated cat, we show the underlying geometry under two views and appearance under three views.

For real data, we still leverage the ground-truth poses for conditioning. The training strategy and the initialization of priors for pose learning in CAMPARI+EG3D follows those in CAMPARI. Other parameters such as camera intrinsic matrix are identical to those used in EG3D.

C. Additional Results and Analysis

C.1. Qualitative Results

We provide more qualitative results in Figs. S1 to S3. A demo video, is also available to show the qualitative comparison with baselines. Our results are on par with those generated from EG3D [1] and much better than those from CAMPARI [6].

C.2. Syntheses under Steep Angles

We synthesize images under steep camera poses on FFHQ dataset [5] in Fig. S4. Since CAMPARI fails to learn a proper pose distribution and generates sharp and bumpy shapes as discussed in Sec. 4.2, it finds it hard to synthesize reasonable images under larger rotation. EG3D leverages ground-truth poses for training and is good at generating images under extreme views. However, it tends to generate extremely sharp noses. Ours, however, can synthesize natural noses and high-quality images under steep angles without using any pose prior.

C.3. Training Behavior

We show the trends of FID, depth error, pose error and Jensen-Shannon divergence in Fig. S5 as training progresses. Generally, the network learns fast at first and slows down later. The learning of the data distribution is slower than the other three aspects.

Figure S3. **Synthesized samples on Shapenet Cars [2] with truncation 0.7**. For each generated car, we show the underlying geometry under two views and appearance under three views.

