FEW-SHOT GENERATION BY MODELING STEREOSCOPIC PRIORS

Yuehui Wang *Qing Wang* *Dongyu Zhang* *

Sun Yat-Sen University

ABSTRACT

Few-shot image generation, which aims to generate images from only a few images for a new category, has attracted some research interest in recent years. However, existing few-shot generation methods only focus on 2D images, ignoring 3D information. In this work, we propose a few-shot generative network which leverages 3D priors to improve the diversity and quality of generated images. Inspired by classic graphics rendering pipelines, we unravel the image generation process into three factors: shape, viewpoint and texture. This disentangled representation enables us to make the most of both 3D and 2D information in few-shot generation. To be specific, by changing the viewpoint and extracting textures from different real images, we can generate various new images even in data-scarce settings. Extensive experiments show the effectiveness of our method.

Index Terms— Few-shot Learning, Generative Adversarial Network, Data Augmentation, Deep Learning

1. INTRODUCTION

The challenge of learning new concept from very few examples, often called *few-shot learning* or *low-shot learning*, is a long-standing problem. In computer vision, a lot of attempts have been made to explore few-shot recogniition, few-shot classification and few-shot image translation. In spite of their remarkably success, few-shot generation has received little attention in the past, possibly due to its considerable difficulty. Some recent works [1, 2, 3] explored the ability of few-shot generation under specific circumstances. To be more concrete, [1] proposed a meta-learning based method of generating personalized talking head images. [2] presented a framework to learn a generative model from a single natural image. However, they only focus on the information brought by 2D image dataset, we consider to use 3D priors to guide image generation.

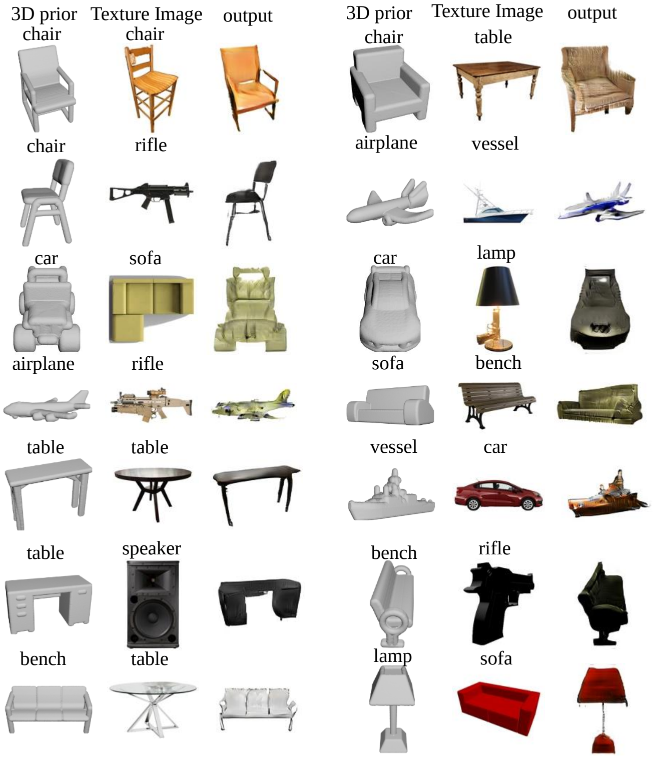

Fig. 1. Qualitative results. When given a real 3D prior (with determined *shape* and *viewpoint*) and a texture image, our model successfully apply the texture to the prior and generate realistic images without mode collapse nor mode confusion.

In this paper, we explore image generation in few-shot settings and simultaneously care for 3D information: shape, viewpoint and texture. First, the *shape* of the objects in the generated images depends on the category of our 2D image dataset (*e.g.*, car, chair and table). Second, by changing the *viewpoint* of the camera in the process of rendering 3D priors, we can get a variety of 2.5D samples (*e.g.*, depth images). After that, we extract the *texture* of an arbitrarily sampled image from the 2D image dataset. Finally, we recombine these three factors, with our novel generative model *Few-shot Generative Network with 3D priors (FGN-3D)*, to generate new images. Figure 1 shows some qualitative results produced by our model, where the desired texture is applied to the specified 3D prior, regardless of their categories.

*Corresponding author (e-mail: zhangdy27@mail.sysu.edu.cn). This work was partly supported by the National Natural Science Foundation of China under Grant No. 61876224, the Natural Science Foundation of Guangdong Province under Grant No. 2019A1515010939, and partly supported by the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (Grant: 2017KZDXM081, 2018KZDXM066), the Featured Innovative Projects of Department of Education of Guangdong Province (Grant: 2019KTSCX175).