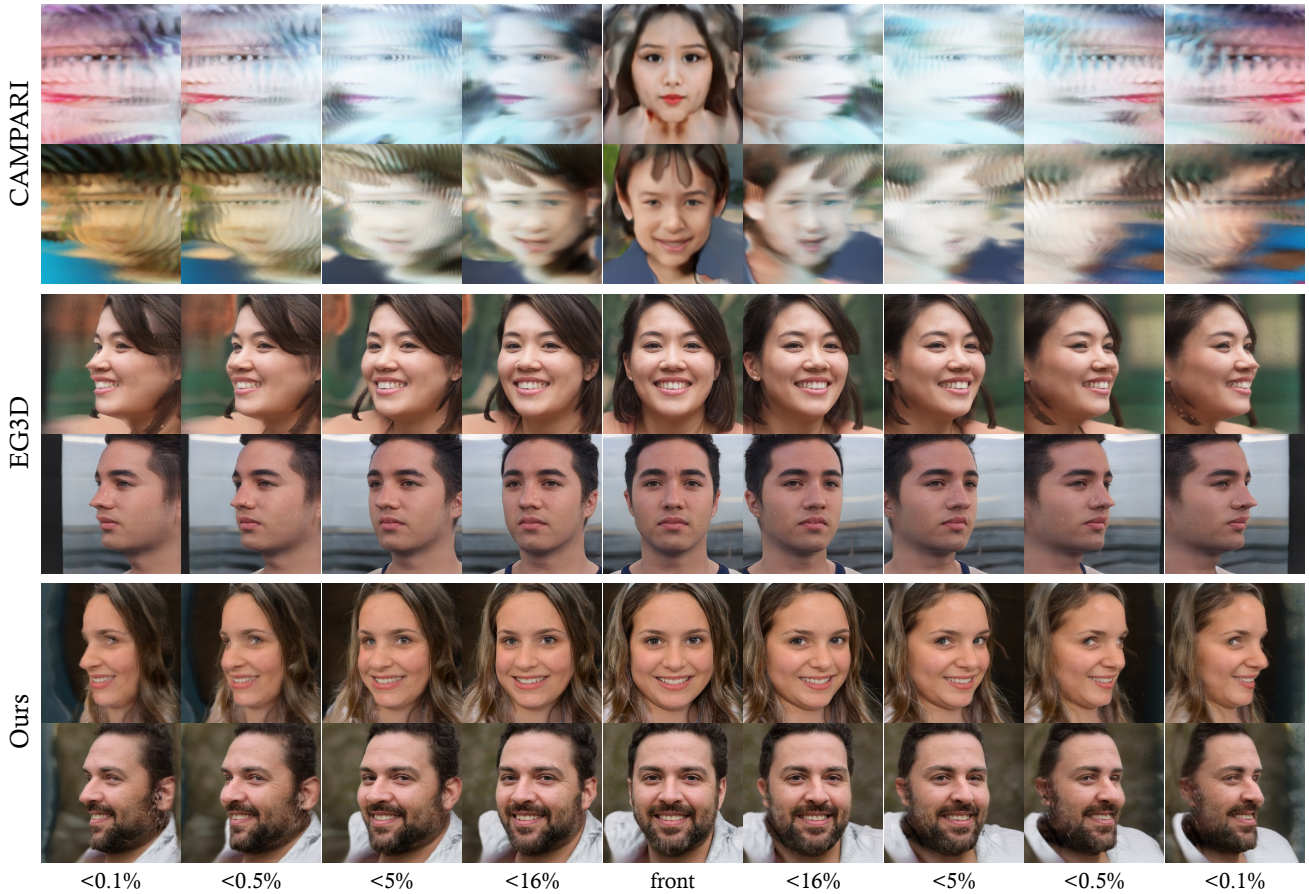


Figure S4. **Syntheses under steep angles.** $< X\%$ denotes less than X percent of training cases are trained under that pose.

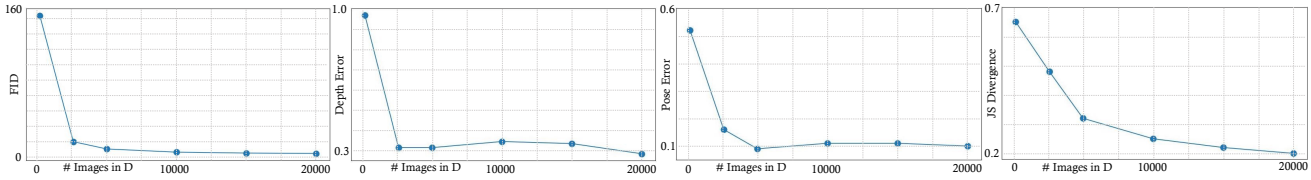


Figure S5. **Behavior as training progresses.** Zoom in for details.

C.4. 3D Reconstruction using COLMAP

We render 128 views from a random code using the same camera trajectory as [1], to reconstruct a point cloud using COLMAP. As shown in Fig. S6, the *dense* point cloud indicates the good multi-view consistency achieved by PoF3D.

Figure S6. **3D reconstruction using COLMAP**.

C.5. Distribution Difference between G and D

Fig. S7 visualizes the distribution discrepancy of G and D on FFHQ, where the pose error is 0.09. The reason for the distribution discrepancy is that in GAN training, it is hard to optimize to the optimal point. A sub-optimal solution brings the difference on pose distributions in G and D, as well as the non-zero FID. How to make G and D equivalent is a long-standing problem.

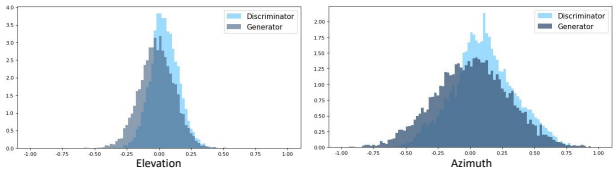


Figure S7. **Pose distributions in G and D** trained on FFHQ.

C.6. Full analysis of Fig. 1

We provide the full analysis of baselines in Fig. 1 with both smaller range of pose distribution and larger range of pose distribution, showing how sensitive existing works are to the pre-estimated pose prior. As shown in Fig. S8, for π-GAN (top), with [-0.5, 0.5] (middle) as the optimal prior, using [-0.3, 0.3] (left) and [-0.7, 0.7] (right) result in *(i)* planar and noisy shape as well as *(ii)* the loss of canonical space. Similarly, for CAMPARI (bottom), with 0.24 (middle) as the optimal pose std, using 0.12 (left) and 0.36 (right) harm the performance drastically.