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

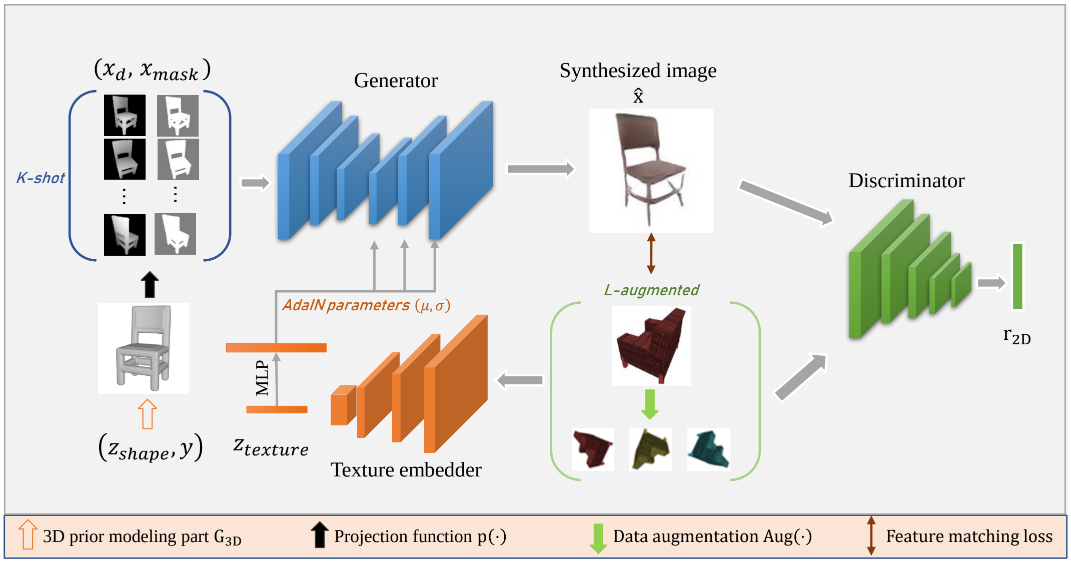

Fig. 2. Overview of the proposed FGN-3D model. To generate image $\hat{x}$, we first extract k depth and mask pairs from a 3D prior, after that we encode l augmented texture images into $Z_{texture}$. Finally we recombine them and choose the one with the lowest feature matching loss as the output.

Note that [4] provided a similar way to generate images by utilizing disentangled 3D information. However, we focus on conditional image generation for multi-category in few-shot settings, while they train one model for a single-category and the model are not designed for few-shot settings.

The few-shot learning ability of our proposed method is obtained through two stages: (a) meta-learning and (b) fine-tuning. Meta-learning is performed on *base classes* where a large training set of 3D collections and corresponding 2D real images is available. In the course of meta-learning, our system simulates few-shot learning tasks and learns to transform 2.5D samples (*e.g.*, depth images) into realistic RGB images. After that, we fine tune our models, with high-capacity generator and discriminator pre-trained via meta-learning, on *novel classes* where the training data is scarce.

Summarizing the contributions of this paper, we:

- Propose a two-stage training model called **Few-shot Generative Network with 3D priors (FGN-3D)** which introduces 3D priors into image generation in few-shot scenarios.
- Demonstrate that our model produces the state-of-the-art results compared to extended baselines while retaining good generalization performance.

2. METHOD

2.1. Architecture and Notation

First we'd like to introduce the necessary notations. Let $\mathbb{I}$ denote the 2D RGB image space $\mathbb{R}^{H\times W\times 3}$, $\mathbb{V}$ denote the 3D prior space $\mathbb{R}^{W\times W\times W}$ and $\mathbb{C} = \{0, \ldots, L\}$ denote the discrete label space. Our training dataset S consists of 3D collections $\{v_i\}_i^N$ and real 2D RGB images $\{x_j\}_j^M$, *i.e.*, $S = \{\{v_i\}_i^N, \{x_j\}_j^M\}$. Note that we use i and j to accentuate *no* pair relationship between 3D and 2D data. For few-shot learning, we separate the label space $\mathbb{C}$ into $\mathbb{C}_{base}$ where large number of training data are available and $\mathbb{C}_{novel}$ which is underrepresented in data.

Then we introduce the network architectures of different modules in the framework. Figure 2 shows an overview of the proposed FGN-3D framework. Specifically, for 3D priors modeling part, two networks are trained:

- The *3D priors generator* G_{3D} takes a latent code z_{shape} sampled from a normal distribution, a class label $y \in \mathbb{C}_{base}$ and outputs a 3D instance $\hat{v}$, *i.e.*, $\hat{v} = G_{3D}(z_{shape}, y)$.
- The *3D priors discriminator* D_{3D} takes a 3D instance v, a class label $y \in \mathbb{C}_{base}$ and outputs a single scalar r_{3D}, *i.e.*, $r_{3D} = D_{3D}(v, y)$. which indicates whether the input v is a real instance from class y.

For 2D image generation part, three networks are trained:

- The *texture embedder* E maps a real image x into a vector $z_{texture}$, *i.e.*, $z_{texture} = E(Aug(x); \phi)$ [5]. Here, $Aug(\cdot)$ represents data augmentation operations and ϕ is the model parameters. Note that E is designed to be class-agnostic to leverage all training data and increase the diversity of generated images.

- The *image generator* G_{2D} takes a depth image x_d, texture latent code $z_{texture}$ and outputs a synthesized image $\hat{x}$, *i.e.*, $\hat{x} = G_{2D}(x_d, z_{texture}; \psi)$. Here x_d is obtained by employing a fully differentiable projection function p with a specific viewpoint vp on a 3D prior v: $x_d = p(v, vp)$ [4]. Here, ψ denotes model parameters that are learned in the meta-learning stage. In general, during meta-learning, we aim to learn ψ such that G_{2D} are able to maximize the similarity between its outputs and the real image.

- The *image discriminator* D_{2D} takes a 2D image x, a class label $y \in \mathbb{C}_{base}$ and outputs a single scalar r_{2D}, *i.e.*, $r_{2D} = D_{2D}(x, y; \varphi)$. which indicates whether the input x is a real image from class y.

The proposed FGN-3D framework aims at generating a 2D RGB image $\hat{x} \in \mathbb{I}$ in which the object shape is determined by label $y \in \mathbb{C}$ and input real image x specifies its texture.

2.2. Meta-learning on Base Classes

3D priors modeling. We base our 3D priors generator G_{3D} and discriminator D_{3D} on the 3D-GAN architecture proposed by [6]. However, vanilla 3D-GAN suffers model collapse and unstable training process when extended to multi-class generation setting. To address these problems, the Wasserstein distance [7] and spectral normalization [8] are used. Besides, following the advice of [9], we feed the conditional information y into the discriminator by projection instead of concatenation. Specifically, when trained separately the loss function of modeling 3D priors is:

$$\begin{aligned}\min_{G_{3D}} \max_{D_{3D}} \mathcal{L}_{3D} &= \mathbb{E}_v[D_{3D}(v, y)] \\ &- \mathbb{E}_{z_{shapep}}[D_{3D}(G_{3D}(z_{shape}, y), y)].\end{aligned} \tag{1}$$

2D image generation. The training process of 2D image generation part is done by simulating episodes of K-shot learning. In each episode, we need a pair of a 3D prior and an image $\{v, x\}$ for training, where x is randomly sampled from the training dataset, while v is either generated by G_{3D} (during joint training) or randomly sampled from the training dataset (during individual training). Then, K depth images $\{x_{d1}, x_{d2}, \dots, x_{dk}\}$ are obtained by changing the viewpoint in the projection function $p(v, vp)$. Additionally, we can also get K corresponding image masks $\{x_{mask1}, x_{mask2}, \dots, x_{mask}\}$ with a simple threshold, which will later be used to regularize the synthesized image. To increase the diversity of generated images, we produce L augmented real images: $\{x_1, x_2, \dots, x_l\} = Aug(x)$ before feeding them into the texture embedder E.

Here we use a CycleGAN-like [10] architecture. We employ two generators and two discriminators: forward (from depth to real RGB) generator G_{fw} and discriminator D_{fw}, backward (from real RGB to depth) generator G_{bw} and discriminator D_{bw}. We train these four networks jointly with adversarial losses and cycle-consistency losses. More formally, when training forward, the adversarial loss is given by:

$$\mathcal{L}_{fw} = \mathbb{E}_x[\log(D_{fw}(x))] + \mathbb{E}_{(x_d, \{x_1, \dots, x_l\})}[\log(1 - D_{fw}(\hat{x})], \tag{2}$$

where

$$\hat{x} = G_{fw}(x_d, E(\{x_1, \dots, x_l\})). \tag{3}$$

When training backward:

$$\mathcal{L}_{bw} = \mathbb{E}_{x_d}[\log(D_{bw}(x_d))] + \mathbb{E}_x[\log(1 - D_{bw}(G_{bw}(x))]. \tag{4}$$