D. Discussion

D.1. Limitations and Future Work

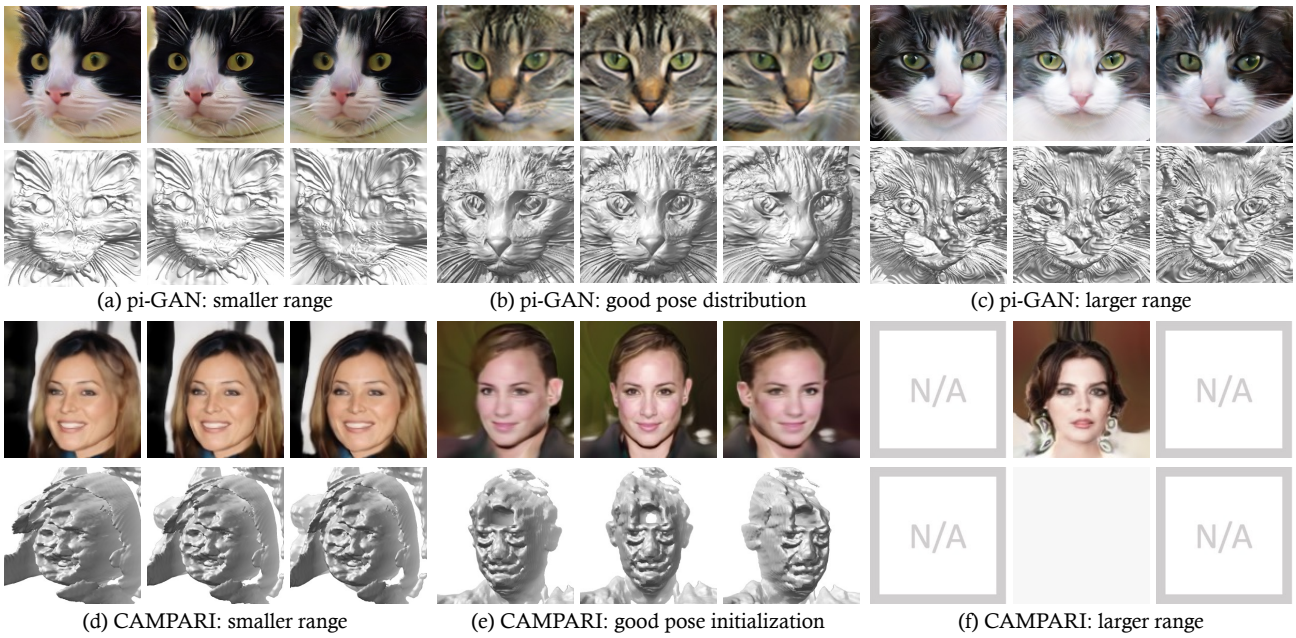
Though PoF3D generates high-quality images and decent underlying shapes without pose priors, there are still some artifacts on the geometry. For example, the eye balls have concave underlying shapes, leading to incorrect movement during rotation. We believe extra geometry supervision shall be added on them to fix the problem. Sometimes bumpy regions can be observed. We think with larger batch size, the pose distribution can be learnt more accurately and thus leads to more decent shapes. Texture sticking effect is also noticed during rotation, which might be mitigated by replacing the StyleGAN2 backbone with StyleGAN3 [4].

Despite the well-captured pose distribution, PoF3D sometimes confuses the front with the rear of the car. The reason is that the front and the rear of cars look similar to each other in Shapenet Cars [2], a synthetic dataset. A more powerful pose predictor should be introduced into the discriminator to improve the ability of judging the front and the rear of cars, which we leave for future work.

We do not model the foreground and the background separately, and thus the background is close to the foreground objects from time to time. Techniques, such as NeRF++ [7], can be integrated into our framework to model the foreground and background independently, which is also a potential future direction to be explored.

D.2. Ethical Considerations

PoF3D can benefit vision and graphics applications, such as gaming and content creation. However, it also poses a threat because generative models can be misused for DeepFake-related applications, *e.g.*, human face editing and talking head generation. We hope that DeepFake detection algorithms can be developed to avoid such misuse. In addition, verification cues, such as forensics, offer another solution to mitigate the problem.



(a) pi-GAN: smaller range (b) pi-GAN: good pose distribution (c) pi-GAN: larger range

(d) CAMPARI: smaller range (e) CAMPARI: good pose initialization (f) CAMPARI: larger range

Figure S8. **Results of different pose priors.** FID scores of (a) to (f) are 12, 17, 13, 36, 28, and 26.

References

[1] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 4

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 3, 4

[3] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1

[4] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Adv. Neural Inform. Process. Syst.*, 2021. 4

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2

[6] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. In *International Conference on 3D Vision (3DV)*, 2021. 1, 2

[7] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 4

[8] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *Eur. Conf. Comput. Vis.*, 2008. 1, 2