Cycle-consistency losses are also used to enforce the bijective relationship between the two domains in the forward and backward phase:

$$\mathcal{L}^{cyc}_{fw} = \mathbb{E}_x[||G_{fw}(G_{bw}(x)) - x||_1^1], \tag{5}$$

and

$$\mathcal{L}^{cyc}_{bw} = \mathbb{E}_{(x_d, \{x_1, \dots, x_l\})}[||G_{bw}(\hat{x}) - x_d||_1^1]. \tag{6}$$

Additionally the feature matching loss [11] is employed to make sure our generated $\hat{x}$ share the same texture as the input real image x in general. Removing the last layer from D_{fw}, we construct a feature extractor D'_{fw} which is then used to extract features from $\hat{x}$ and $\{x_1, \dots, x_l\}$:

$$\mathcal{L}_{FM} = \mathbb{E}_{(\hat{x}, \{x_1, \dots, x_l\})}[||D'_{fw}(\hat{x}) - \sum_l \frac{D'_{fw}(x_l)}{L}||_1^1]. \tag{7}$$

At this point, we write the full loss of the 2D image generation process as

$$\mathcal{L}_{2D} = \mathcal{L}_{fw} + \mathcal{L}_{bw} + \mathcal{L}^{cyc}_{fw} + \mathcal{L}^{cyc}_{bw} + \lambda_{fm}\mathcal{L}_{FM}, \tag{8}$$

where λ_{fm} shows the weight of feature matching loss.

Full Model. Our full objective in this stage is as follows:

$$\min_{(G_{3D}, G_{fw}, G_{bw})} \max_{(D_{fw}, D_{bw})} \mathcal{L}_{3D} + \mathcal{L}_{2D}. \tag{9}$$

2.3. Fine-tuning on Novel Classes

Once the meta-learning has finished, the forward generator G_{fw} is able to generate RGB image for novel class, which is unseen during meta-learning stage, conditioned on the depth images projected from 3D priors. In this stage, the fine-tuning loss of image generation is:

$$\mathcal{L}^{finetune}_{2D} = \mathbb{E}[\log(D_{2D}(x))] + \mathbb{E}[\log(1 - D_{2D}(\hat{x})], \tag{10}$$

where

$$\hat{x} = G_{2D}(p(v, vp), E(\{x_1, \dots, x_l\})). \tag{11}$$

The full objective in this stage is:

$$\min_{G_{2D}} \max_{D_{2D}} \mathcal{L}^{finetune}_{2D} + \lambda_{fm}\mathcal{L}_{FM}. \tag{12}$$

Methods\Classes	car	chair	airplane	sofa	rifle	table	lamp	vessel	bench	speaker	mFID ↓
c-DCGAN	153.2	245.0	258.8	201.5	186.5	-	-	-	-	-	209.0
c-LSGAN	175.6	235.4	224.6	177.6	137.3	-	-	-	-	-	190.1
c-WGAN-GP	143.1	174.1	217.6	156.9	110.9	-	-	-	-	-	160.5
extended-VON	81.3	**58.8**	96.1	58.9	89.8	219.7	240.5	223.3	281.3	266.6	161.6
FGN-3D (ours)	**77.2**	64.7	**90.2**	**55.6**	**86.2**	**89.0**	**102.4**	**111.8**	**98.6**	**106.4**	**88.2**

Table 1. Quantitative comparisons with FID, the lower the better. Here '-' represents severe model collapse. Note that even in base class, where baselines use all the training data while we use only part of it, our model also shows SOTA performance.

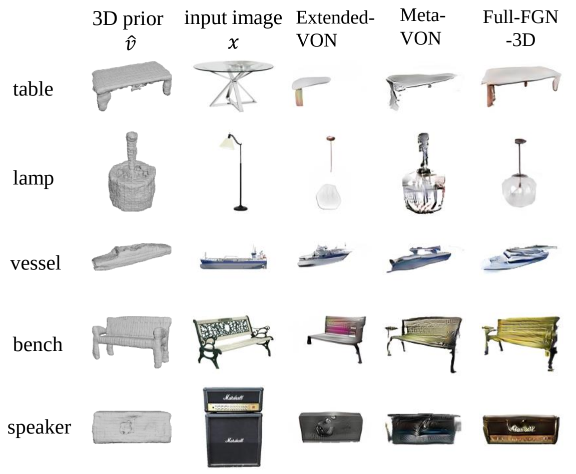

Fig. 3. Quantitative comparison between meta-VON and our method with $T = 20$ on novel classes.

Methods\Classes	table	lamp	vessel	bench	speaker
meta-VON	93.4	105.0	133.3	144.9	106.9
meta-FGN-3D	95.5	118.9	115.1	102.3	116.8
full-FGN-3D	**89.0**	**102.4**	**111.8**	**98.6**	**106.4**

Table 2. Analysis on benefits of introducing two-stage training strategy for few-shot generation.

3. EXPERIMENT

3.1. Experimental Setting

Baselines. We compare our method against five popular GAN variants: DCGAN [12], LSGAN [13], WGAN-GP [7] and VON [4]. Since the vanilla baselines are class-specific, we extend them to support multi-class generation for fair comparison. Detail extensions are as follows: *3D-free GAN variants:* We simply extend them into conditional generation based on class labels, *i.e.*, c-DCGAN, c-LSGAN and c-WGAN-GP. *Extended-VON:* We introduce multi-class generation setting (conditional 3D-GAN) and texture extraction ability (texture encoder) into VON [4].

Datasets. *3D collections:* We use ShapeNet [14] for 3D priors modeling. Specifically, we choose the five largest classes (car, chair, airplane, sofa and rifle) as our base classes $\mathbb{C}_{base}$. For each one of them, we limit the number of CAD models to 500. The next five largest classes (table, lamp, vessel, bench, speaker) are novel classes $\mathbb{C}_{novel}$, where there are at most 20 models for each one of them. *2D images:* We crawled 500 images from Google for each class in $\mathbb{C}_{base}$, while each class in $\mathbb{C}_{novel}$ holds 20 images at most. Note that there is no dataset limitation for baselines, for example, extended-VON uses 6,777 and 3,513 CAD models with 1,963 and 2,605 images for chair and car respectively, and are trained for 500 epochs according to [4], whereas we train our models in few-shot setting for 200 epochs on base classes and fine-tune for 20 epochs to generate better results.

Metrics. We calculate Fréchet Inception Distance (FID) [15] to evaluate distribution matching between generated images and real images, lower FID values mean better image quality and diversity.

3.2. Main Results

Qualitative evaluation. Figure 1 demonstrates some images generated by the proposed model, when given a 3D prior and a texture image. Figure 3 shows more examples on novel classes with $T = 20$, where T represents the number of samples used in fine-tuning stage. Note that the generated images from the proposed model hold better quality and diversity both in base classes and few-shot setting.

Quantitative evaluation. Table 1 reports quantitative results of our model and all baselines on both base classes and novel classes. Table 2 analyses the benefits of introducing two-stage training strategy. We set $K = L = 1$ in meta-FGN-3D and $K = 5$ and $L = 4$ in full-FGN-3D. Comparing Table 1 and Table 2, we can verify that our training strategy is very helpful for few-shot generation and that making full use of the information brought by 3D priors (increasing K and L) helps to generate better images.

4. CONCLUSION

In this paper, we propose a two-stage training model (FGN-3D), which introduces 3D priors into image generation in few-shot scenarios, based on generative adversarial networks. Empirical evidence has been provided that by fully utilizing 3D structure information, our model outperforms all extended baselines.

5. REFERENCES

[1] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky, “Few-shot adversarial learning of realistic neural talking head models,” in *ICCV*, 2019.

[2] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli, “Singan: Learning a generative model from a single natural image,” in *ICCV*, 2019.

[3] Yan Hong, Li Niu, Jianfu Zhang, Weijie Zhao, Chen Fu, and Liqing Zhang, “F2gan: Fusing-and-filling gan for few-shot image generation,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2535–2543.

[4] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and William T. Freeman, “Visual object networks: Image generation with disentangled 3D representations,” in *NeurIPS*, 2018.

[5] Xun Huang and Serge Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[6] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum, “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling,” in *NeurIPS*, 2016.

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, “Improved training of wasserstein gans,” in *NeurIPS*, 2017.

[8] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida, “Spectral normalization for generative adversarial networks,” in *ICLR*, 2018.

[9] Takeru Miyato and Masanori Koyama, “cgans with projection discriminator,” in *ICLR*, 2018.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017.

[11] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz, “Few-shot unsupervised image-to-image translation,” in *ICCV*, 2019.

[12] Alec Radford, Luke Metz, and Soumith Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[13] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[14] 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.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.